
Aries Cloud Agent Python Documentation

See Contributors list on [GitHub](#)

Feb 25, 2021

Aries Cloud Agent Python - Modules

1	aries_cloudagent	3
1.1	aries_cloudagent package	3
1.1.1	Subpackages	3
1.1.2	Submodules	374
1.1.3	aries_cloudagent.version module	374
2	Indices and tables	375
	Python Module Index	377
	Index	385

Hyperledger Aries Cloud Agent Python (ACA-Py) is a foundation for building decentralized identity applications and services running in non-mobile environments.

This is the Read The Docs site for the Hyperledger [Aries Cloud Agent Python](#). This site contains only the ACA-Py docstrings documentation extracted from the Python Code. For other documentation, please consult the links in the Readme for the [ACA-Py GitHub Repo](#).

If you are getting started with verifiable credentials or Aries, we recommend that you start with this [verifiable credentials and agents getting started guide](#).

Want to quick overview of the deployment model for ACA-Py? See [this document](#).

To investigate the code, use search or click the package links in the left menu to drill into the modules, subpackages and submodules that make up ACA-Py.

Developers that are interested in what DIDComm protocols are supported in ACA-Py should take a look at the [protocols](#) package. These should align with the corresponding [aries-rfcs protocols](#). Decorators defined in aries-rfcs and implemented in ACA-Py can be found [here](#). Some general purpose subpackages that might be of interest include [wallet](#) and [storage](#). For those agents playing different roles in a verifiable credential exchange, take a look at the [issuer](#), [holder](#) and [verifier](#) packages.

Please see the [ACA-Py Contribution guidelines](#) for how to contribute to ACA-Py, including for how to submit issues about ACA-Py.

1.1 aries_cloudagent package

Aries Cloud Agent.

1.1.1 Subpackages

aries_cloudagent.admin package

Submodules

aries_cloudagent.admin.base_server module

Abstract admin server interface.

class aries_cloudagent.admin.base_server.**BaseAdminServer**

Bases: `abc.ABC`

Admin HTTP server class.

add_webhook_target (*target_url: str, topic_filter: Sequence[str] = None, max_attempts: int = None*)

Add a webhook target.

remove_webhook_target (*target_url: str*)

Remove a webhook target.

send_webhook (*profile: aries_cloudagent.core.profile.Profile, topic: str, payload: dict*)

Add a webhook to the queue, to send to all registered targets.

start () → None

Start the webserver.

Raises AdminSetupError – If there was an error starting the webserver

stop () → None
Stop the webserver.

aries_cloudagent.admin.error module

Admin error classes.

exception aries_cloudagent.admin.error.**AdminError** (*args, error_code: str = None, **kwargs)

Bases: *aries_cloudagent.core.error.BaseError*

Base class for Admin-related errors.

exception aries_cloudagent.admin.error.**AdminSetupError** (*args, error_code: str = None, **kwargs)

Bases: *aries_cloudagent.admin.error.AdminError*

Admin server setup or configuration error.

aries_cloudagent.admin.request_context module

Admin request context class.

A request context provided by the admin server to admin route handlers.

class aries_cloudagent.admin.request_context.**AdminRequestContext** (profile: *aries_cloudagent.core.profile.Profile*, *, context: *aries_cloudagent.config.injection_container.InjectionContainer* = None, settings: *Mapping[str, object]* = None)

Bases: *object*

Context established by the Conductor and passed into message handlers.

inject (base_cls: *Type[Inject]*, settings: *Mapping[str, object]* = None, *, required: bool = True) → *Optional[Inject]*
Get the provided instance of a given class identifier.

Parameters

- **cls** – The base class to retrieve an instance of
- **settings** – An optional mapping providing configuration to the provider

Returns An instance of the base class, or None

injector

Accessor for the associated *Injector* instance.

profile

Accessor for the associated *Profile* instance.

session () → *aries_cloudagent.core.profile.ProfileSession*

Start a new interactive session with no transaction support requested.

settings

Accessor for the context settings.


```
classmethod test_context (session_inject: dict = None, profile:  

                           aries_cloudagent.core.profile.Profile = None) →  

                           aries_cloudagent.admin.request_context.AdminRequestContext
```

Quickly set up a new admin request context for tests.

```
transaction () → aries_cloudagent.core.profile.ProfileSession  

    Start a new interactive session with commit and rollback support.
```

If the current backend does not support transactions, then commit and rollback operations of the session will not have any effect.

```
update_settings (settings: Mapping[str, object])  

    Update the current scope with additional settings.
```

aries_cloudagent.admin.server module

Admin server classes.

```
class aries_cloudagent.admin.server.AdminModulesSchema (*args, **kwargs)  

    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
```

Schema for the modules endpoint.

```
result  

    Used by autodoc_mock_imports.
```

```
class aries_cloudagent.admin.server.AdminResponder (profile:  

                                                    aries_cloudagent.core.profile.Profile,  

                                                    send: Coroutine[T_co, T_contra,  

                                                    V_co], webhook: Coroutine[T_co,  

                                                    T_contra, V_co], **kwargs)  

    Bases: aries_cloudagent.messaging.responder.BaseResponder
```

Handle outgoing messages from message handlers.

```
send_fn  

    Accessor for async function to send outbound message.
```

```
send_outbound (message: aries_cloudagent.transport.outbound.message.OutboundMessage)  

    Send outbound message.
```

Parameters **message** – The *OutboundMessage* to be sent

```
send_webhook (topic: str, payload: dict)  

    Dispatch a webhook.
```

Parameters

- **topic** – the webhook topic identifier
- **payload** – the webhook payload value

```
webhook_fn  

    Accessor for the async function to dispatch a webhook.
```

```
class aries_cloudagent.admin.server.AdminServer (host: str, port: int, context:
    aries_cloudagent.config.injection_context.InjectionContext,
    root_profile:
    aries_cloudagent.core.profile.Profile,
    outbound_message_router: Coroutine[T_co, T_contra, V_co], web-
    hook_router: Callable, con-
    ductor_stop: Coroutine[T_co,
    T_contra, V_co], task_queue:
    aries_cloudagent.utils.task_queue.TaskQueue
    = None, conductor_stats: Corou-
    tine[T_co, T_contra, V_co] = None)
```

Bases: `aries_cloudagent.admin.base_server.BaseAdminServer`

Admin HTTP server class.

add_webhook_target (target_url: str, topic_filter: Sequence[str] = None, max_attempts: int = None)
Add a webhook target.

liveliness_handler (request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>)
Request handler for liveliness check.

Parameters request – aiohttp request object

Returns The web response, always indicating True

make_application () → <sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>
Get the aiohttp application instance.

notify_fatal_error ()
Set our readiness flags to force a restart (openshift).

on_startup (app: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>)
Perform webserver startup actions.

plugins_handler (request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>)
Request handler for the loaded plugins list.

Parameters request – aiohttp request object

Returns The module list response

readiness_handler (request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>)
Request handler for liveliness check.

Parameters request – aiohttp request object

Returns The web response, indicating readiness for further calls

redirect_handler (request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>)
Perform redirect to documentation.

remove_webhook_target (target_url: str)
Remove a webhook target.

send_webhook (profile: aries_cloudagent.core.profile.Profile, topic: str, payload: dict)
Add a webhook to the queue, to send to all registered targets.

shutdown_handler (*request*: `<sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>`)
Request handler for server shutdown.

Parameters *request* – aiohttp request object

Returns The web response (empty production)

start () → None
Start the webserver.

Raises `AdminSetupError` – If there was an error starting the webserver

status_handler (*request*: `<sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>`)
Request handler for the server status information.

Parameters *request* – aiohttp request object

Returns The web response

status_reset_handler (*request*: `<sphinx.ext.autodoc.importer._MockObject object at 0x7f760fc402d0>`)
Request handler for resetting the timing statistics.

Parameters *request* – aiohttp request object

Returns The web response

stop () → None
Stop the webserver.

websocket_handler (*request*)
Send notifications to admin client over websocket.

class `aries_cloudagent.admin.server.AdminShutdownSchema` (**args*, ***kwargs*)
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`
Response schema for admin Module.

class `aries_cloudagent.admin.server.AdminStatusLivelinessSchema` (**args*, ***kwargs*)
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`
Schema for the liveliness endpoint.

alive
Used by `autodoc_mock_imports`.

class `aries_cloudagent.admin.server.AdminStatusReadinessSchema` (**args*, ***kwargs*)
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`
Schema for the readiness endpoint.

ready
Used by `autodoc_mock_imports`.

class `aries_cloudagent.admin.server.AdminStatusSchema` (**args*, ***kwargs*)
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`
Schema for the status endpoint.

class `aries_cloudagent.admin.server.WebhookTarget` (*endpoint*: *str*, *topic_filter*: *Sequence[str] = None*, *max_attempts*: *int = None*)
Bases: `object`

Class for managing webhook target information.

topic_filter

Accessor for the target's topic filter.

`aries_cloudagent.admin.server.debug_middleware` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760fc402d0>, handler:
Coroutine[T_co, T_contra, V_co])

Show request detail in debug log.

`aries_cloudagent.admin.server.ready_middleware` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760fc402d0>, handler:
Coroutine[T_co, T_contra, V_co])

Only continue if application is ready to take work.

aries_cloudagent.cache package

Submodules

aries_cloudagent.cache.base module

Abstract base classes for cache.

class `aries_cloudagent.cache.base.BaseCache`

Bases: `abc.ABC`

Abstract cache interface.

acquire (*key: str*)

Acquire a lock on a given cache key.

clear (*key: str*)

Remove an item from the cache, if present.

Parameters **key** – the key to remove

flush ()

Remove all items from the cache.

get (*key: str*)

Get an item from the cache.

Parameters **key** – the key to retrieve an item for

Returns The record found or *None*

release (*key: str*)

Release the lock on a given cache key.

set (*keys: Union[str, Sequence[str]], value: Any, ttl: int = None*)

Add an item to the cache with an optional ttl.

Parameters

- **keys** – the key or keys for which to set an item
- **value** – the value to store in the cache
- **ttl** – number of second that the record should persist

exception `aries_cloudagent.cache.base.CacheError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Base class for cache-related errors.

class `aries_cloudagent.cache.base.CacheKeyLock` (cache: `aries_cloudagent.cache.base.BaseCache`, key: str)

Bases: `object`

A lock on a particular cache key.

Used to prevent multiple async threads from generating or querying the same semi-expensive data. Not thread safe.

done

Accessor for the done state.

future

Fetch the result in the form of an awaitable future.

parent

Accessor for the parent key lock, if any.

release ()

Release the cache lock.

result

Fetch the current result, if any.

set_result (value: Any, ttl: int = None)

Set the result, updating the cache and any waiters.

`aries_cloudagent.cache.in_memory` module

Basic in-memory cache implementation.

class `aries_cloudagent.cache.in_memory.InMemoryCache`

Bases: `aries_cloudagent.cache.base.BaseCache`

Basic in-memory cache class.

clear (key: str)

Remove an item from the cache, if present.

Parameters **key** – the key to remove

flush ()

Remove all items from the cache.

get (key: str)

Get an item from the cache.

Parameters **key** – the key to retrieve an item for

Returns The record found or *None*

set (keys: Union[str, Sequence[str]], value: Any, ttl: int = None)

Add an item to the cache with an optional ttl.

Overwrites existing cache entries.

Parameters

- **keys** – the key or keys for which to set an item
- **value** – the value to store in the cache
- **ttl** – number of seconds that the record should persist

aries_cloudagent.commands package

Commands module common setup.

`aries_cloudagent.commands.available_commands()`
Index available commands.

`aries_cloudagent.commands.load_command(command: str)`
Load the module corresponding with a named command.

`aries_cloudagent.commands.run_command(command: str, argv: Sequence[str] = None)`
Execute a named command with command line arguments.

Submodules

aries_cloudagent.commands.help module

Help command for indexing available commands.

`aries_cloudagent.commands.help.execute(argv: Sequence[str] = None)`
Execute the help command.

`aries_cloudagent.commands.help.main()`
Execute the main line.

aries_cloudagent.commands.provision module

Provision command for setting up agent settings before starting.

exception `aries_cloudagent.commands.provision.ProvisionError(*args, error_code: str = None, **kwargs)`

Bases: `aries_cloudagent.core.error.BaseError`

Base exception for provisioning errors.

`aries_cloudagent.commands.provision.execute(argv: Sequence[str] = None)`
Entrypoint.

`aries_cloudagent.commands.provision.init_argument_parser(parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f362ad0>)`

Initialize an argument parser with the module's arguments.

`aries_cloudagent.commands.provision.main()`
Execute the main line.

`aries_cloudagent.commands.provision.provision(settings: dict)`
Perform provisioning.

aries_cloudagent.commands.start module

Entrypoint.

`aries_cloudagent.commands.start.execute` (*argv: Sequence[str] = None*)

Entrypoint.

`aries_cloudagent.commands.start.init_argument_parser` (*parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f15d650>*)

Initialize an argument parser with the module's arguments.

`aries_cloudagent.commands.start.main` ()

Execute the main line.

`aries_cloudagent.commands.start.run_loop` (*startup: Coroutine[T_co, T_contra, V_co], shutdown: Coroutine[T_co, T_contra, V_co]*)

Execute the application, handling signals and ctrl-c.

`aries_cloudagent.commands.start.shutdown_app` (*conductor: aries_cloudagent.core.conductor.Conductor*)

Shut down.

`aries_cloudagent.commands.start.start_app` (*conductor: aries_cloudagent.core.conductor.Conductor*)

Start up.

aries_cloudagent.config package

Submodules

aries_cloudagent.config.argparse module

Command line option parsing.

class `aries_cloudagent.config.argparse.AdminGroup`

Bases: `aries_cloudagent.config.argparse.ArgumentGroup`

Admin server settings.

CATEGORIES = ('start',)

GROUP_NAME = 'Admin'

add_arguments (*parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>*)

Add admin-specific command line arguments to the parser.

get_settings (*args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>*)

Extract admin settings.

class `aries_cloudagent.config.argparse.ArgumentGroup`

Bases: `abc.ABC`

A class representing a group of related command line arguments.

GROUP_NAME = None

add_arguments (*parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>*)

Add arguments to the provided argument parser.

get_settings (*args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>*) → dict

Extract settings from the parsed arguments.

```
class aries_cloudagent.config.argparse.DebugGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup

    Debug settings.

    CATEGORIES = ('start',)

    GROUP_NAME = 'Debug'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add debug command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>) → dict
        Extract debug settings.

class aries_cloudagent.config.argparse.GeneralGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup

    General settings.

    CATEGORIES = ('general', 'start')

    GROUP_NAME = 'General'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add general command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>) → dict
        Extract general settings.

class aries_cloudagent.config.argparse.LedgerGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup

    Ledger settings.

    CATEGORIES = ('start', 'general')

    GROUP_NAME = 'Ledger'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add ledger-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>) → dict
        Extract ledger settings.

class aries_cloudagent.config.argparse.LoggingGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup

    Logging settings.

    CATEGORIES = ('general', 'start')

    GROUP_NAME = 'Logging'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add logging-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>) → dict
        Extract logging settings.

class aries_cloudagent.config.argparse.MediationGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup

    Mediation settings.

    CATEGORIES = ('start',)
```



```

GROUP_NAME = 'Mediation'

add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
    Add mediation command line arguments to the parser.

get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>)
    Extract mediation settings.

class aries_cloudagent.config.argparse.MultitenantGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup
    Multitenant settings.

    CATEGORIES = ('start',)

    GROUP_NAME = 'Multitenant'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add multitenant-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>)
        Extract multitenant settings.

class aries_cloudagent.config.argparse.ProtocolGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup
    Protocol settings.

    CATEGORIES = ('start',)

    GROUP_NAME = 'Protocol'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add protocol-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>) → dict
        Get protocol settings.

class aries_cloudagent.config.argparse.StartupGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup
    Startup settings.

    CATEGORIES = ('start',)

    GROUP_NAME = 'Start-up'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add startup-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>)
        Extract startup settings.

class aries_cloudagent.config.argparse.TransportGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup
    Transport settings.

    CATEGORIES = ('start',)

    GROUP_NAME = 'Transport'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add transport-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>)
        Extract transport settings.

```

```
class aries_cloudagent.config.argparse.WalletGroup
    Bases: aries_cloudagent.config.argparse.ArgumentGroup

    Wallet settings.

    CATEGORIES = ('general', 'start')

    GROUP_NAME = 'Wallet'

    add_arguments (parser: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f2cb7d0>)
        Add wallet-specific command line arguments to the parser.

    get_settings (args: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f1fd650>) → dict
        Extract wallet settings.

aries_cloudagent.config.argparse.create_argument_parser (*, prog: str = None)
    Create an instance of an arg parser, force yaml format for external config.

class aries_cloudagent.config.argparse.group (*categories)
    Bases: object

    Decorator for registering argument groups.

    classmethod get_registered (category: str = None)
        Fetch the set of registered classes in a category.

aries_cloudagent.config.argparse.load_argument_groups (parser:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760f2cb7d0>,
    *groups)

    Log a set of argument groups into a parser.

    Returns A callable to convert loaded arguments into a settings dictionary
```

aries_cloudagent.config.base module

Configuration base classes.

```
class aries_cloudagent.config.base.BaseInjector
    Bases: abc.ABC

    Base injector class.

    copy () → aries_cloudagent.config.base.BaseInjector
        Produce a copy of the injector instance.

    inject (base_cls: Type[Inject], settings: Mapping[str, object] = None, *, required: bool = True) →
        Optional[Inject]
        Get the provided instance of a given class identifier.

    Parameters
        • cls – The base class to retrieve an instance of
        • settings – An optional mapping providing configuration to the provider

    Returns An instance of the base class, or None

class aries_cloudagent.config.base.BaseProvider
    Bases: abc.ABC

    Base provider class.
```

provide (*settings:* `aries_cloudagent.config.base.BaseSettings`, *injector:* `aries_cloudagent.config.base.BaseInjector`)
Provide the object instance given a config and injector.

class `aries_cloudagent.config.base.BaseSettings`

Bases: `collections.abc.Mapping`, `typing.Generic`

Base settings class.

copy () → `aries_cloudagent.config.base.BaseSettings`
Produce a copy of the settings instance.

extend (*other:* `Mapping[str, object]`) → `aries_cloudagent.config.base.BaseSettings`
Merge another mapping to produce a new settings instance.

get_bool (**var_names*, *default=None*) → `bool`
Fetch a setting as a boolean value.

Parameters

- **var_names** – A list of variable name alternatives
- **default** – The default value to return if none are defined

get_int (**var_names*, *default=None*) → `int`
Fetch a setting as an integer value.

Parameters

- **var_names** – A list of variable name alternatives
- **default** – The default value to return if none are defined

get_str (**var_names*, *default=None*) → `str`
Fetch a setting as a string value.

Parameters

- **var_names** – A list of variable name alternatives
- **default** – The default value to return if none are defined

get_value (**var_names*, *default=None*)
Fetch a setting.

Parameters

- **var_names** – A list of variable name alternatives
- **default** – The default value to return if none are defined

Returns The setting value, if defined, otherwise the default value

exception `aries_cloudagent.config.base.ConfigError` (**args*, *error_code:* `str = None`, ***kwargs*)

Bases: `aries_cloudagent.core.error.BaseError`

A base exception for all configuration errors.

exception `aries_cloudagent.config.base.InjectionError` (**args*, *error_code:* `str = None`, ***kwargs*)

Bases: `aries_cloudagent.config.base.ConfigError`

The base exception raised by Injector and Provider implementations.

exception `aries_cloudagent.config.base.SettingsError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.config.base.ConfigError`

The base exception raised by *BaseSettings* implementations.

`aries_cloudagent.config.base_context` module

Base injection context builder classes.

class `aries_cloudagent.config.base_context.ContextBuilder` (settings: Mapping[str, object] = None)

Bases: `abc.ABC`

Base injection context builder class.

build_context () → `aries_cloudagent.config.injection_context.InjectionContext`

Build the base injection context.

update_settings (settings: Mapping[str, object])

Update the context builder with additional settings.

`aries_cloudagent.config.default_context` module

Classes for configuring the default injection context.

class `aries_cloudagent.config.default_context.DefaultContextBuilder` (settings: Mapping[str, object] = None)

Bases: `aries_cloudagent.config.base_context.ContextBuilder`

Default context builder.

bind_providers (context: `aries_cloudagent.config.injection_context.InjectionContext`)

Bind various class providers.

build_context () → `aries_cloudagent.config.injection_context.InjectionContext`

Build the base injection context; set DIDComm prefix to emit.

load_plugins (context: `aries_cloudagent.config.injection_context.InjectionContext`)

Set up plugin registry and load plugins.

`aries_cloudagent.config.error` module

Errors for config modules.

exception `aries_cloudagent.config.error.ArgsParseError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.config.base.ConfigError`

Error raised when there is a problem parsing the command-line arguments.

aries_cloudagent.config.injection_context module

Injection context implementation.

```
class aries_cloudagent.config.injection_context.InjectionContext (*, settings:
                                                                    Mapping[str,
                                                                    object] =
                                                                    None, en-
                                                                    force_typing:
                                                                    bool = True)
```

Bases: *aries_cloudagent.config.base.BaseInjector*

Manager for configuration settings and class providers.

ROOT_SCOPE = 'application'

copy () → aries_cloudagent.config.injection_context.InjectionContext
Produce a copy of the injector instance.

inject (base_cls: Type[Inject], settings: Mapping[str, object] = None, *, required: bool = True) → Optional[Inject]
Get the provided instance of a given class identifier.

Parameters

- **cls** – The base class to retrieve an instance of
- **settings** – An optional mapping providing configuration to the provider

Returns An instance of the base class, or None

injector
Accessor for scope-specific injector.

injector_for_scope (scope_name: str) → aries_cloudagent.config.injector.Injector
Fetch the injector for a specific scope.

Parameters **scope_name** – The unique scope identifier

scope_name
Accessor for the current scope name.

settings
Accessor for scope-specific settings.

start_scope (scope_name: str, settings: Mapping[str, object] = None) → aries_cloudagent.config.injection_context.InjectionContext
Begin a new named scope.

Parameters

- **scope_name** – The unique name for the scope being entered
- **settings** – An optional mapping of additional settings to apply

Returns A new injection context representing the scope

update_settings (settings: Mapping[str, object])
Update the scope with additional settings.

```
exception aries_cloudagent.config.injection_context.InjectionContextError (*args,
                                                                    er-
                                                                    ror_code:
                                                                    str
                                                                    =
                                                                    None,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.config.base.InjectionError*

Base class for issues in the injection context.

```
class aries_cloudagent.config.injection_context.Scope (name, injector)
```

Bases: *tuple*

injector

Alias for field number 1

name

Alias for field number 0

aries_cloudagent.config.injector module

Standard Injector implementation.

```
class aries_cloudagent.config.injector.Injector (settings: Mapping[str, object] = None,
                                                *, enforce_typing: bool = True)
```

Bases: *aries_cloudagent.config.base.BaseInjector*

Injector implementation with static and dynamic bindings.

```
bind_instance (base_cls: Type[Inject], instance: Inject)
```

Add a static instance as a class binding.

```
bind_provider (base_cls: Type[Inject], provider: aries_cloudagent.config.base.BaseProvider, *,
               cache: bool = False)
```

Add a dynamic instance resolver as a class binding.

```
clear_binding (base_cls: Type[Inject])
```

Remove a previously-added binding.

```
copy () → aries_cloudagent.config.base.BaseInjector
```

Produce a copy of the injector instance.

```
get_provider (base_cls: Type[Inject])
```

Find the provider associated with a class binding.

```
inject (base_cls: Type[Inject], settings: Mapping[str, object] = None, *, required: bool = True) →
Optional[Inject]
```

Get the provided instance of a given class identifier.

Parameters

- **cls** – The base class to retrieve an instance of
- **params** – An optional dict providing configuration to the provider

Returns An instance of the base class, or None

settings

Accessor for scope-specific settings.

aries_cloudagent.config.ledger module

Ledger configuration.

`aries_cloudagent.config.ledger.accept_taa` (*ledger: aries_cloudagent.ledger.base.BaseLedger, taa_info, provision: bool = False*) → bool

Perform TAA acceptance.

`aries_cloudagent.config.ledger.fetch_genesis_transactions` (*genesis_url: str*) → str

Get genesis transactions.

`aries_cloudagent.config.ledger.get_genesis_transactions` (*settings: aries_cloudagent.config.settings.Settings*) → str

Fetch genesis transactions if necessary.

`aries_cloudagent.config.ledger.ledger_config` (*profile: aries_cloudagent.core.profile.Profile, public_did: str, provision: bool = False*) → bool

Perform Indy ledger configuration.

aries_cloudagent.config.logging module

Utilities related to logging.

class `aries_cloudagent.config.logging.LoggingConfigurator`

Bases: `object`

Utility class used to configure logging and print an informative start banner.

classmethod `configure` (*logging_config_path: str = None, log_level: str = None, log_file: str = None*)

Configure logger.

Parameters

- **logging_config_path** – str: (Default value = None) Optional path to custom logging config
- **log_level** – str: (Default value = None)

classmethod `print_banner` (*agent_label, inbound_transports, outbound_transports, public_did, admin_server=None, banner_length=40, border_character=':'*)

Print a startup banner describing the configuration.

Parameters

- **agent_label** – Agent Label
- **inbound_transports** – Configured inbound transports
- **outbound_transports** – Configured outbound transports
- **admin_server** – Admin server info
- **public_did** – Public DID
- **banner_length** – (Default value = 40) Length of the banner
- **border_character** – (Default value = “:”) Character to use in banner
- **border** –

`aries_cloudagent.config.logging.load_resource` (*path: str, encoding: str = None*) → `TextIO`
 Open a resource file located in a python package or the local filesystem.

Parameters *path* – The resource path in the form of *dir/file* or *package:dir/file*

Returns A file-like object representing the resource

aries_cloudagent.config.provider module

Service provider implementations.

class `aries_cloudagent.config.provider.CachedProvider` (*provider:*
`aries_cloudagent.config.base.BaseProvider,`
unique_settings_keys: tuple =
`()`)

Bases: `aries_cloudagent.config.base.BaseProvider`

Cache the result of another provider.

provide (*config:* `aries_cloudagent.config.base.BaseSettings,` *injector:*
`aries_cloudagent.config.base.BaseInjector`)
 Provide the object instance given a config and injector.

Instances are cached keyed on a SHA256 digest of the relevant subset of settings.

class `aries_cloudagent.config.provider.ClassProvider` (*instance_cls: Union[str, type],*
**ctor_args, init_method: str =*
`None, **ctor_kwargs`)

Bases: `aries_cloudagent.config.base.BaseProvider`

Provider for a particular class.

class `Inject` (*base_cls: type*)
 Bases: `object`

A class for passing injected arguments to the constructor.

provide (*config:* `aries_cloudagent.config.base.BaseSettings,` *injector:*
`aries_cloudagent.config.base.BaseInjector`)
 Provide the object instance given a config and injector.

class `aries_cloudagent.config.provider.InstanceProvider` (*instance*)
 Bases: `aries_cloudagent.config.base.BaseProvider`

Provider for a previously-created instance.

provide (*config:* `aries_cloudagent.config.base.BaseSettings,` *injector:*
`aries_cloudagent.config.base.BaseInjector`)
 Provide the object instance given a config and injector.

class `aries_cloudagent.config.provider.StatsProvider` (*provider:*
`aries_cloudagent.config.base.BaseProvider,`
*methods: Sequence[str], *, ig-*
nore_missing: bool = True)

Bases: `aries_cloudagent.config.base.BaseProvider`

Add statistics to the results of another provider.

provide (*config:* `aries_cloudagent.config.base.BaseSettings,` *injector:*
`aries_cloudagent.config.base.BaseInjector`)
 Provide the object instance given a config and injector.

aries_cloudagent.config.settings module

Settings implementation.

class aries_cloudagent.config.settings.**Settings** (*values: Mapping[str, object] = None*)
Bases: *aries_cloudagent.config.base.BaseSettings*

Mutable settings implementation.

clear_value (*var_name: str*)
Remove a setting.

Parameters **var_name** – The name of the setting

copy () → aries_cloudagent.config.base.BaseSettings
Produce a copy of the settings instance.

extend (*other: Mapping[str, object]*) → aries_cloudagent.config.base.BaseSettings
Merge another settings instance to produce a new instance.

get_value (**var_names, default=None*)
Fetch a setting.

Parameters

- **var_names** – A list of variable name alternatives
- **default** – The default value to return if none are defined

set_default (*var_name: str, value*)
Add a setting if not currently defined.

Parameters

- **var_name** – The name of the setting
- **value** – The value to assign

set_value (*var_name: str, value*)
Add a setting.

Parameters

- **var_name** – The name of the setting
- **value** – The value to assign

update (*other: Mapping[str, object]*)
Update the settings in place.

aries_cloudagent.config.util module

Entrypoint.

class aries_cloudagent.config.util.**ByteSize** (*min_size: int = 0, max_size: int = 0*)
Bases: *object*

Argument value parser for byte sizes.

aries_cloudagent.config.util.**common_config** (*settings: Mapping[str, Any]*)
Perform common app configuration.

aries_cloudagent.config.wallet module

Wallet configuration.

```
aries_cloudagent.config.wallet.wallet_config (context: aries_cloudagent.config.injection_context.InjectionContext,
                                              provision: bool = False) -> (<class
                                              'aries_cloudagent.core.profile.Profile'>,
                                              <class 'aries_cloudagent.wallet.base.DIDInfo'>)
```

Initialize the root profile.

aries_cloudagent.connections package

Subpackages

aries_cloudagent.connections.models package

Subpackages

aries_cloudagent.connections.models.diddoc package

DID Document model support.

Copyright 2017-2019 Government of Canada Public Services and Procurement Canada - buyandsell.gc.ca

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
class aries_cloudagent.connections.models.diddoc.DIDDoc (did: str = None)
```

Bases: `object`

DID document, grouping a DID with verification keys and services.

Retains DIDs as raw values (orientated toward indy-facing operations), everything else as URIs (oriented toward W3C-facing operations).

```
CONTEXT = 'https://w3id.org/did/v1'
```

```
add_service_pubkeys (service: dict, tags: Union[Sequence[str], str]) →
```

List[aries_cloudagent.connections.models.diddoc.publickey.PublicKey]

Add public keys specified in service. Return public keys so discovered.

Parameters

- **service** – service from DID document
- **tags** – potential tags marking public keys of type of interest (the standard is still coalescing)

Raises `ValueError` – for public key reference not present in DID document.

Returns: list of public keys from the document service specification

authnkey

Accessor for public keys marked as authentication keys, by identifier.

classmethod deserialize (*did_doc: dict*) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Construct DIDDoc object from dict representation.

Parameters *did_doc* – DIDDoc dict representation

Raises `ValueError` – for bad DID or missing mandatory item.

Returns: DIDDoc from input json

did

Accessor for DID.

classmethod from_json (*did_doc_json: str*) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Construct DIDDoc object from json representation.

Parameters *did_doc_json* – DIDDoc json representation

Returns: DIDDoc from input json

pubkey

Accessor for public keys by identifier.

serialize () → dict

Dump current object to a JSON-compatible dictionary.

Returns dict representation of current DIDDoc

service

Accessor for services by identifier.

set (*item: Union[aries_cloudagent.connections.models.diddoc.service.Service, aries_cloudagent.connections.models.diddoc.publickey.PublicKey]*) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Add or replace service or public key; return current DIDDoc.

Raises `ValueError` – if input item is neither service nor public key.

Parameters *item* – service or public key to set

Returns: the current DIDDoc

to_json () → str

Dump current object as json (JSON-LD).

Returns json representation of current DIDDoc

class aries_cloudagent.connections.models.diddoc.**LinkedDataKeySpec** (*ver_type, au-
thn_type, specifier*)

Bases: `tuple`

authn_type

Alias for field number 1

specifier

Alias for field number 2

ver_type

Alias for field number 0

```
class aries_cloudagent.connections.models.diddoc.PublicKey(did: str, ident: str,  
                                                         value: str, pk_type:  
                                                         aries_cloudagent.connections.models.diddoc.p  
                                                         = None, controller:  
                                                         str = None, authn:  
                                                         bool = False)
```

Bases: `object`

Public key specification to embed in DID document.

Retains DIDs as raw values (orientated toward indy-facing operations), everything else as URIs (oriented toward W3C-facing operations).

authn

Accessor for the authentication marker.

Returns: whether public key is marked as having DID authentication privilege

controller

Accessor for the controller DID.

did

Accessor for the DID.

id

Accessor for the public key identifier.

to_dict () → dict

Return dict representation of public key to embed in DID document.

type

Accessor for the public key type.

value

Accessor for the public key value.

```
class aries_cloudagent.connections.models.diddoc.PublicKeyType
```

Bases: `enum.Enum`

Class encapsulating public key types.

```
ED25519_SIG_2018 = LinkedDataKeySpec(ver_type='Ed25519VerificationKey2018', authn_type
```

```
EDDSA_SA_SIG_SECP256K1 = LinkedDataKeySpec(ver_type='Secp256k1VerificationKey2018', au
```

```
RSA_SIG_2018 = LinkedDataKeySpec(ver_type='RsaVerificationKey2018', authn_type='RsaSig
```

authn_type

Accessor for the authentication type identifier.

```
get = <function PublicKeyType.get>
```

```
specification (val: str) → str
```

Return specifier and input value for use in public key specification.

Parameters **val** – value of public key

Returns: dict mapping applicable specifier to input value

specifier

Accessor for the value specifier.

ver_type

Accessor for the verification type identifier.

```

class aries_cloudagent.connections.models.diddoc.Service (did: str, ident: str,
                                                         typ: str, recip_keys:
                                                         Union[Sequence[T_co],
                                                         aries_cloudagent.connections.models.diddoc.publ
                                                         routing_keys:
                                                         Union[Sequence[T_co],
                                                         aries_cloudagent.connections.models.diddoc.publ
                                                         endpoint: str, priority:
                                                         int = 0)

```

Bases: `object`

Service specification to embed in DID document.

Retains DIDs as raw values (orientated toward indy-facing operations), everything else as URIs (oriented toward W3C-facing operations).

did

Accessor for the DID value.

endpoint

Accessor for the endpoint value.

id

Accessor for the service identifier.

priority

Accessor for the priority value.

recip_keys

Accessor for the recipient keys.

routing_keys

Accessor for the routing keys.

to_dict () → dict

Return dict representation of service to embed in DID document.

type

Accessor for the service type.

Submodules

aries_cloudagent.connections.models.diddoc.diddoc module

DID Document classes.

Copyright 2017-2019 Government of Canada Public Services and Procurement Canada - buyandsell.gc.ca

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```

class aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc (did: str =
                                                                None)

```

Bases: `object`

DID document, grouping a DID with verification keys and services.

Retains DIDs as raw values (orientated toward indy-facing operations), everything else as URIs (oriented toward W3C-facing operations).

CONTEXT = 'https://w3id.org/did/v1'

add_service_pubkeys (*service:* dict, *tags:* Union[Sequence[str], str]) → List[aries_cloudagent.connections.models.diddoc.publickey.PublicKey]
Add public keys specified in service. Return public keys so discovered.

Parameters

- **service** – service from DID document
- **tags** – potential tags marking public keys of type of interest (the standard is still coalescing)

Raises `ValueError` – for public key reference not present in DID document.

Returns: list of public keys from the document service specification

authnkey

Accessor for public keys marked as authentication keys, by identifier.

classmethod deserialize (*did_doc:* dict) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Construct DIDDoc object from dict representation.

Parameters **did_doc** – DIDDoc dict representation

Raises `ValueError` – for bad DID or missing mandatory item.

Returns: DIDDoc from input json

did

Accessor for DID.

classmethod from_json (*did_doc_json:* str) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Construct DIDDoc object from json representation.

Parameters **did_doc_json** – DIDDoc json representation

Returns: DIDDoc from input json

pubkey

Accessor for public keys by identifier.

serialize () → dict
Dump current object to a JSON-compatible dictionary.

Returns dict representation of current DIDDoc

service

Accessor for services by identifier.

set (*item:* Union[aries_cloudagent.connections.models.diddoc.service.Service, aries_cloudagent.connections.models.diddoc.publickey.PublicKey]) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Add or replace service or public key; return current DIDDoc.

Raises `ValueError` – if input item is neither service nor public key.

Parameters **item** – service or public key to set

Returns: the current DIDDoc

to_json() → str

Dump current object as json (JSON-LD).

Returns json representation of current DIDDoc

aries_cloudagent.connections.models.diddoc.publickey module

DID Document Public Key classes.

Copyright 2017-2019 Government of Canada Public Services and Procurement Canada - buyandsell.gc.ca

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
class aries_cloudagent.connections.models.diddoc.publickey.LinkedDataKeySpec(ver_type,
                                                                              au-
                                                                              thn_type,
                                                                              spec-
                                                                              i-
                                                                              fier)
```

Bases: tuple

authn_type

Alias for field number 1

specifier

Alias for field number 2

ver_type

Alias for field number 0

```
class aries_cloudagent.connections.models.diddoc.publickey.PublicKey(did:
                                                                              str,
                                                                              ident:
                                                                              str,
                                                                              value:
                                                                              str,
                                                                              pk_type:
                                                                              aries_cloudagent.connections.m
                                                                              =
                                                                              None,
                                                                              con-
                                                                              troller:
                                                                              str =
                                                                              None,
                                                                              authn:
                                                                              bool =
                                                                              False)
```

Bases: object

Public key specification to embed in DID document.

Retains DIDs as raw values (orientated toward indy-facing operations), everything else as URIs (oriented toward W3C-facing operations).

authn

Accessor for the authentication marker.

Returns: whether public key is marked as having DID authentication privilege

controller

Accessor for the controller DID.

did

Accessor for the DID.

id

Accessor for the public key identifier.

to_dict () → dict

Return dict representation of public key to embed in DID document.

type

Accessor for the public key type.

value

Accessor for the public key value.

class aries_cloudagent.connections.models.diddoc.publickey.**PublicKeyType**

Bases: `enum.Enum`

Class encapsulating public key types.

ED25519_SIG_2018 = `LinkedDataKeySpec(ver_type='Ed25519VerificationKey2018', authn_type=`

`EDDSA_SA_SIG_SECP256K1` = `LinkedDataKeySpec(ver_type='Secp256k1VerificationKey2018', au`

`RSA_SIG_2018` = `LinkedDataKeySpec(ver_type='RsaVerificationKey2018', authn_type='RsaSig`

authn_type

Accessor for the authentication type identifier.

get = `<function PublicKeyType.get>`

specification (*val: str*) → str

Return specifier and input value for use in public key specification.

Parameters **val** – value of public key

Returns: dict mapping applicable specifier to input value

specifier

Accessor for the value specifier.

ver_type

Accessor for the verification type identifier.

aries_cloudagent.connections.models.diddoc.service module

DID Document Service classes.

Copyright 2017-2019 Government of Canada Public Services and Procurement Canada - buyandsell.gc.ca

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
class aries_cloudagent.connections.models.diddoc.service.Service (did: str;  
ident: str;  
typ: str;  
recip_keys:  
Union[Sequence[T_co],  
aries_cloudagent.connections.models.  
routing_keys:  
Union[Sequence[T_co],  
aries_cloudagent.connections.models.  
endpoint: str;  
priority: int  
= 0)
```

Bases: `object`

Service specification to embed in DID document.

Retains DIDs as raw values (orientated toward indy-facing operations), everything else as URIs (oriented toward W3C-facing operations).

did

Accessor for the DID value.

endpoint

Accessor for the endpoint value.

id

Accessor for the service identifier.

priority

Accessor for the priority value.

recip_keys

Accessor for the recipient keys.

routing_keys

Accessor for the routing keys.

to_dict () → dict

Return dict representation of service to embed in DID document.

type

Accessor for the service type.

aries_cloudagent.connections.models.diddoc.util module

DIDDoc utility methods.

Copyright 2017-2019 Government of Canada Public Services and Procurement Canada - buyandsell.gc.ca

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

`aries_cloudagent.connections.models.diddoc.util.canon_did(uri: str) → str`

Convert a URI into a DID if need be, left-stripping ‘did:sov:’ if present.

Parameters `uri` – input URI or DID

Raises `ValueError` – for invalid input.

`aries_cloudagent.connections.models.diddoc.util.canon_ref(did: str, ref: str, delimiter: str = None)`

Given a reference in a DID document, return it in its canonical form of a URI.

Parameters

- **did** – DID acting as the identifier of the DID document
- **ref** – reference to canonicalize, either a DID or a fragment pointing to a location in the DID doc
- **delimiter** – delimiter character marking fragment (default ‘#’) or introducing identifier (‘;’) against DID resource

`aries_cloudagent.connections.models.diddoc.util.ok_did(token: str) → bool`

Whether input token looks like a valid decentralized identifier.

Parameters `token` – candidate string

Returns: whether input token looks like a valid schema identifier

`aries_cloudagent.connections.models.diddoc.util.resource(ref: str, delimiter: str = None) → str`

Extract the resource for an identifier.

Given a (URI) reference, return up to its delimiter (exclusively), or all of it if there is none.

Parameters

- **ref** – reference
- **delimiter** – delimiter character (default None maps to ‘#’, or ‘;’ introduces identifiers)

Submodules

`aries_cloudagent.connections.models.conn_record` module

Handle connection information interface with non-secrets storage.

```

class aries_cloudagent.connections.models.conn_record.ConnRecord(*,
    connection_id: str
    = None,
    my_did: str
    = None,
    their_did:
    str = None,
    their_label:
    str = None,
    their_role:
    Union[str,
    ConnRecord.Role]
    = None,
    invitation_key:
    str = None,
    invitation_msg_id:
    str = None,
    request_id:
    str = None,
    state:
    Union[str,
    ConnRecord.State]
    = None,
    inbound_connection_id:
    str =
    None,
    error_msg: str
    = None,
    routing_state:
    str = None,
    accept: str =
    None,
    invitation_mode:
    str = None,
    alias: str
    = None,
    their_public_did:
    str = None,
    rfc23_state=None,
    initiator: str
    = None,
    **kwargs)

```

Bases: `aries_cloudagent.messaging.models.base_record.BaseRecord`

Represents a single pairwise connection.

ACCEPT_AUTO = 'auto'

ACCEPT_MANUAL = 'manual'

INVITATION_MODE_MULTI = 'multi'

INVITATION_MODE_ONCE = 'once'

```
INVITATION_MODE_STATIC = 'static'
LOG_STATE_FLAG = 'debug.connections'

class Meta
    Bases: object
    ConnRecord metadata.
    schema_class = 'ConnRecordSchema'
RECORD_ID_NAME = 'connection_id'
RECORD_TYPE = 'connection'
RECORD_TYPE_INVITATION = 'connection_invitation'
RECORD_TYPE_METADATA = 'connection_metadata'
RECORD_TYPE_REQUEST = 'connection_request'
ROUTING_STATE_ACTIVE = 'active'
ROUTING_STATE_ERROR = 'error'
ROUTING_STATE_NONE = 'none'
ROUTING_STATE_REQUEST = 'request'

class Role
    Bases: enum.Enum
    RFC 160 (inviter, invitee) = RFC 23 (responder, requester).
    REQUESTER = ('invitee', 'requester')
    RESPONDER = ('inviter', 'responder')
    flip()
        Return opposite interlocutor role: theirs for ours, ours for theirs.
    get = <bound method ConnRecord.Role.get of <enum 'Role'>>
    rfc160
        Return RFC 160 (connection protocol) nomenclature.
    rfc23
        Return RFC 23 (DID exchange protocol) nomenclature.

class State
    Bases: enum.Enum
    Collator for equivalent states between RFC 160 and RFC 23.
    On the connection record, the state has to serve for both RFCs. Hence, internally, RFC23 requester/responder states collate to their RFC160 condensed equivalent.
    ABANDONED = ('error', 'abandoned')
    COMPLETED = ('active', 'completed')
    INIT = ('init', 'start')
    INVITATION = ('invitation', 'invitation')
    REQUEST = ('request', 'request')
    RESPONSE = ('response', 'response')
```

```

get = <bound method ConnRecord.State.get of <enum 'State'>>

rfc160
    Return RFC 160 (connection protocol) nomenclature.

rfc23
    Return RFC 23 (DID exchange protocol) nomenclature to record logic.

rfc23strict (their_role: aries_cloudagent.connections.models.conn_record.ConnRecord.Role)
    Return RFC 23 (DID exchange protocol) nomenclature to role as per RFC.

TAG_NAMES = {'invitation_key', 'my_did', 'request_id', 'their_did'}

WEBHOOK_TOPIC = 'connections'

attach_invitation (session: aries_cloudagent.core.profile.ProfileSession, invitation:
    Union[aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation,
    aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationMessage])
    Persist the related connection invitation to storage.

    Parameters
        • session – The active profile session
        • invitation – The invitation to relate to this connection record

attach_request (session: aries_cloudagent.core.profile.ProfileSession, request:
    Union[aries_cloudagent.protocols.connections.v1_0.messages.connection_request.ConnectionRequest,
    aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequest])
    Persist the related connection request to storage.

    Parameters
        • session – The active profile session
        • request – The request to relate to this connection record

connection_id
    Accessor for the ID associated with this connection.

is_multiuse_invitation
    Accessor for multi use invitation mode.

is_ready
    Accessor for connection readiness.

metadata_delete (session: aries_cloudagent.core.profile.ProfileSession, key: str)
    Delete custom metadata associated with this connection.

    Parameters
        • session (ProfileSession) – session used for storage
        • key (str) – key of metadata to delete

metadata_get (session: aries_cloudagent.core.profile.ProfileSession, key: str, default: Any = None)
    → Any
    Retrieve arbitrary metadata associated with this connection.

    Parameters
        • session (ProfileSession) – session used for storage
        • key (str) – key identifying metadata
        • default (Any) – default value to get; type should be a JSON

```

- **value.** (*compatible*) –

Returns metadata stored by key

Return type Any

metadata_get_all (*session: aries_cloudagent.core.profile.ProfileSession*) → dict

Return all custom metadata associated with this connection.

Parameters **session** (*ProfileSession*) – session used for storage

Returns dictionary representation of all metadata values

Return type dict

metadata_set (*session: aries_cloudagent.core.profile.ProfileSession, key: str, value: Any*)

Set arbitrary metadata associated with this connection.

Parameters

- **session** (*ProfileSession*) – session used for storage
- **key** (*str*) – key identifying metadata
- **value** (*Any*) – value to set

post_save (*session: aries_cloudagent.core.profile.ProfileSession, *args, **kwargs*)

Perform post-save actions.

Parameters **session** – The active profile session

record_value

Accessor to for the JSON record value properties for this connection.

classmethod retrieve_by_did (*session: aries_cloudagent.core.profile.ProfileSession, their_did: str = None, my_did: str = None, their_role: str = None*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Retrieve a connection record by target DID.

Parameters

- **session** – The active profile session
- **their_did** – The target DID to filter by
- **my_did** – One of our DIDs to filter by
- **my_role** – Filter connections by their role

classmethod retrieve_by_invitation_key (*session: aries_cloudagent.core.profile.ProfileSession, invitation_key: str, their_role: str = None*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Retrieve a connection record by invitation key.

Parameters

- **session** – The active profile session
- **invitation_key** – The key on the originating invitation
- **initiator** – Filter by the initiator value

classmethod retrieve_by_request_id (*session: aries_cloudagent.core.profile.ProfileSession, request_id: str*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Retrieve a connection record from our previous request ID.

Parameters

- **session** – The active profile session
- **request_id** – The ID of the originating connection request

retrieve_invitation (*session*: *aries_cloudagent.core.profile.ProfileSession*) →
 Union[aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation, aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationMessage]
 Retrieve the related connection invitation.

Parameters session – The active profile session

retrieve_request (*session*: *aries_cloudagent.core.profile.ProfileSession*) →
 Union[aries_cloudagent.protocols.connections.v1_0.messages.connection_request.ConnectionRequest, aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequest]
 Retrieve the related connection invitation.

Parameters session – The active profile session

rfc23_state
 RFC23 state per RFC text, potentially particular to role.

class aries_cloudagent.connections.models.conn_record.ConnRecordSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

Schema to allow serialization/deserialization of connection records.

class Meta

Bases: *object*

ConnRecordSchema metadata.

model_class

alias of *ConnRecord*

```
accept = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf
alias = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
connection_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
error_msg = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
inbound_connection_id = <fields.String(default=<marshmallow.missing>, attribute=None,
invitation_key = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
invitation_mode = <fields.String(default=<marshmallow.missing>, attribute=None, valida
invitation_msg_id = <fields.String(default=<marshmallow.missing>, attribute=None, vali
my_did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries
request_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=No
rfc23_state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=N
routing_state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
their_did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
their_label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=N
their_public_did = <fields.String(default=<marshmallow.missing>, attribute=None, valid
their_role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<O
```

aries_cloudagent.connections.models.connection_target module

Record used to handle routing of messages to another agent.

```
class aries_cloudagent.connections.models.connection_target.ConnectionTarget (*,
    did:
        str
    =
    None,
    end-
    point:
        str
    =
    None,
    la-
    bel:
        str
    =
    None,
    re-
    cip-
    i-
    ent_keys:
        Se-
        quence[str]
    =
    None,
    rout-
    ing_keys:
        Se-
        quence[str]
    =
    None,
    sender_key:
        str
    =
    None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Record used to handle routing of messages to another agent.

```
class Meta
    Bases: object
    ConnectionTarget metadata.
    schema_class = 'ConnectionTargetSchema'
```

```
class aries_cloudagent.connections.models.connection_target.ConnectionTargetSchema (*args,
    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

ConnectionTarget schema.

```
class Meta
    Bases: object
    ConnectionTargetSchema metadata.
```


model_class
alias of *ConnectionTarget*

unknown
Used by autodoc_mock_imports.

did
Used by autodoc_mock_imports.

endpoint
Used by autodoc_mock_imports.

label
Used by autodoc_mock_imports.

recipient_keys
Used by autodoc_mock_imports.

routing_keys
Used by autodoc_mock_imports.

sender_key
Used by autodoc_mock_imports.

Submodules

aries_cloudagent.connections.base_manager module

Class to provide some common utilities.

For Connection, DIDExchange and OutOfBand Manager.

class aries_cloudagent.connections.base_manager.**BaseConnectionManager** (*session:*
aries_cloudagent.core.profile.I

Bases: *object*

Class to provide utilities regarding connection_targets.

RECORD_TYPE_DID_DOC = 'did_doc'

RECORD_TYPE_DID_KEY = 'did_key'

add_key_for_did (*did: str, key: str*)
Store a verkey for lookup against a DID.

Parameters

- **did** – The DID to associate with this key
- **key** – The verkey to be added

create_did_document (*did_info: aries_cloudagent.wallet.base.DIDInfo, inbound_connection_id:*
str = None, svc_endpoints: Sequence[str] = None, mediation_records:
List[aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRe
= None) → aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc
Create our DID doc for a given DID.

Parameters

- **did_info** – The DID information (DID and verkey) used in the connection
- **inbound_connection_id** – The ID of the inbound routing connection to use

- **svc_endpoints** – Custom endpoints for the DID Document
- **mediation_record** – The record for mediation that contains routing_keys and service endpoint

Returns The prepared *DIDDoc* instance

diddoc_connection_targets (*doc: aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc*,
sender_verkey: str, their_label: str = None) → Sequence[aries_cloudagent.connections.models.connection_target.ConnectionTarget]
 Get a list of connection targets from a DID Document.

Parameters

- **doc** – The DID Document to create the target from
- **sender_verkey** – The verkey we are using
- **their_label** – The connection label they are using

fetch_connection_targets (*connection: aries_cloudagent.connections.models.conn_record.ConnRecord*)
 → Sequence[aries_cloudagent.connections.models.connection_target.ConnectionTarget]
 Get a list of connection targets from a *ConnRecord*.

Parameters connection – The connection record (with associated *DIDDoc*) used to generate the connection target

fetch_did_document (*did: str*) → Tuple[aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc,
 aries_cloudagent.storage.record.StorageRecord]
 Retrieve a DID Document for a given DID.

Parameters did – The DID to search for

find_did_for_key (*key: str*) → str
 Find the DID previously associated with a key.

Parameters key – The verkey to look up

remove_keys_for_did (*did: str*)
 Remove all keys associated with a DID.

Parameters did – The DID for which to remove keys

store_did_document (*did_doc: aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc*)
 Store a DID document.

Parameters did_doc – The *DIDDoc* instance to persist

exception aries_cloudagent.connections.base_manager.**BaseConnectionManagerError** (*args,
 error_code:
 str
 =
 None,
 **kwargs)

Bases: *aries_cloudagent.core.error.BaseError*

BaseConnectionManager error.

aries_cloudagent.connections.util module

Class for providing base utilities for Mediator support.

```
aries_cloudagent.connections.util.mediation_record_if_id(session:
                                                         aries_cloudagent.core.profile.ProfileSession,
                                                         mediation_id: str =
                                                         None, or_default: bool =
                                                         False)
```

Validate mediation and return record.

If mediation_id is not None, validate mediation record state and return record else, return None

aries_cloudagent.core package

Submodules

aries_cloudagent.core.conductor module

The Conductor.

The conductor is responsible for coordinating messages that are received over the network, communicating with the ledger, passing messages to handlers, instantiating concrete implementations of required modules and storing data in the wallet.

```
class aries_cloudagent.core.conductor.Conductor (context_builder:
                                                         aries_cloudagent.config.base_context.ContextBuilder)
```

Bases: `object`

Conductor class.

Class responsible for initializing concrete implementations of our require interfaces and routing inbound and outbound message data.

context

Accessor for the injection context.

```
dispatch_complete (message: aries_cloudagent.transport.inbound.message.InboundMessage,
                      completed: aries_cloudagent.utils.task_queue.CompletedTask)
```

Handle completion of message dispatch.

```
get_stats () → dict
```

Get the current stats tracked by the conductor.

```
handle_not_delivered (profile: aries_cloudagent.core.profile.Profile, outbound:
                        aries_cloudagent.transport.outbound.message.OutboundMessage)
```

Handle a message that failed delivery via outbound transports.

```
handle_not_returned (profile: aries_cloudagent.core.profile.Profile, outbound:
                      aries_cloudagent.transport.outbound.message.OutboundMessage)
```

Handle a message that failed delivery via an inbound session.

```
inbound_message_router (profile: aries_cloudagent.core.profile.Profile, message:
                          aries_cloudagent.transport.inbound.message.InboundMessage,
                          can_respond: bool = False)
```

Route inbound messages.

Parameters

- **context** – The context associated with the inbound message
- **message** – The inbound message instance
- **can_respond** – If the session supports return routing

outbound_message_router (*profile: aries_cloudagent.core.profile.Profile, outbound: aries_cloudagent.transport.outbound.message.OutboundMessage, inbound: aries_cloudagent.transport.inbound.message.InboundMessage = None*) → None

Route an outbound message.

Parameters

- **profile** – The active profile for the request
- **message** – An outbound message to be sent
- **inbound** – The inbound message that produced this response, if available

queue_outbound (*profile: aries_cloudagent.core.profile.Profile, outbound: aries_cloudagent.transport.outbound.message.OutboundMessage, inbound: aries_cloudagent.transport.inbound.message.InboundMessage = None*)

Queue an outbound message.

Parameters

- **profile** – The active profile
- **message** – An outbound message to be sent
- **inbound** – The inbound message that produced this response, if available

setup ()

Initialize the global request context.

start () → None

Start the agent.

stop (*timeout=1.0*)

Stop the agent.

webhook_router (*topic: str, payload: dict, endpoint: str, max_attempts: int = None, metadata: dict = None*)

Route a webhook through the outbound transport manager.

Parameters

- **topic** – The webhook topic
- **payload** – The webhook payload
- **endpoint** – The endpoint of the webhook target
- **max_attempts** – The maximum number of attempts
- **metadata** – Additional metadata associated with the payload

aries_cloudagent.core.dispatcher module

The Dispatcher.

The dispatcher is responsible for coordinating data flow between handlers, providing lifecycle hook callbacks storing state for message threads, etc.

class `aries_cloudagent.core.dispatcher.Dispatcher` (*profile: aries_cloudagent.core.profile.Profile*)

Bases: `object`

Dispatcher class.

Class responsible for dispatching messages to message handlers and responding to other agents.

complete (*timeout: float = 0.1*)

Wait for pending tasks to complete.

handle_message (*profile: aries_cloudagent.core.profile.Profile, inbound_message: aries_cloudagent.transport.inbound.message.InboundMessage, send_outbound: Coroutine[T_co, T_contra, V_co], send_webhook: Coroutine[T_co, T_contra, V_co] = None*)

Configure responder and message context and invoke the message handler.

Parameters

- **profile** – The profile associated with the inbound message
- **inbound_message** – The inbound message instance
- **send_outbound** – Async function to send outbound messages
- **send_webhook** – Async function to dispatch a webhook

Returns The response from the handler

log_task (*task: aries_cloudagent.utils.task_queue.CompletedTask*)

Log a completed task using the stats collector.

make_message (*parsed_msg: dict*) → *aries_cloudagent.messaging.agent_message.AgentMessage*

Deserialize a message dict into the appropriate message instance.

Given a dict describing a message, this method returns an instance of the related message class.

Parameters **parsed_msg** – The parsed message

Returns An instance of the corresponding message class for this message

Raises

- `MessageParseError` – If the message doesn't specify @type
- `MessageParseError` – If there is no message class registered to handle
- the given type

put_task (*coro: Coroutine[T_co, T_contra, V_co], complete: Callable = None, ident: str = None*) → *aries_cloudagent.utils.task_queue.PendingTask*

Run a task in the task queue, potentially blocking other handlers.

queue_message (*profile: aries_cloudagent.core.profile.Profile, inbound_message: aries_cloudagent.transport.inbound.message.InboundMessage, send_outbound: Coroutine[T_co, T_contra, V_co], send_webhook: Coroutine[T_co, T_contra, V_co] = None, complete: Callable = None*) → *aries_cloudagent.utils.task_queue.PendingTask*

Add a message to the processing queue for handling.

Parameters

- **profile** – The profile associated with the inbound message
- **inbound_message** – The inbound message instance
- **send_outbound** – Async function to send outbound messages
- **send_webhook** – Async function to dispatch a webhook
- **complete** – Function to call when the handler has completed

Returns A pending task instance resolving to the handler task

run_task (*coro: Coroutine[T_co, T_contra, V_co], complete: Callable = None, ident: str = None*) → *_asyncio.Task*
Run a task in the task queue, potentially blocking other handlers.

setup ()
Perform async instance setup.

class `aries_cloudagent.core.dispatcher.DispatcherResponder` (*context: aries_cloudagent.messaging.request_context.RequestContext, inbound_message: aries_cloudagent.transport.inbound.message.InboundMessage, send_outbound: Coroutine[T_co, T_contra, V_co], send_webhook: Coroutine[T_co, T_contra, V_co] = None, **kwargs*)

Bases: `aries_cloudagent.messaging.responder.BaseResponder`

Handle outgoing messages from message handlers.

create_outbound (*message: Union[aries_cloudagent.messaging.agent_message.AgentMessage, str, bytes], **kwargs*) → `aries_cloudagent.transport.outbound.message.OutboundMessage`
Create an OutboundMessage from a message body.

Parameters *message* – The message payload

send_outbound (*message: aries_cloudagent.transport.outbound.message.OutboundMessage*)
Send outbound message.

Parameters *message* – The *OutboundMessage* to be sent

send_webhook (*topic: str, payload: dict*)
Dispatch a webhook.

Parameters

- **topic** – the webhook topic identifier
- **payload** – the webhook payload value

aries_cloudagent.core.error module

Common exception classes.

exception `aries_cloudagent.core.error.BaseError` (**args, error_code: str = None, **kwargs*)

Bases: `Exception`

Generic exception class which other exceptions should inherit from.

message
Accessor for the error message.

roll_up
Accessor for nested error messages rolled into one line.
For display: aiohttp.web errors truncate after newline.

```
exception aries_cloudagent.core.error.ProfileDuplicateError(*args, error_code:
                                                         str = None,
                                                         **kwargs)
```

Bases: *aries_cloudagent.core.error.ProfileError*

Profile with the given name already exists.

```
exception aries_cloudagent.core.error.ProfileError(*args, error_code: str = None,
                                                    **kwargs)
```

Bases: *aries_cloudagent.core.error.BaseError*

Base error for profile operations.

```
exception aries_cloudagent.core.error.ProfileNotFoundError(*args, error_code: str
                                                         = None, **kwargs)
```

Bases: *aries_cloudagent.core.error.ProfileError*

Requested profile was not found.

```
exception aries_cloudagent.core.error.ProfileSessionInactiveError(*args, error_code:
                                                                str = None,
                                                                **kwargs)
```

Bases: *aries_cloudagent.core.error.ProfileError*

Error raised when a profile session is not currently active.

```
exception aries_cloudagent.core.error.ProtocolDefinitionValidationError(*args,
                                                                error_code:
                                                                str
                                                                =
                                                                None,
                                                                **kwargs)
```

Bases: *aries_cloudagent.core.error.BaseError*

Error raised when there is a problem validating a protocol definition.

```
exception aries_cloudagent.core.error.ProtocolMinorVersionNotSupported(*args,
                                                                error_code:
                                                                str
                                                                =
                                                                None,
                                                                **kwargs)
```

Bases: *aries_cloudagent.core.error.BaseError*

Minimum minor version protocol error.

Error raised when protocol support exists but minimum minor version is higher than in @type parameter.

```
exception aries_cloudagent.core.error.StartupError(*args, error_code: str = None,
                                                    **kwargs)
```

Bases: *aries_cloudagent.core.error.BaseError*

Error raised when there is a problem starting the conductor.

aries_cloudagent.core.in_memory module

Manage in-memory profile interaction.

```
class aries_cloudagent.core.in_memory.InMemoryProfile (*, context:
    aries_cloudagent.config.injection_context.InjectionContext
    = None, name: str = None)
```

Bases: *aries_cloudagent.core.profile.Profile*

Provide access to in-memory profile management.

Used primarily for testing.

BACKEND_NAME = 'in_memory'

TEST_PROFILE_NAME = 'test-profile'

session (context: *aries_cloudagent.config.injection_context.InjectionContext* = None) →
aries_cloudagent.core.profile.ProfileSession
Start a new interactive session with no transaction support requested.

classmethod test_profile (settings: *Mapping[str, Any]* = None, bind: *Mapping[Type[CT_co], Any]* = None) →
aries_cloudagent.core.in_memory.InMemoryProfile
Used in tests to create a standard InMemoryProfile.

classmethod test_session (settings: *Mapping[str, Any]* = None, bind: *Mapping[Type[CT_co], Any]* = None) →
aries_cloudagent.core.in_memory.InMemoryProfileSession
Used in tests to quickly create InMemoryProfileSession.

transaction (context: *aries_cloudagent.config.injection_context.InjectionContext* = None) →
aries_cloudagent.core.profile.ProfileSession
Start a new interactive session with commit and rollback support.

If the current backend does not support transactions, then commit and rollback operations of the session will not have any effect.

```
class aries_cloudagent.core.in_memory.InMemoryProfileManager
    Bases: aries_cloudagent.core.profile.ProfileManager
```

Manager for producing in-memory wallet/storage implementation.

open (context: *aries_cloudagent.config.injection_context.InjectionContext*, config: *Mapping[str, Any]* = None) → *aries_cloudagent.core.profile.Profile*
Open an instance of an existing profile.

provision (context: *aries_cloudagent.config.injection_context.InjectionContext*, config: *Mapping[str, Any]* = None) → *aries_cloudagent.core.profile.Profile*
Provision a new instance of a profile.

```
class aries_cloudagent.core.in_memory.InMemoryProfileSession (profile:
    aries_cloudagent.core.profile.Profile,
    *, context:
    aries_cloudagent.config.injection_context.InjectionContext
    = None, settings:
    Mapping[str, Any]
    = None)
```

Bases: *aries_cloudagent.core.profile.ProfileSession*

An active connection to the profile management backend.

storage
Get the *BaseStorage* implementation (helper specific to in-memory profile).

wallet
Get the *BaseWallet* implementation (helper specific to in-memory profile).

aries_cloudagent.core.plugin_registry module

Handle registration of plugin modules for extending functionality.

```
class aries_cloudagent.core.plugin_registry.PluginRegistry
    Bases: object

    Plugin registry for indexing application plugins.

    init_context (context: aries_cloudagent.config.injection_context.InjectionContext)
        Call plugin setup methods on the current context.

    load_protocol_version (context: aries_cloudagent.config.injection_context.InjectionContext,
                           mod: module, version_definition: dict = None)
        Load a particular protocol version.

    load_protocols (context: aries_cloudagent.config.injection_context.InjectionContext, plugin: module)
        For modules that don't implement setup, register protocols manually.

    plugin_names
        Accessor for a list of all plugin modules.

    plugins
        Accessor for a list of all plugin modules.

    post_process_routes (app)
        Call route binary file response OpenAPI fixups if applicable.

    register_admin_routes (app)
        Call route registration methods on the current context.

    register_package (package_name: str) → Sequence[module]
        Register all modules (sub-packages) under a given package name.

    register_plugin (module_name: str) → module
        Register a plugin module.

    validate_version (version_list, module_name)
        Validate version dict format.
```

aries_cloudagent.core.profile module

Classes for managing profile information within a request context.

```
class aries_cloudagent.core.profile.Profile (*, context: aries_cloudagent.config.injection_context.InjectionContext
                                             = None, name: str = None, created: bool = False)

    Bases: abc.ABC

    Base abstraction for handling identity-related state.

    BACKEND_NAME = None

    DEFAULT_NAME = 'default'

    backend
        Accessor for the backend implementation name.

    close ()
        Close the profile instance.
```

context

Accessor for the injection context.

created

Accessor for the created flag indicating a new profile.

inject (*base_cls: Type[Inject], settings: Mapping[str, object] = None, *, required: bool = True*) → Optional[Inject]
 Get the provided instance of a given class identifier.

Parameters

- **cls** – The base class to retrieve an instance of
- **settings** – An optional mapping providing configuration to the provider

Returns An instance of the base class, or None

name

Accessor for the profile name.

remove()

Remove the profile.

session (*context: aries_cloudagent.config.injection_context.InjectionContext = None*) → aries_cloudagent.core.profile.ProfileSession
 Start a new interactive session with no transaction support requested.

settings

Accessor for scope-specific settings.

transaction (*context: aries_cloudagent.config.injection_context.InjectionContext = None*) → aries_cloudagent.core.profile.ProfileSession
 Start a new interactive session with commit and rollback support.

If the current backend does not support transactions, then commit and rollback operations of the session will not have any effect.

class aries_cloudagent.core.profile.ProfileManager

Bases: abc.ABC

Handle provision and open for profile instances.

open (*context: aries_cloudagent.config.injection_context.InjectionContext, config: Mapping[str, Any] = None*) → aries_cloudagent.core.profile.Profile
 Open an instance of an existing profile.

provision (*context: aries_cloudagent.config.injection_context.InjectionContext, config: Mapping[str, Any] = None*) → aries_cloudagent.core.profile.Profile
 Provision a new instance of a profile.

class aries_cloudagent.core.profile.ProfileManagerProvider

Bases: aries_cloudagent.config.base.BaseProvider

The standard profile manager provider which keys off the selected wallet type.

MANAGER_TYPES = {'askar': 'aries_cloudagent.askar.profile.AskarProfileManager', 'in_m

provide (*settings: aries_cloudagent.config.base.BaseSettings, injector: aries_cloudagent.config.base.BaseInjector*)
 Create the profile manager instance.

```
class aries_cloudagent.core.profile.ProfileSession (profile:
                                                    aries_cloudagent.core.profile.Profile,
                                                    *,
                                                    context:
                                                    aries_cloudagent.config.injection_context.InjectionContext
                                                    = None, settings: Mapping[str,
                                                    Any] = None)
```

Bases: `abc.ABC`

An active connection to the profile management backend.

active

Accessor for the session active state.

commit ()

Commit any updates performed within the transaction.

If the current session is not a transaction, then nothing is performed.

context

Accessor for the associated injection context.

inject (*base_cls: Type[Inject], settings: Mapping[str, object] = None, *, required: bool = True*) → *Optional[Inject]*

Get the provided instance of a given class identifier.

Parameters

- **cls** – The base class to retrieve an instance of
- **settings** – An optional mapping providing configuration to the provider

Returns An instance of the base class, or None

is_transaction

Check if the session supports commit and rollback operations.

profile

Accessor for the associated profile instance.

rollback ()

Roll back any updates performed within the transaction.

If the current session is not a transaction, then nothing is performed.

settings

Accessor for scope-specific settings.

aries_cloudagent.core.protocol_registry module

Handle registration and publication of supported protocols.

```
class aries_cloudagent.core.protocol_registry.ProtocolRegistry
```

Bases: `object`

Protocol registry for indexing message families.

controllers

Accessor for a list of all protocol controller functions.

message_types

Accessor for a list of all message types.

parse_type_string (*message_type*)

Parse message type string and return dict with info.

prepare_disclosed (*context: aries_cloudagent.config.injection_context.InjectionContext, protocols: Sequence[str]*)

Call controllers and return publicly supported message families and roles.

protocols

Accessor for a list of all message protocols.

protocols_matching_query (*query: str*) → Sequence[str]

Return a list of message protocols matching a query string.

register_controllers (**controller_sets, version_definition=None*)

Add new controllers.

Parameters controller_sets – Mappings of message families to coroutines

register_message_types (**typesets, version_definition=None*)

Add new supported message types.

Parameters

- **typesets** – Mappings of message types to register
- **version_definition** – Optional version definition dict

resolve_message_class (*message_type: str*) → type

Resolve a message_type to a message class.

Given a message type identifier, this method returns the corresponding registered message class.

Parameters message_type – Message type to resolve

Returns The resolved message class

aries_cloudagent.holder package

Submodules

aries_cloudagent.holder.routes module

Holder admin routes.

class aries_cloudagent.holder.routes.**AttributeMimeTypesResultSchema** (**args, **kwargs*)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for credential attribute MIME type.

class aries_cloudagent.holder.routes.**CredBriefListSchema** (**args, **kwargs*)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for a credential query.

results = <fields.List (default=<marshmallow.missing>, attribute=None, validate=None, req

class aries_cloudagent.holder.routes.**CredBriefSchema** (**args, **kwargs*)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema with credential brief for a credential query.

attrs = <fields.Dict (default=<marshmallow.missing>, attribute=None, validate=None, req

```

cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
cred_rev_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
referent = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
rev_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
class aries_cloudagent.holder.routes.CredIdMatchInfoSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Path parameters and validators for request taking credential id.

    credential_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
class aries_cloudagent.holder.routes.CredRevokedQueryStringSchema (*args,
                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Path parameters and validators for request seeking cred revocation status.

    fro = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    to = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_clo
class aries_cloudagent.holder.routes.CredRevokedResultSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Result schema for credential revoked request.

    revoked = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None
class aries_cloudagent.holder.routes.CredentialsListQueryStringSchema (*args,
                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Parameters and validators for query string in credentials list query.

    count = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
    start = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
    wql = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
class aries_cloudagent.holder.routes.HolderModuleResponseSchema (*args,
                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Response schema for Holder Module.

aries_cloudagent.holder.routes.credentials_attr_mime_types_get (request:
                                                                    <sphinx.ext.autodoc.importer._MockObj
                                                                    object at
                                                                    0x7f760f561990>)

    Request handler for getting credential attribute MIME types.

    Parameters request – aiohttp request object

    Returns The MIME types response
aries_cloudagent.holder.routes.credentials_get (request:
                                                                    <sphinx.ext.autodoc.importer._MockObject
                                                                    object at 0x7f760f561990>)

    Request handler for retrieving a credential.

    Parameters request – aiohttp request object

```

Returns The credential response

```
aries_cloudagent.holder.routes.credentials_list (request:
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760f561990>)
```

Request handler for searching credential records.

Parameters **request** – aiohttp request object

Returns The credential list response

```
aries_cloudagent.holder.routes.credentials_remove (request:
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760f561990>)
```

Request handler for searching connection records.

Parameters **request** – aiohttp request object

Returns The connection list response

```
aries_cloudagent.holder.routes.credentials_revoked (request:
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760f561990>)
```

Request handler for querying revocation status of credential.

Parameters **request** – aiohttp request object

Returns The credential response

```
aries_cloudagent.holder.routes.post_process_routes (app:
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760f561990>)
```

Amend swagger API.

```
aries_cloudagent.holder.routes.register (app: <sphinx.ext.autodoc.importer._MockObject
object at 0x7f760f561990>)
```

Register routes.

aries_cloudagent.indy package

Subpackages

aries_cloudagent.indy.sdk package

Submodules

aries_cloudagent.indy.sdk.error module

Indy error handling.

```
class aries_cloudagent.indy.sdk.error.IndyErrorHandler (message:      str      =
None,                      error_cls:
Type[aries_cloudagent.core.error.BaseError]
=
<class
'aries_cloudagent.core.error.BaseError'>)
```

Bases: `object`

Trap IndyError and raise an appropriate LedgerError instead.

```

classmethod wrap_error (err_value: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760f05cd50>, message: str = None, error_cls: Type[aries_cloudagent.core.error.BaseError] = <class 'aries_cloudagent.core.error.BaseError'>) →
    aries_cloudagent.core.error.BaseError

```

Create an instance of BaseError from an IndyError.

aries_cloudagent.indy.sdk.holder module

Indy SDK holder implementation.

```

class aries_cloudagent.indy.sdk.holder.IndySdkHolder (wallet:
    aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet
    Bases: aries_cloudagent.indy.holder.IndyHolder

```

Indy-SDK holder implementation.

```

create_credential_request (credential_offer: dict, credential_definition: dict, holder_did: str)
    → Tuple[str, str]

```

Create a credential request for the given credential offer.

Parameters

- **credential_offer** – The credential offer to create request for
- **credential_definition** – The credential definition to create an offer for
- **holder_did** – the DID of the agent making the request

Returns A tuple of the credential request and credential request metadata

```

create_presentation (presentation_request: dict, requested_credentials: dict, schemas: dict, credential_definitions: dict, rev_states: dict = None) → str

```

Get credentials stored in the wallet.

Parameters

- **presentation_request** – Valid indy format presentation request
- **requested_credentials** – Indy format requested credentials
- **schemas** – Indy formatted schemas JSON
- **credential_definitions** – Indy formatted credential definitions JSON
- **rev_states** – Indy format revocation states JSON

```

create_revocation_state (cred_rev_id: str, rev_reg_def: dict, rev_reg_delta: dict, timestamp:
    int, tails_file_path: str) → str

```

Create current revocation state for a received credential.

Parameters

- **cred_rev_id** – credential revocation id in revocation registry
- **rev_reg_def** – revocation registry definition
- **rev_reg_delta** – revocation delta
- **timestamp** – delta timestamp

Returns the revocation state

```

credential_revoked (ledger: aries_cloudagent.ledger.base.BaseLedger, credential_id: str, fro: int
    = None, to: int = None) → bool

```

Check ledger for revocation status of credential by cred id.

Parameters `credential_id` – Credential id to check

delete_credential (*credential_id: str*)

Remove a credential stored in the wallet.

Parameters `credential_id` – Credential id to remove

get_credential (*credential_id: str*) → str

Get a credential stored in the wallet.

Parameters `credential_id` – Credential id to retrieve

get_credentials (*start: int, count: int, wql: dict*)

Get credentials stored in the wallet.

Parameters

- **start** – Starting index
- **count** – Number of records to return
- **wql** – wql query dict

get_credentials_for_presentation_request_by_referent (*presentation_request: dict, referents: Sequence[str], start: int, count: int, extra_query: dict = {}*)

Get credentials stored in the wallet.

Parameters

- **presentation_request** – Valid presentation request from issuer
- **referents** – Presentation request referents to use to search for creds
- **start** – Starting index
- **count** – Maximum number of records to return
- **extra_query** – wql query dict

get_mime_type (*credential_id: str, attr: str = None*) → Union[dict, str]

Get MIME type per attribute (or for all attributes).

Parameters

- **credential_id** – credential id
- **attr** – attribute of interest or omit for all

Returns: Attribute MIME type or dict mapping attribute names to MIME types `attr_meta_json = all_meta.tags.get(attr)`

store_credential (*credential_definition: dict, credential_data: dict, credential_request_metadata: dict, credential_attr_mime_types=None, credential_id: str = None, rev_reg_def: dict = None*) → str

Store a credential in the wallet.

Parameters

- **credential_definition** – Credential definition for this credential
- **credential_data** – Credential data generated by the issuer

- **credential_request_metadata** – credential request metadata generated by the issuer
- **credential_attr_mime_types** – dict mapping attribute names to (optional) MIME types to store as non-secret record, if specified
- **credential_id** – optionally override the stored credential id
- **rev_reg_def** – revocation registry definition in json

Returns the ID of the stored credential

aries_cloudagent.indy.sdk.issuer module

Indy SDK issuer implementation.

class `aries_cloudagent.indy.sdk.issuer.IndySdkIssuer` (*profile:*
aries_cloudagent.indy.sdk.profile.IndySdkProfile)
 Bases: `aries_cloudagent.indy.issuer.IndyIssuer`

Indy-SDK issuer implementation.

create_and_store_credential_definition (*origin_did: str, schema: dict, signature_type: str = None, tag: str = None, support_revocation: bool = False*) → *Tuple[str, str]*

Create a new credential definition and store it in the wallet.

Parameters

- **origin_did** – the DID issuing the credential definition
- **schema** – the schema used as a basis
- **signature_type** – the credential definition signature type (default ‘CL’)
- **tag** – the credential definition tag
- **support_revocation** – whether to enable revocation for this credential def

Returns A tuple of the credential definition ID and JSON

create_and_store_revocation_registry (*origin_did: str, cred_def_id: str, revoc_def_type: str, tag: str, max_cred_num: int, tails_base_path: str*) → *Tuple[str, str, str]*

Create a new revocation registry and store it in the wallet.

Parameters

- **origin_did** – the DID issuing the revocation registry
- **cred_def_id** – the identifier of the related credential definition
- **revoc_def_type** – the revocation registry type (default CL_ACCUM)
- **tag** – the unique revocation registry tag
- **max_cred_num** – the number of credentials supported in the registry
- **tails_base_path** – where to store the tails file

Returns A tuple of the revocation registry ID, JSON, and entry JSON

create_credential (*schema: dict, credential_offer: dict, credential_request: dict, credential_values: dict, cred_ex_id: str, rev_reg_id: str = None, tails_file_path: str = None*) → Tuple[str, str]

Create a credential.

Args *schema*: Schema to create credential for *credential_offer*: Credential Offer to create credential for *credential_request*: Credential request to create credential for *credential_values*: Values to go in credential *cred_ex_id*: credential exchange identifier to use in issuer cred *rev_reg_id*: ID of the revocation registry *tails_file_path*: Path to the local tails file

Returns A tuple of created credential and revocation id

create_credential_offer (*credential_definition_id: str*) → str

Create a credential offer for the given credential definition id.

Parameters *credential_definition_id* – The credential definition to create an offer for

Returns The created credential offer

create_schema (*origin_id: str, schema_name: str, schema_version: str, attribute_names: Sequence[str]*) → Tuple[str, str]

Create a new credential schema.

Parameters

- **origin_id** – the DID issuing the credential definition
- **schema_name** – the schema name
- **schema_version** – the schema version
- **attribute_names** – a sequence of schema attribute names

Returns A tuple of the schema ID and JSON

credential_definition_in_wallet (*credential_definition_id: str*) → bool

Check whether a given credential definition ID is present in the wallet.

Parameters *credential_definition_id* – The credential definition ID to check

merge_revocation_registry_deltas (*fro_delta: str, to_delta: str*) → str

Merge revocation registry deltas.

Parameters

- **fro_delta** – original delta in JSON format
- **to_delta** – incoming delta in JSON format

Returns Merged delta in JSON format

revoke_credentials (*rev_reg_id: str, tails_file_path: str, cred_rev_ids: Sequence[str]*) → (<class 'str'>, typing.Sequence[str])

Revoke a set of credentials in a revocation registry.

Parameters

- **rev_reg_id** – ID of the revocation registry
- **tails_file_path** – path to the local tails file
- **cred_rev_ids** – sequences of credential indexes in the revocation registry

Returns Tuple with the combined revocation delta, list of cred rev ids not revoked

aries_cloudagent.indy.sdk.profile module

Manage Indy-SDK profile interaction.

```
class aries_cloudagent.indy.sdk.profile.IndySdkProfile (opened:
                                                    aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWa
                                                    context:
                                                    aries_cloudagent.config.injection_context.InjectionC
                                                    = None)

    Bases: aries_cloudagent.core.profile.Profile
    Provide access to Indy profile interaction methods.

    BACKEND_NAME = 'indy'

    bind_providers ()
        Initialize the profile-level instance providers.

    close ()
        Close the profile instance.

    init_ledger_pool ()
        Initialize the ledger pool.

    name
        Accessor for the profile name.

    remove ()
        Remove the profile associated with this instance.

    session (context: aries_cloudagent.config.injection_context.InjectionContext = None) →
        aries_cloudagent.core.profile.ProfileSession
        Start a new interactive session with no transaction support requested.

    transaction (context: aries_cloudagent.config.injection_context.InjectionContext = None) →
        aries_cloudagent.core.profile.ProfileSession
        Start a new interactive session with commit and rollback support.

        If the current backend does not support transactions, then commit and rollback operations of the session
        will not have any effect.

    wallet
        Accessor for the opened wallet instance.

class aries_cloudagent.indy.sdk.profile.IndySdkProfileManager
    Bases: aries_cloudagent.core.profile.ProfileManager
    Manager for Indy-SDK wallets.

    open (context: aries_cloudagent.config.injection_context.InjectionContext, config: Mapping[str, Any] =
        None) → aries_cloudagent.core.profile.Profile
        Open an instance of an existing profile.

    provision (context: aries_cloudagent.config.injection_context.InjectionContext, config: Mapping[str,
        Any] = None) → aries_cloudagent.core.profile.Profile
        Provision a new instance of a profile.
```

```
class aries_cloudagent.indy.sdk.profile.IndySdkProfileSession (profile:
    aries_cloudagent.core.profile.Profile,
    *, context:
    aries_cloudagent.config.injection_context.
    = None, settings:
    Mapping[str,
    Any] = None)

Bases: aries_cloudagent.core.profile.ProfileSession

An active connection to the profile management backend.
```

aries_cloudagent.indy.sdk.util module

Indy utilities.

```
aries_cloudagent.indy.sdk.util.create_tails_reader (tails_file_path: str) → int
    Get a handle for the blob_storage file reader.
```

```
aries_cloudagent.indy.sdk.util.create_tails_writer (tails_base_dir: str) → int
    Get a handle for the blob_storage file writer.
```

aries_cloudagent.indy.sdk.verifier module

Indy SDK verifier implementation.

```
class aries_cloudagent.indy.sdk.verifier.IndySdkVerifier (ledger:
    aries_cloudagent.ledger.indy.IndySdkLedger)

Bases: aries_cloudagent.indy.verifier.IndyVerifier
```

Indy-SDK verifier implementation.

```
check_timestamps (pres_req: Mapping[KT, VT_co], pres: Mapping[KT, VT_co], rev_reg_defs:
    Mapping[KT, VT_co])
```

Check for suspicious, missing, and superfluous timestamps.

Raises ValueError on timestamp in the future, prior to rev reg creation, superfluous or missing.

Parameters

- **pres_req** – indy proof request
- **pres** – indy proof request
- **rev_reg_defs** – rev reg defs by rev reg id, augmented with transaction times

```
non_revoc_intervals (pres_req: dict, pres: dict)
```

Remove superfluous non-revocation intervals in presentation request.

Irrevocable credentials constitute proof of non-revocation, but indy rejects proof requests with non-revocation intervals lining up with non-revocable credentials in proof: seek and remove.

Parameters

- **pres_req** – presentation request
- **pres** – corresponding presentation

```
pre_verify (pres_req: dict, pres: dict)
```

Check for essential components and tampering in presentation.

Visit encoded attribute values against raw, and predicate bounds, in presentation, cross-reference against presentation request.

Parameters

- **pres_req** – presentation request
- **pres** – corresponding presentation

verify_presentation (*pres_req, pres, schemas, credential_definitions, rev_reg_defs, rev_reg_entries*) → bool

Verify a presentation.

Parameters

- **pres_req** – Presentation request data
- **pres** – Presentation data
- **schemas** – Schema data
- **credential_definitions** – credential definition data
- **rev_reg_defs** – revocation registry definitions
- **rev_reg_entries** – revocation registry entries

aries_cloudagent.indy.sdk.wallet_plugin module

Utility for loading Postgres wallet plug-in.

aries_cloudagent.indy.sdk.wallet_plugin.file_ext()
Determine file extension based on platform.

aries_cloudagent.indy.sdk.wallet_plugin.load_postgres_plugin (*storage_config, storage_creds, raise_exc=False*)
Load postgres dll and configure postgres wallet.

aries_cloudagent.indy.sdk.wallet_setup module

Indy-SDK wallet setup and configuration.

class **aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet** (*config: aries_cloudagent.indy.sdk.wallet_setup.IndyV created, handle, master_secret_id: str*)

Bases: `object`

Handle and metadata for an opened Indy wallet.

close()
Close previously-opened wallet, removing it if so configured.

name
Accessor for the opened wallet name.

class **aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig** (*config: Mapping[str, Any] = None*)

Bases: `object`

A helper class for handling Indy-SDK wallet configuration.

```
DEFAULT_FRESHNESS = False
```

```
DEFAULT_KEY = ''
```

```
DEFAULT_KEY_DERIVATION = 'ARGON2I_MOD'
```

```
DEFAULT_STORAGE_TYPE = None
```

```
KEY_DERIVATION_ARGON2I_INT = 'ARGON2I_INT'
```

```
KEY_DERIVATION_ARGON2I_MOD = 'ARGON2I_MOD'
```

```
KEY_DERIVATION_RAW = 'RAW'
```

```
create_wallet() → aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet
```

Create a new wallet.

Raises

- ProfileDuplicateError – If there was an existing wallet with the same name
- ProfileError – If there was a problem removing the wallet
- ProfileError – If there was another libindy error

```
open_wallet(created: bool = False) → aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet
```

Open wallet, removing and/or creating it if so configured.

Raises

- ProfileError – If wallet not found after creation
- ProfileNotFoundError – If the wallet is not found
- ProfileError – If the wallet is already open
- ProfileError – If there is another libindy error

```
remove_wallet()
```

Remove an existing wallet.

Raises

- ProfileNotFoundError – If the wallet could not be found
- ProfileError – If there was another libindy error

```
wallet_access
```

Accessor the Indy wallet access info.

```
wallet_config
```

Accessor for the Indy wallet config.

Submodules

aries_cloudagent.indy.holder module

Base Indy Holder class.

```
class aries_cloudagent.indy.holder.IndyHolder
```

Bases: `abc.ABC`

Base class for holder.

```
CHUNK = 256
```

RECORD_TYPE_MIME_TYPES = 'attribute-mime-types'

create_credential_request (*credential_offer: dict, credential_definition: dict, holder_did: str*)
→ Tuple[str, str]

Create a credential request for the given credential offer.

Parameters

- **credential_offer** – The credential offer to create request for
- **credential_definition** – The credential definition to create an offer for
- **holder_did** – the DID of the agent making the request

Returns A tuple of the credential request and credential request metadata

create_presentation (*presentation_request: dict, requested_credentials: dict, schemas: dict, credential_definitions: dict, rev_states: dict = None*) → str

Get credentials stored in the wallet.

Parameters

- **presentation_request** – Valid indy format presentation request
- **requested_credentials** – Indy format requested credentials
- **schemas** – Indy formatted schemas JSON
- **credential_definitions** – Indy formatted credential definitions JSON
- **rev_states** – Indy format revocation states JSON

create_revocation_state (*cred_rev_id: str, rev_reg_def: dict, rev_reg_delta: dict, timestamp: int, tails_file_path: str*) → str

Create current revocation state for a received credential.

Parameters

- **cred_rev_id** – credential revocation id in revocation registry
- **rev_reg_def** – revocation registry definition
- **rev_reg_delta** – revocation delta
- **timestamp** – delta timestamp

Returns the revocation state

credential_revoked (*ledger: aries_cloudagent.ledger.base.BaseLedger, credential_id: str, fro: int = None, to: int = None*) → bool

Check ledger for revocation status of credential by cred id.

Parameters **credential_id** – Credential id to check

delete_credential (*credential_id: str*)

Remove a credential stored in the wallet.

Parameters **credential_id** – Credential id to remove

get_credential (*credential_id: str*) → str

Get a credential stored in the wallet.

Parameters **credential_id** – Credential id to retrieve

get_mime_type (*credential_id: str, attr: str = None*) → Union[dict, str]

Get MIME type per attribute (or for all attributes).

Parameters

- **credential_id** – credential id
- **attr** – attribute of interest or omit for all

Returns: Attribute MIME type or dict mapping attribute names to MIME types `attr_meta_json = all_meta.tags.get(attr)`

store_credential (*credential_definition: dict, credential_data: dict, credential_request_metadata: dict, credential_attr_mime_types=None, credential_id: str = None, rev_reg_def: dict = None*)
Store a credential in the wallet.

Parameters

- **credential_definition** – Credential definition for this credential
- **credential_data** – Credential data generated by the issuer
- **credential_request_metadata** – credential request metadata generated by the issuer
- **credential_attr_mime_types** – dict mapping attribute names to (optional) MIME types to store as non-secret record, if specified
- **credential_id** – optionally override the stored credential id
- **rev_reg_def** – revocation registry definition in json

Returns the ID of the stored credential

exception `aries_cloudagent.indy.holder.IndyHolderError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Base class for holder exceptions.

aries_cloudagent.indy.issuer module

Base Indy Issuer class.

class `aries_cloudagent.indy.issuer.IndyIssuer`
Bases: `abc.ABC`

Base class for Indy Issuer.

create_and_store_credential_definition (*origin_did: str, schema: dict, signature_type: str = None, tag: str = None, support_revocation: bool = False*) → Tuple[str, str]

Create a new credential definition and store it in the wallet.

Parameters

- **origin_did** – the DID issuing the credential definition
- **schema_json** – the schema used as a basis
- **signature_type** – the credential definition signature type (default 'CL')
- **tag** – the credential definition tag
- **support_revocation** – whether to enable revocation for this credential def

Returns A tuple of the credential definition ID and JSON

create_and_store_revocation_registry (*origin_did: str, cred_def_id: str, revoc_def_type: str, tag: str, max_cred_num: int, tails_base_path: str*) → Tuple[str, str, str]

Create a new revocation registry and store it in the wallet.

Parameters

- **origin_did** – the DID issuing the revocation registry
- **cred_def_id** – the identifier of the related credential definition
- **revoc_def_type** – the revocation registry type (default CL_ACCUM)
- **tag** – the unique revocation registry tag
- **max_cred_num** – the number of credentials supported in the registry
- **tails_base_path** – where to store the tails file

Returns A tuple of the revocation registry ID, JSON, and entry JSON

create_credential (*schema: dict, credential_offer: dict, credential_request: dict, credential_values: dict, cred_ex_id: str, revoc_reg_id: str = None, tails_reader_handle: int = None*) → Tuple[str, str]

Create a credential.

Args *schema*: Schema to create credential for *credential_offer*: Credential Offer to create credential for *credential_request*: Credential request to create credential for *credential_values*: Values to go in credential *cred_ex_id*: credential exchange identifier to use in issuer cred rev rec *revoc_reg_id*: ID of the revocation registry *tails_reader_handle*: Handle for the tails file blob reader

Returns A tuple of created credential and revocation id

create_credential_offer (*credential_definition_id*) → str

Create a credential offer for the given credential definition id.

Parameters **credential_definition_id** – The credential definition to create an offer for

Returns The created credential offer

create_schema (*origin_did: str, schema_name: str, schema_version: str, attribute_names: Sequence[str]*) → Tuple[str, str]

Create a new credential schema and store it in the wallet.

Parameters

- **origin_did** – the DID issuing the credential definition
- **schema_name** – the schema name
- **schema_version** – the schema version
- **attribute_names** – a sequence of schema attribute names

Returns A tuple of the schema ID and JSON

credential_definition_in_wallet (*credential_definition_id: str*) → bool

Check whether a given credential definition ID is present in the wallet.

Parameters **credential_definition_id** – The credential definition ID to check

make_credential_definition_id (*origin_did: str, schema: dict, signature_type: str = None, tag: str = None*) → str

Derive the ID for a credential definition.

make_schema_id (*origin_id: str, schema_name: str, schema_version: str*) → str
Derive the ID for a schema.

merge_revocation_registry_deltas (*fro_delta: str, to_delta: str*) → str
Merge revocation registry deltas.

Parameters

- **fro_delta** – original delta in JSON format
- **to_delta** – incoming delta in JSON format

Returns Merged delta in JSON format

revoke_credentials (*revoc_reg_id: str, tails_file_path: str, cred_rev_ids: Sequence[str]*) -> (*<class 'str'>, typing.Sequence[str]*)

Revoke a set of credentials in a revocation registry.

Parameters

- **revoc_reg_id** – ID of the revocation registry
- **tails_file_path** – path to the local tails file
- **cred_rev_ids** – sequences of credential indexes in the revocation registry

Returns Tuple with the combined revocation delta, list of cred rev ids not revoked

exception `aries_cloudagent.indy.issuer.IndyIssuerError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Generic issuer error.

exception `aries_cloudagent.indy.issuer.IndyIssuerRevocationRegistryFullError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.indy.issuer.IndyIssuerError`

Revocation registry is full when issuing a new credential.

aries_cloudagent.indy.util module

Utilities for dealing with Indy conventions.

`aries_cloudagent.indy.util.generate_pr_nonce` () → str
Generate a nonce for a proof request.

`aries_cloudagent.indy.util.indy_client_dir` (*subpath: str = None, create: bool = False*) → str
Return '/'-terminated subdirectory of indy-client directory.

Parameters

- **subpath** – subpath within indy-client structure
- **create** – whether to create subdirectory if absent

`aries_cloudagent.indy.util.tails_path` (*rev_reg_id: str*) → str
Return path to indy tails file for input rev reg id.

aries_cloudagent.indy.verifier module

Base Indy Verifier class.

class aries_cloudagent.indy.verifier.IndyVerifier

Bases: `abc.ABC`

Base class for Indy Verifier.

verify_presentation(*presentation_request, presentation, schemas, credential_definitions, rev_reg_defs, rev_reg_entries*)

Verify a presentation.

Parameters

- **presentation_request** – Presentation request data
- **presentation** – Presentation data
- **schemas** – Schema data
- **credential_definitions** – credential definition data
- **rev_reg_defs** – revocation registry definitions
- **rev_reg_entries** – revocation registry entries

aries_cloudagent.ledger package

Submodules

aries_cloudagent.ledger.base module

Ledger base class.

class aries_cloudagent.ledger.base.BaseLedger

Bases: `abc.ABC`

Base class for ledger.

BACKEND_NAME = None

accept_txn_author_agreement(*taa_record: dict, mechanism: str, accept_time: int = None*)

Save a new record recording the acceptance of the TAA.

backend

Accessor for the ledger backend name.

create_and_send_credential_definition(*issuer: aries_cloudagent.indy.issuer.IndyIssuer, schema_id: str, signature_type: str = None, tag: str = None, support_revocation: bool = False*)
→ Tuple[str, dict, bool]

Send credential definition to ledger and store relevant key matter in wallet.

Parameters

- **issuer** – The issuer instance to use for credential definition creation
- **schema_id** – The schema id of the schema to create cred def for
- **signature_type** – The signature type to use on the credential definition
- **tag** – Optional tag to distinguish multiple credential definitions

- **support_revocation** – Optional flag to enable revocation for this cred def

Returns Tuple with cred def id, cred def structure, and whether it's novel

create_and_send_schema (*issuer: aries_cloudagent.indy.issuer.IndyIssuer, schema_name: str, schema_version: str, attribute_names: Sequence[str]*) → Tuple[str, dict]

Send schema to ledger.

Parameters

- **issuer** – The issuer instance to use for schema creation
- **schema_name** – The schema name
- **schema_version** – The schema version
- **attribute_names** – A list of schema attributes

did_to_nym (*did: str*) → str

Remove the ledger's DID prefix to produce a nym.

fetch_txn_author_agreement ()

Fetch the current AML and TAA from the ledger.

get_credential_definition (*credential_definition_id: str*) → dict

Get a credential definition from the cache if available, otherwise the ledger.

Parameters **credential_definition_id** – The schema id of the schema to fetch cred def for

get_endpoint_for_did (*did: str, endpoint_type: aries_cloudagent.ledger.endpoint_type.EndpointType = <EndpointType.ENDPOINT: EndpointTypeName(w3c='Endpoint', indy='endpoint')>*) → str

Fetch the endpoint for a ledger DID.

Parameters

- **did** – The DID to look up on the ledger or in the cache
- **endpoint_type** – The type of the endpoint (default 'endpoint')

get_key_for_did (*did: str*) → str

Fetch the verkey for a ledger DID.

Parameters **did** – The DID to look up on the ledger or in the cache

get_latest_txn_author_acceptance ()

Look up the latest TAA acceptance.

get_nym_role (*did: str*)

Return the role registered to input public DID on the ledger.

Parameters **did** – DID to register on the ledger.

get_revoc_reg_def (*revoc_reg_id: str*) → dict

Look up a revocation registry definition by ID.

get_revoc_reg_delta (*revoc_reg_id: str, timestamp_from=0, timestamp_to=None*) -> (<class 'dict'>, <class 'int'>)

Look up a revocation registry delta by ID.

get_revoc_reg_entry (*revoc_reg_id: str, timestamp: int*)

Get revocation registry entry by revocation registry ID and timestamp.

get_schema (*schema_id: str*) → dict

Get a schema from the cache if available, otherwise fetch from the ledger.

Parameters **schema_id** – The schema id (or stringified sequence number) to retrieve

get_txn_author_agreement (*reload: bool = False*)

Get the current transaction author agreement, fetching it if necessary.

nym_to_did (*nym: str*) → str

Format a nym with the ledger's DID prefix.

read_only

Accessor for the ledger read-only flag.

register_nym (*did: str, verkey: str, alias: str = None, role: str = None*)

Register a nym on the ledger.

Parameters

- **did** – DID to register on the ledger.
- **verkey** – The verification key of the keypair.
- **alias** – Human-friendly alias to assign to the DID.
- **role** – For permissioned ledgers, what role should the new DID have.

rotate_public_did_keypair (*next_seed: str = None*) → None

Rotate keypair for public DID: create new key, submit to ledger, update wallet.

Parameters **next_seed** – seed for incoming ed25519 keypair (default random)

send_revoc_reg_def (*revoc_reg_def: dict, issuer_did: str = None*)

Publish a revocation registry definition to the ledger.

send_revoc_reg_entry (*revoc_reg_id: str, revoc_def_type: str, revoc_reg_entry: dict, issuer_did: str = None*)

Publish a revocation registry entry to the ledger.

taa_digest (*version: str, text: str*)

Generate the digest of a TAA record.

update_endpoint_for_did (*did: str, endpoint: str, endpoint_type: aries_cloudagent.ledger.endpoint_type.EndpointType = <EndpointType.ENDPOINT: EndpointTypeName(w3c='Endpoint', indy='endpoint')>*) → bool

Check and update the endpoint on the ledger.

Parameters

- **did** – The ledger DID
- **endpoint** – The endpoint address
- **endpoint_type** – The type of the endpoint (default 'endpoint')

class aries_cloudagent.ledger.base.**Role**

Bases: `enum.Enum`

Enum for indy roles.

ENDORSER = (101,)

NETWORK_MONITOR = (201,)

ROLE_REMOVE = ('',)

```
STEWARD = (2,)
TRUSTEE = (0,)
USER = (None, '')
get = <function Role.get>
to_indy_num_str() → str
    Return (typically, numeric) string value that indy-sdk associates with role.
    Recall that None signifies USER and "" signifies a role undergoing reset.
token() → str
    Return token identifying role to indy-sdk.
```

aries_cloudagent.ledger.endpoint_type module

Ledger utilities.

```
class aries_cloudagent.ledger.endpoint_type.EndpointType
    Bases: enum.Enum
    Enum for endpoint/service types.
    ENDPOINT = EndpointTypeName(w3c='Endpoint', indy='endpoint')
    LINKED_DOMAINS = EndpointTypeName(w3c='LinkedDomains', indy='linked_domains')
    PROFILE = EndpointTypeName(w3c='Profile', indy='profile')
    get = <function EndpointType.get>
    indy
        internally-facing, on ledger and in wallet.
        Type Indy name of endpoint type
    w3c
        externally-facing.
        Type W3C name of endpoint type
class aries_cloudagent.ledger.endpoint_type.EndpointTypeName(w3c, indy)
    Bases: tuple
    indy
        Alias for field number 1
    w3c
        Alias for field number 0
```

aries_cloudagent.ledger.error module

Ledger related errors.

```
exception aries_cloudagent.ledger.error.BadLedgerRequestError(*args, error_code: str = None, **kwargs)
    Bases: aries_cloudagent.ledger.error.LedgerError
    The current request cannot proceed.
```

exception `aries_cloudagent.ledger.error.ClosedPoolError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.ledger.error.LedgerError`

Indy pool is closed.

exception `aries_cloudagent.ledger.error.LedgerConfigError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.ledger.error.LedgerError`

Base class for ledger configuration errors.

exception `aries_cloudagent.ledger.error.LedgerError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Base class for ledger errors.

exception `aries_cloudagent.ledger.error.LedgerTransactionError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.ledger.error.LedgerError`

The ledger rejected the transaction.

aries_cloudagent.ledger.indy module

Indy ledger implementation.

class `aries_cloudagent.ledger.indy.IndySdkLedger` (pool: `aries_cloudagent.ledger.indy.IndySdkLedgerPool`, wallet: `aries_cloudagent.wallet.indy.IndySdkWallet`)

Bases: `aries_cloudagent.ledger.base.BaseLedger`

Indy ledger class.

BACKEND_NAME = 'indy'

accept_txn_author_agreement (taa_record: dict, mechanism: str, accept_time: int = None)
Save a new record recording the acceptance of the TAA.

check_existing_schema (public_id: str, schema_name: str, schema_version: str, attribute_names: Sequence[str]) → Tuple[str, dict]
Check if a schema has already been published.

create_and_send_credential_definition (issuer: `aries_cloudagent.indy.issuer.IndyIssuer`, schema_id: str, signature_type: str = None, tag: str = None, support_revocation: bool = False)
→ Tuple[str, dict, bool]
Send credential definition to ledger and store relevant key matter in wallet.

Parameters

- **issuer** – The issuer instance to use for credential definition creation
- **schema_id** – The schema id of the schema to create cred def for
- **signature_type** – The signature type to use on the credential definition
- **tag** – Optional tag to distinguish multiple credential definitions
- **support_revocation** – Optional flag to enable revocation for this cred def

Returns Tuple with cred def id, cred def structure, and whether it's novel

create_and_send_schema (*issuer: aries_cloudagent.indy.issuer.IndyIssuer, schema_name: str, schema_version: str, attribute_names: Sequence[str]*) → Tuple[str, dict]

Send schema to ledger.

Parameters

- **issuer** – The issuer instance creating the schema
- **schema_name** – The schema name
- **schema_version** – The schema version
- **attribute_names** – A list of schema attributes

credential_definition_id2schema_id (*credential_definition_id*)

From a credential definition, get the identifier for its schema.

Parameters **credential_definition_id** – The identifier of the credential definition from which to identify a schema

fetch_credential_definition (*credential_definition_id: str*) → dict

Get a credential definition from the ledger by id.

Parameters **credential_definition_id** – The cred def id of the cred def to fetch

fetch_schema_by_id (*schema_id: str*) → dict

Get schema from ledger.

Parameters **schema_id** – The schema id (or stringified sequence number) to retrieve

Returns Indy schema dict

fetch_schema_by_seq_no (*seq_no: int*)

Fetch a schema by its sequence number.

Parameters **seq_no** – schema ledger sequence number

Returns Indy schema dict

fetch_txn_author_agreement () → dict

Fetch the current AML and TAA from the ledger.

get_all_endpoints_for_did (*did: str*) → dict

Fetch all endpoints for a ledger DID.

Parameters **did** – The DID to look up on the ledger or in the cache

get_credential_definition (*credential_definition_id: str*) → dict

Get a credential definition from the cache if available, otherwise the ledger.

Parameters **credential_definition_id** – The schema id of the schema to fetch cred def for

get_endpoint_for_did (*did: str, endpoint_type: aries_cloudagent.ledger.endpoint_type.EndpointType = None*) → str

Fetch the endpoint for a ledger DID.

Parameters

- **did** – The DID to look up on the ledger or in the cache
- **endpoint_type** – The type of the endpoint. If none given, returns all

get_indy_storage () → aries_cloudagent.storage.indy.IndySdkStorage

Get an IndySdkStorage instance for the current wallet.

get_key_for_did (did: str) → str

Fetch the verkey for a ledger DID.

Parameters **did** – The DID to look up on the ledger or in the cache

get_latest_txn_author_acceptance () → dict

Look up the latest TAA acceptance.

get_nym_role (did: str) → aries_cloudagent.ledger.base.Role

Return the role of the input public DID's NYM on the ledger.

Parameters **did** – DID to query for role on the ledger.

get_revoc_reg_def (revoc_reg_id: str) → dict

Get revocation registry definition by ID; augment with ledger timestamp.

get_revoc_reg_delta (revoc_reg_id: str, fro=0, to=None) -> (<class 'dict'>, <class 'int'>)

Look up a revocation registry delta by ID.

:param revoc_reg_id revocation registry id :param fro earliest EPOCH time of interest :param to latest EPOCH time of interest

:returns delta response, delta timestamp

get_revoc_reg_entry (revoc_reg_id: str, timestamp: int)

Get revocation registry entry by revocation registry ID and timestamp.

get_schema (schema_id: str) → dict

Get a schema from the cache if available, otherwise fetch from the ledger.

Parameters **schema_id** – The schema id (or stringified sequence number) to retrieve

get_txn_author_agreement (reload: bool = False) → dict

Get the current transaction author agreement, fetching it if necessary.

nym_to_did (nym: str) → str

Format a nym with the ledger's DID prefix.

pool_handle

Accessor for the ledger pool handle.

pool_name

Accessor for the ledger pool name.

read_only

Accessor for the ledger read-only flag.

register_nym (did: str, verkey: str, alias: str = None, role: str = None)

Register a nym on the ledger.

Parameters

- **did** – DID to register on the ledger.
- **verkey** – The verification key of the keypair.
- **alias** – Human-friendly alias to assign to the DID.
- **role** – For permissioned ledgers, what role should the new DID have.

rotate_public_did_keypair (next_seed: str = None) → None

Rotate keypair for public DID: create new key, submit to ledger, update wallet.

Parameters `next_seed` – seed for incoming ed25519 keypair (default random)

send_revoc_reg_def (*revoc_reg_def: dict, issuer_did: str = None*)

Publish a revocation registry definition to the ledger.

send_revoc_reg_entry (*revoc_reg_id: str, revoc_def_type: str, revoc_reg_entry: dict, issuer_did: str = None*)

Publish a revocation registry entry to the ledger.

taa_digest (*version: str, text: str*)

Generate the digest of a TAA record.

taa_rough_timestamp () → int

Get a timestamp accurate to the day.

Anything more accurate is a privacy concern.

update_endpoint_for_did (*did: str, endpoint: str, endpoint_type: aries_cloudagent.ledger.endpoint_type.EndpointType = None*) → bool

Check and update the endpoint on the ledger.

Parameters

- **did** – The ledger DID
- **endpoint** – The endpoint address
- **endpoint_type** – The type of the endpoint

```
class aries_cloudagent.ledger.indy.IndySdkLedgerPool (name: str, *, checked: bool = False, keepalive: int = 0, cache: aries_cloudagent.cache.base.BaseCache = None, cache_duration: int = 600, genesis_transactions: str = None, read_only: bool = False)
```

Bases: `object`

Indy ledger manager class.

check_pool_config () → bool

Check if a pool config has been created.

close ()

Close the pool ledger.

context_close ()

Release the reference and schedule closing of the pool ledger.

context_open ()

Open the ledger if necessary and increase the number of active references.

create_pool_config (*genesis_transactions: str, recreate: bool = False*)

Create the pool ledger configuration.

open ()

Open the pool ledger, creating it if necessary.

```
class aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider
```

Bases: `aries_cloudagent.config.base.BaseProvider`

Indy ledger pool provider which keys off the selected pool name.

provide (*settings:* `aries_cloudagent.config.base.BaseSettings`, *injector:* `aries_cloudagent.config.base.BaseInjector`)
Create and open the pool instance.

aries_cloudagent.ledger.routes module

Ledger admin routes.

class `aries_cloudagent.ledger.routes.AMLRecordSchema (*args, **kwargs)`
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Ledger AML record.

`aml = <fields.Dict (default=<marshmallow.missing>, attribute=None, validate=None, requi`

`amlContext = <fields.String (default=<marshmallow.missing>, attribute=None, validate=No`

`version = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None,`

class `aries_cloudagent.ledger.routes.LedgerModulesResultSchema (*args, **kwargs)`
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Schema for the modules endpoint.

class `aries_cloudagent.ledger.routes.QueryStringDIDSchema (*args, **kwargs)`
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Parameters and validators for query string with DID only.

`did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries_cl`

class `aries_cloudagent.ledger.routes.QueryStringEndpointSchema (*args, **kwargs)`
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Parameters and validators for query string with DID and endpoint type.

`did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries_cl`

`endpoint_type = <fields.String (default=<marshmallow.missing>, attribute=None, validate=`

class `aries_cloudagent.ledger.routes.RegisterLedgerNymQueryStringSchema (*args, **kwargs)`
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Query string parameters and validators for register ledger nym request.

`alias = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None, r`

`did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries_cl`

`role = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<OneOf(c`

`verkey = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries`

class `aries_cloudagent.ledger.routes.TAAAcceptSchema (*args, **kwargs)`
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Input schema for accepting the TAA.

`mechanism = <fields.String (default=<marshmallow.missing>, attribute=None, validate=Non`

`text = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None, re`

`version = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None,`

```
class aries_cloudagent.ledger.routes.TAAAcceptanceSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

TAA acceptance record.

```
mechanism = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
time = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=<aries_
```

```
class aries_cloudagent.ledger.routes.TAAInfoSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Transaction author agreement info.

```
aml_record = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
taa_accepted = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
taa_record = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
taa_required = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.ledger.routes.TAARecordSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Ledger TAA record.

```
digest = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_
```

```
text = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
```

```
version = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
class aries_cloudagent.ledger.routes.TAAResultSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for a transaction author agreement.

```
result = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, if_
```

```
aries_cloudagent.ledger.routes.get_did_endpoint (request:
```

```
<sphinx.ext.autodoc.importer._MockObject  
object at 0x7f760e35ee90>)
```

Request handler for getting a verkey for a DID from the ledger.

Parameters **request** – aiohttp request object

```
aries_cloudagent.ledger.routes.get_did_verkey (request: <sphinx.ext.autodoc.importer._MockObject  
object at 0x7f760e35ee90>)
```

Request handler for getting a verkey for a DID from the ledger.

Parameters **request** – aiohttp request object

```
aries_cloudagent.ledger.routes.get_nym_role (request: <sphinx.ext.autodoc.importer._MockObject  
object at 0x7f760e35ee90>)
```

Request handler for getting the role from the NYM registration of a public DID.

Parameters **request** – aiohttp request object

```
aries_cloudagent.ledger.routes.ledger_accept_taa (request:
```

```
<sphinx.ext.autodoc.importer._MockObject  
object at 0x7f760e35ee90>)
```

Request handler for accepting the current transaction author agreement.

Parameters **request** – aiohttp request object

Returns The DID list response

`aries_cloudagent.ledger.routes.ledger_get_taa` (*request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e35ee90>*)

Request handler for fetching the transaction author agreement.

Parameters `request` – aiohttp request object

Returns The TAA information including the AML

`aries_cloudagent.ledger.routes.post_process_routes` (*app: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e35ee90>*)

Amend swagger API.

`aries_cloudagent.ledger.routes.register` (*app: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e35ee90>*)

Register routes.

`aries_cloudagent.ledger.routes.register_ledger_nym` (*request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e35ee90>*)

Request handler for registering a NYM with the ledger.

Parameters `request` – aiohttp request object

`aries_cloudagent.ledger.routes.rotate_public_did_keypair` (*request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e35ee90>*)

Request handler for rotating key pair associated with public DID.

Parameters `request` – aiohttp request object

`aries_cloudagent.ledger.util` module

Ledger utilities.

`aries_cloudagent.messaging` package

Subpackages

`aries_cloudagent.messaging.ack` package

Submodules

`aries_cloudagent.messaging.ack.message` module

Represents an explicit ack message as per Aries RFC 15.

class `aries_cloudagent.messaging.ack.message.Ack` (*status: str = None, **kwargs*)
 Bases: `aries_cloudagent.messaging.agent_message.AgentMessage`

Base class representing an explicit ack message.

Subclass to adopt, specify Meta message type and handler class.

class `Meta`
 Bases: `object`

Ack metadata.

```
schema_class = 'AckSchema'
```

```
class aries_cloudagent.messaging.ack.message.AckSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
```

Schema for Ack base class.

```
class Meta
```

Bases: object

Ack schema metadata.

model_class

alias of *Ack*

```
unknown = 'exclude'
```

```
status = <fields.Constant(default='OK', attribute=None, validate=None, required=True,
```

aries_cloudagent.messaging.credential_definitions package

Submodules

aries_cloudagent.messaging.credential_definitions.routes module

Credential definition admin routes.

```
class aries_cloudagent.messaging.credential_definitions.routes.CredDefIdMatchInfoSchema(*args, **kwargs):
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
```

Path parameters and validators for request taking cred def id.

```
cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
```

```
class aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionGetResu
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
```

Results schema for schema get request.

```
credential_definition = <fields.Nested(default=<marshmallow.missing>, attribute=None, ...
```

```
class aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionSchema(Base):
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
```

Credential definition schema.

```
ident = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
```

```
schemaId = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
```

```
tag = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, req
```

```
typ = <fields.Constant (default='CL', attribute=None, validate=None, required=False, lo
```

```
value = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, req
```

```
ver = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
```

class aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionSendReq

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for schema send request.

revocation_registry_size = <fields.Integer(default=<marshmallow.missing>, attribute=None

schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar

support_revocation = <fields.Boolean(default=<marshmallow.missing>, attribute=None, va

tag = <fields.String(default='default', attribute=None, validate=None, required=False,

class aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionSendRes

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Results schema for schema send request.

credential_definition_id = <fields.String(default=<marshmallow.missing>, attribute=None

class aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionsCreate

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Results schema for cred-defs-created request.

credential_definition_ids = <fields.List(default=<marshmallow.missing>, attribute=None

aries_cloudagent.messaging.credential_definitions.routes.credential_definitions_created (req

<sp
ob-
ject
at
0x7

Request handler for retrieving credential definitions that current agent created.

Parameters **request** – aiohttp request object

Returns The identifiers of matching credential definitions.

aries_cloudagent.messaging.credential_definitions.routes.credential_definitions_get_credent

Request handler for getting a credential definition from the ledger.

Parameters **request** – aiohttp request object

Returns The credential definition details.

aries_cloudagent.messaging.credential_definitions.routes.credential_definitions_send_creden

Request handler for sending a credential definition to the ledger.

Parameters **request** – aiohttp request object

Returns The credential definition identifier

```
aries_cloudagent.messaging.credential_definitions.routes.post_process_routes (app:
    <sphinx.ext.autodoc
    ob-
    ject
    at
    0x7f760e2aa990>)
```

Amend swagger API.

```
aries_cloudagent.messaging.credential_definitions.routes.register (app:
    <sphinx.ext.autodoc.importer._Mock
    object    at
    0x7f760e2aa990>)
```

Register routes.

aries_cloudagent.messaging.credential_definitions.util module

Credential definition utilities.

```
class aries_cloudagent.messaging.credential_definitions.util.CredDefQueryStringSchema (*args,
    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Query string parameters for credential definition searches.
    cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
    issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
    schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
    schema_issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, vali
    schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=N
    schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

aries_cloudagent.messaging.decorators package

Submodules

aries_cloudagent.messaging.decorators.attach_decorator module

A message decorator for attachments.

An attach decorator embeds content or specifies appended content.


```

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator(*,
                                                                            ident:
                                                                              str
                                                                              =
                                                                              None,
                                                                              de-
                                                                              scrip-
                                                                              tion:
                                                                              str
                                                                              =
                                                                              None,
                                                                              file-
                                                                              name:
                                                                              str
                                                                              =
                                                                              None,
                                                                              mime_type:
                                                                              str
                                                                              =
                                                                              None,
                                                                              last-
                                                                              mod_time:
                                                                              str
                                                                              =
                                                                              None,
                                                                              byte_count:
                                                                              int
                                                                              =
                                                                              None,
                                                                              data:
                                                                              aries_cloudagent.m
                                                                              **kwargs)

```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing attach decorator.

class Meta

Bases: *object*

AttachDecorator metadata.

schema_class = 'AttachDecoratorSchema'

content

Return attachment content.

Returns: data attachment, decoded if necessary and json-loaded, or data links

classmethod data_base64 (*mapping: Mapping[KT, VT_co], *, ident: str = None, description: str = None, filename: str = None, lastmod_time: str = None, byte_count: int = None*)

Create *AttachDecorator* instance on base64-encoded data from input mapping.

Given mapping, JSON dump, base64-encode, and embed it as data; mark *application/json* MIME type.

Parameters

- **mapping** – (dict) data structure; e.g., indy production
- **ident** – optional attachment identifier (default random UUID4)

- **description** – optional attachment description
- **filename** – optional attachment filename
- **lastmod_time** – optional attachment last modification time
- **byte_count** – optional attachment byte count

classmethod data_json (*mapping: dict, *, ident: str = None, description: str = None, filename: str = None, lastmod_time: str = None, byte_count: int = None*)

Create *AttachDecorator* instance on json-encoded data from input mapping.

Given message object (dict), JSON dump, and embed it as data; mark *application/json* MIME type.

Parameters

- **mapping** – (dict) data structure; e.g., Aries message
- **ident** – optional attachment identifier (default random UUID4)
- **description** – optional attachment description
- **filename** – optional attachment filename
- **lastmod_time** – optional attachment last modification time
- **byte_count** – optional attachment byte count

classmethod data_links (*links: Union[str, Sequence[str]], sha256: str = None, *, ident: str = None, mime_type: str = None, description: str = None, filename: str = None, lastmod_time: str = None, byte_count: int = None*)

Create *AttachDecorator* instance on json-encoded data from input mapping.

Given message object (dict), JSON dump, and embed it as data; mark *application/json* MIME type.

Parameters

- **links** – URL or list of URLs
- **sha256** – optional sha-256 hash for content
- **ident** – optional attachment identifier (default random UUID4)
- **mime_type** – optional MIME type
- **description** – optional attachment description
- **filename** – optional attachment filename
- **lastmod_time** – optional attachment last modification time
- **byte_count** – optional attachment byte count

```

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorData(*,
    jws_:
    aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorData
    =
    None,
    sha256_:
    str
    =
    None,
    links_:
    Union[Sequence[str], None]
    =
    None,
    base64_:
    str
    =
    None,
    json_:
    dict
    =
    None)

```

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Attach decorator data.

class Meta

Bases: `object`

AttachDecoratorData metadata.

schema_class = 'AttachDecoratorDataSchema'

base64

Accessor for base64 decorator data, or None.

header_map (*idx: int = 0, jose: bool = True*) → Mapping[KT, VT_co]

Accessor for header info at input index, default 0 or unique for singly-signed.

Parameters

- **idx** – index of interest, zero-based (default 0)
- **jose** – True to return unprotected header attributes, False for protected only

json

Accessor for json decorator data, or None.

jws

Accessor for JWS, or None.

links

Accessor for links decorator data, or None.

sha256

Accessor for sha256 decorator data, or None.

sign (*verkeys: Union[str, Sequence[str]], wallet: aries_cloudagent.wallet.base.BaseWallet*)

Sign base64 data value of attachment.

Parameters

- **verkeys** – verkey(s) of the signing party (in raw or DID key format)
- **wallet** – The wallet to use for the signature

signatures

Accessor for number of signatures.

signed

Accessor for signed content (payload), None for unsigned.

verify (*wallet*: *aries_cloudagent.wallet.base.BaseWallet*) → bool
Verify the signature(s).

Parameters **wallet** – Wallet to use to verify signature

Returns True if verification succeeds else False

```
class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorData1JWS (*,
header: aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorData1JWS.header,
protected: str,
signature: str)
= None,
signature: str)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Single Detached JSON Web Signature for inclusion in attach decorator data.

class Meta

Bases: *object*

AttachDecoratorData1JWS metadata.

schema_class = 'AttachDecoratorData1JWSSchema'

```
class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorData1JWSSchema
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Single attach decorator data JWS schema.

class Meta

Bases: *object*

Single attach decorator data JWS schema metadata.

model_class

alias of *AttachDecoratorData1JWS*

unknown = 'exclude'

header = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, ...)

protected = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar...

signature = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar...

```

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataJWS (*,
header:
aries_clo
=
None,
pro-
tected:
str
=
None,
sig-
na-
ture:
str
=
None,
sig-
na-
tures:
Se-
quence[a
=
None)

```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Detached JSON Web Signature for inclusion in attach decorator data.

May hold one signature in flattened format, or multiple signatures in the “signatures” member.

class Meta

Bases: *object*

AttachDecoratorDataJWS metadata.

schema_class = 'AttachDecoratorDataJWSSchema'

```

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataJWSHeader (k
S

```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Attach decorator data JWS header.

class Meta

Bases: *object*

AttachDecoratorDataJWS metadata.

schema_class = 'AttachDecoratorDataJWSHeaderSchema'

```

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataJWSHeaderSchema

```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Attach decorator data JWS header schema.

class Meta

Bases: *object*

Attach decorator data schema metadata.

model_class

alias of *AttachDecoratorDataJWSHeader*

```
        unknown = 'exclude'

    kid = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataJWSSchema (
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema
    Schema for detached JSON Web Signature for inclusion in attach decorator data.

    class Meta
        Bases: object
        Metadata for schema for detached JWS for inclusion in attach deco data.

        model_class
            alias of AttachDecoratorDataJWS

        unknown = 'exclude'

    header = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,
    protected = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
    signature = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
    signatures = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None
    validate_single_xor_multi_sig (data: Mapping[KT, VT_co], **kwargs)
        Ensure model is for either 1 or many sigatures, not mishmash of both.

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataSchema (*args,
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema
    Attach decorator data schema.

    class Meta
        Bases: object
        Attach decorator data schema metadata.

        model_class
            alias of AttachDecoratorData

        unknown = 'exclude'

    base64_ = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<arie
    json_ = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, req
    jws_ = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, re
    links_ = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, re
    sha256_ = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<arie
    validate_data_spec (data: Mapping[KT, VT_co], **kwargs)
        Ensure model chooses exactly one of base64, json, or links.

class aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema (*args,
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema
    Attach decorator schema used in serialization/deserialization.
```

```

class Meta
    Bases: object

    AttachDecoratorSchema metadata.

    model_class
        alias of AttachDecorator

    unknown = 'exclude'

byte_count = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None)
data = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, re
description = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
filename = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
ident = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
lastmod_time = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
mime_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)

aries_cloudagent.messaging.decorators.attach_decorator.did_key(verkey: str) →
    str
    Qualify verkey into DID key if need be.

aries_cloudagent.messaging.decorators.attach_decorator.raw_key(verkey: str) →
    str
    Strip qualified key to raw key if need be.

```

aries_cloudagent.messaging.decorators.base module

Classes for managing a collection of decorators.

```

class aries_cloudagent.messaging.decorators.base.BaseDecoratorSet (models:
    dict      =
    None)

    Bases: collections.OrderedDict

    Collection of decorators.

    add_model (key: str; model: Type[aries_cloudagent.messaging.models.base.BaseModel])
        Add a registered decorator model.

    copy () → aries_cloudagent.messaging.decorators.base.BaseDecoratorSet
        Return a copy of the decorator set.

    extract_decorators (message: Mapping[KT, VT_co], schema:
        Type[<sphinx.ext.autodoc.importer._MockObject object at
        0x7f760f57ee90>] = None, serialized: bool = True, skip_attrs: Se-
        quence[str] = None) → collections.OrderedDict
        Extract decorators and return the remaining properties.

    field (name: str) → aries_cloudagent.messaging.decorators.base.BaseDecoratorSet
        Access a named decorated field.

    fields
        Accessor for the set of currently defined fields.

    has_field (name: str) → bool
        Check for the existence of a named decorator field.

```

load_decorator (*key: str, value, serialized=False*)
 Convert a decorator value to its loaded representation.

models
 Accessor for the models dictionary.

prefix
 Accessor for the decorator prefix.

remove_field (*name: str*)
 Remove a named decorated field.

remove_model (*key: str*)
 Remove a registered decorator model.

to_dict (*prefix: str = None*) → collections.OrderedDict
 Convert to a dictionary (serialize).

Raises BaseModelError – on decorator validation errors

exception aries_cloudagent.messaging.decorators.base.DecoratorError (**args, error_code: str = None, **kwargs*)

Bases: *aries_cloudagent.core.error.BaseError*

Base error for decorator issues.

aries_cloudagent.messaging.decorators.default module

Default decorator set implementation.

class aries_cloudagent.messaging.decorators.default.DecoratorSet (*models: dict = None*)

Bases: *aries_cloudagent.messaging.decorators.base.BaseDecoratorSet*

Default decorator set implementation.

aries_cloudagent.messaging.decorators.localization_decorator module

The localization decorator (~110n) for message localization information.


```
class aries_cloudagent.messaging.decorators.localization_decorator.LocalizationDecorator(*,
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing the localization decorator.

```
class Meta
```

Bases: *object*

LocalizationDecorator metadata.

```
schema_class = 'LocalizationDecoratorSchema'
```

```
class aries_cloudagent.messaging.decorators.localization_decorator.LocalizationDecoratorSchema
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Localization decorator schema used in serialization/deserialization.

```
class Meta
```

Bases: *object*

LocalizationDecoratorSchema metadata.

```
model_class
```

alias of *LocalizationDecorator*

```
unknown
```

Used by autodoc_mock_imports.

```
catalogs
```

Used by autodoc_mock_imports.

```
locale
```

Used by autodoc_mock_imports.

```
localizable
```

Used by autodoc_mock_imports.

aries_cloudagent.messaging.decorators.please_ack_decorator module

The please-ack decorator to request acknowledgement.

```
class aries_cloudagent.messaging.decorators.please_ack_decorator.PleaseAckDecorator (message_id: str, on: None, sequence: Sequence[str])
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing the please-ack decorator.

```
class Meta
```

Bases: *object*

PleaseAckDecorator metadata.

```
schema_class = 'PleaseAckDecoratorSchema'
```

```
class aries_cloudagent.messaging.decorators.please_ack_decorator.PleaseAckDecoratorSchema (message_id: str, on: Sequence[str])
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

PleaseAck decorator schema used in serialization/deserialization.

```
class Meta
```

Bases: *object*

PleaseAckDecoratorSchema metadata.

```
model_class
```

alias of *PleaseAckDecorator*

```
unknown = 'exclude'
```

```
message_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
on = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
```

aries_cloudagent.messaging.decorators.signature_decorator module

Model and schema for working with field signatures within message bodies.

```

class aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator(*,
signature_type:
str
=
None,
signature_data:
str
=
None,
signer:
str
=
None)

```

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Class representing a field value signed by a known verkey.

class Meta

Bases: `object`

SignatureDecorator metadata.

schema_class = 'SignatureDecoratorSchema'

TYPE_ED25519SHA512 = 'signature/1.0/ed25519Sha512_single'

classmethod create (value, signer: str, wallet: aries_cloudagent.wallet.base.BaseWallet, timestamp=None) → aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator

Create a Signature.

Sign a field value and return a newly constructed *SignatureDecorator* representing the resulting signature.

Parameters

- **value** – Value to sign
- **signer** – Verkey of the signing party
- **wallet** – The wallet to use for the signature

Returns The created *SignatureDecorator* object

decode () -> (<class 'object'>, <class 'int'>)

Decode the signature to its timestamp and value.

Returns A tuple of (decoded message, timestamp)

verify (wallet: aries_cloudagent.wallet.base.BaseWallet) → bool

Verify the signature against the signer's public key.

Parameters **wallet** – Wallet to use to verify signature

Returns True if verification succeeds else False

```
class aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecoratorSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema
    SignatureDecorator schema.

    class Meta
        Bases: object
        SignatureDecoratorSchema metadata.

    model_class
        alias of SignatureDecorator

    unknown
        Used by autodoc_mock_imports.

    sig_data
        Used by autodoc_mock_imports.

    signature
        Used by autodoc_mock_imports.

    signature_type
        Used by autodoc_mock_imports.

    signer
        Used by autodoc_mock_imports.
```

aries_cloudagent.messaging.decorators.thread_decorator module

A message decorator for threads.

A thread decorator identifies a message that may require additional context from previous messages.

```
class aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecorator (*args, **kwargs)
    thid: str = None,
    pthid: str = None,
    sender_order: int = None,
    received_orders: Mapping[str, VT_co] = None

    Bases: aries_cloudagent.messaging.models.base.BaseModel
    Class representing thread decorator.
```

```

class Meta
    Bases: object
    ThreadDecorator metadata.
    schema_class = 'ThreadDecoratorSchema'

pthid
    Accessor for parent thread identifier.
    Returns This thread's pthid

received_orders
    Get received orders.
    Returns The highest sender_order value that the sender has seen from other sender(s) on the
    thread.

sender_order
    Get sender order.
    Returns A number that tells where this message fits in the sequence of all messages that the
    current sender has contributed to this thread

thid
    Accessor for thread identifier.
    Returns This thread's thid

class aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecoratorSchema(*args,
                                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema
    Thread decorator schema used in serialization/deserialization.

class Meta
    Bases: object
    ThreadDecoratorSchema metadata.
    model_class
        alias of ThreadDecorator
    unknown
        Used by autodoc_mock_imports.

pthid
    Used by autodoc_mock_imports.

received_orders
    Used by autodoc_mock_imports.

sender_order
    Used by autodoc_mock_imports.

thid
    Used by autodoc_mock_imports.

```

aries_cloudagent.messaging.decorators.timing_decorator module

The timing decorator (~timing).

This decorator allows the timing of agent messages to be communicated and constrained.

```

class aries_cloudagent.messaging.decorators.timing_decorator.TimingDecorator(*
    in_time:
    Union[str,
    date-
    time.datetime]
    =
    None,
    out_time:
    Union[str,
    date-
    time.datetime]
    =
    None,
    stale_time:
    Union[str,
    date-
    time.datetime]
    =
    None,
    expires_time:
    Union[str,
    date-
    time.datetime]
    =
    None,
    delay_milli:
    int
    =
    None,
    wait_until_time:
    Union[str,
    date-
    time.datetime]
    =
    None)

```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing the timing decorator.

```

class Meta
    Bases: object
    TimingDecorator metadata.
    schema_class = 'TimingDecoratorSchema'

```

```

class aries_cloudagent.messaging.decorators.timing_decorator.TimingDecoratorSchema(*args,
    **kwargs)

```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Timing decorator schema used in serialization/deserialization.

```

class Meta
    Bases: object
    TimingDecoratorSchema metadata.

```

model_class
alias of *TimingDecorator*

unknown
Used by autodoc_mock_imports.

delay_milli
Used by autodoc_mock_imports.

expires_time
Used by autodoc_mock_imports.

in_time
Used by autodoc_mock_imports.

out_time
Used by autodoc_mock_imports.

stale_time
Used by autodoc_mock_imports.

wait_until_time
Used by autodoc_mock_imports.

aries_cloudagent.messaging.decorators.trace_decorator module

A message decorator for trace events.

A trace decorator identifies a responsibility on the processor to record information on message processing events.

```
class aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator (*,
                                                                    tar-
                                                                    get:
                                                                    str
                                                                    =
                                                                    None,
                                                                    full_thread:
                                                                    bool
                                                                    =
                                                                    True,
                                                                    trace_reports:
                                                                    Se-
                                                                    quence[T_co]
                                                                    =
                                                                    None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing trace decorator.

```
class Meta
    Bases: object
    TraceDecorator metadata.
    schema_class = 'TraceDecoratorSchema'
```

```
append_trace_report (trace_report: aries_cloudagent.messaging.decorators.trace_decorator.TraceReport)
    Append a trace report to this decorator.
```

```
full_thread
    Accessor for full_thread flag.
```

Returns The full_thread flag

target

Accessor for trace target.

Returns The target for tracing messages

trace_reports

Set of trace reports for this message.

Returns The trace reports that have been logged on this message/thread so far. (Only for target="message".)

class aries_cloudagent.messaging.decorators.trace_decorator.**TraceDecoratorSchema** (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Trace decorator schema used in serialization/deserialization.

class **Meta**

Bases: *object*

TraceDecoratorSchema metadata.

model_class

alias of *TraceDecorator*

unknown

Used by autodoc_mock_imports.

full_thread

Used by autodoc_mock_imports.

target

Used by autodoc_mock_imports.

trace_reports

Used by autodoc_mock_imports.


```
class aries_cloudagent.messaging.decorators.trace_decorator.TraceReport (*,  
                                                                    msg_id:  
                                                                    str  
                                                                    =  
                                                                    None,  
                                                                    thread_id:  
                                                                    str  
                                                                    =  
                                                                    None,  
                                                                    traced_type:  
                                                                    str  
                                                                    =  
                                                                    None,  
                                                                    times-  
                                                                    tamp:  
                                                                    str  
                                                                    =  
                                                                    None,  
                                                                    str_time:  
                                                                    str  
                                                                    =  
                                                                    None,  
                                                                    han-  
                                                                    dler:  
                                                                    str  
                                                                    =  
                                                                    None,  
                                                                    el-  
                                                                    lapsed_milli:  
                                                                    int  
                                                                    =  
                                                                    None,  
                                                                    out-  
                                                                    come:  
                                                                    str  
                                                                    =  
                                                                    None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a Trace Report.

class Meta

Bases: `object`

TraceReport metadata.

schema_class = 'TraceReport'

ellapsed_milli

Accessor for ellapsed_milli.

Returns The sender ellapsed_milli

handler

Accessor for handler.

Returns The sender handler

msg_id

Accessor for msg_id.

Returns The msg_id

outcome

Accessor for outcome.

Returns The sender outcome

str_time

Accessor for str_time.

Returns Formatted representation of the sender timestamp

thread_id

Accessor for thread_id.

Returns The thread_id

timestamp

Accessor for timestamp.

Returns The sender timestamp

traced_type

Accessor for traced_type.

Returns The sender traced_type

class aries_cloudagent.messaging.decorators.trace_decorator.**TraceReportSchema** (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Trace report schema.

class Meta

Bases: *object*

TraceReportSchema metadata.

model_class

alias of *TraceReport*

unknown

Used by autodoc_mock_imports.

elapseded_milli

Used by autodoc_mock_imports.

handler

Used by autodoc_mock_imports.

msg_id

Used by autodoc_mock_imports.

outcome

Used by autodoc_mock_imports.

str_time

Used by autodoc_mock_imports.

thread_id

Used by autodoc_mock_imports.

timestamp
Used by autodoc_mock_imports.

traced_type
Used by autodoc_mock_imports.

aries_cloudagent.messaging.decorators.transport_decorator module

The transport decorator (~transport).

This decorator allows changes to agent response behaviour and queue status updates.

```
class aries_cloudagent.messaging.decorators.transport_decorator.TransportDecorator (*,
    re-
    turn_route:
    str
    =
    None,
    re-
    turn_route:
    str
    =
    None,
    queued_me
    int
    =
    None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing the transport decorator.

```
class Meta
    Bases: object
    TransportDecorator metadata.
    schema_class = 'TransportDecoratorSchema'
```

```
class aries_cloudagent.messaging.decorators.transport_decorator.TransportDecoratorSchema (*,
    **)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Transport decorator schema used in serialization/deserialization.

```
class Meta
    Bases: object
    TransportDecoratorSchema metadata.
    model_class
        alias of TransportDecorator
    unknown
        Used by autodoc_mock_imports.
    queued_message_count
        Used by autodoc_mock_imports.
    return_route
        Used by autodoc_mock_imports.
```

return_route_thread

Used by autodoc_mock_imports.

aries_cloudagent.messaging.jsonld package

Submodules

aries_cloudagent.messaging.jsonld.create_verify_data module

Contains the functions needed to produce and verify a json-ld signature.

This file was ported from <https://github.com/transmute-industries/Ed25519Signature2018/blob/master/src/createVerifyData/index.js>

exception aries_cloudagent.messaging.jsonld.create_verify_data.**DroppedAttributeException**
Bases: **Exception**

Exception used to track that an attribute was removed.

aries_cloudagent.messaging.jsonld.create_verify_data.**create_verify_data**(data, signature_options)
Encapsulate the process of constructing the string used during sign and verify.

aries_cloudagent.messaging.jsonld.credential module

Sign and verify functions for json-ld based credentials.

aries_cloudagent.messaging.jsonld.credential.**b64decode**(bytes)
Url Safe B64 Decode.

aries_cloudagent.messaging.jsonld.credential.**b64encode**(str)
Url Safe B64 Encode.

aries_cloudagent.messaging.jsonld.credential.**create_jws**(encoded_header, verify_data)
Compose JWS.

aries_cloudagent.messaging.jsonld.credential.**did_key**(verkey: str) → str
Qualify verkey into DID key if need be.

aries_cloudagent.messaging.jsonld.credential.**jws_sign**(verify_data, verkey, wallet)
Sign JWS.

aries_cloudagent.messaging.jsonld.credential.**jws_verify**(verify_data, signature, public_key, wallet)
Detatched jws verify handling.

aries_cloudagent.messaging.jsonld.credential.**sign_credential**(credential, signature_options, verkey, wallet)
Sign Credential.

aries_cloudagent.messaging.jsonld.credential.**verify_credential**(doc, verkey, wallet)
Verify credential.

```
aries_cloudagent.messaging.jsonld.credential.verify_jws_header(header)
    Check header requirements.
```

aries_cloudagent.messaging.jsonld.routes module

jsonld admin routes.

```

class aries_cloudagent.messaging.jsonld.routes.SignRequestSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for signing a jsonld doc.

    doc = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    verkey = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>

class aries_cloudagent.messaging.jsonld.routes.SignResponseSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Response schema for a signed jsonld doc.

    signed_doc = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>

class aries_cloudagent.messaging.jsonld.routes.VerifyRequestSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for signing a jsonld doc.

    doc = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    verkey = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>

class aries_cloudagent.messaging.jsonld.routes.VerifyResponseSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Response schema for verification result.

    valid = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>

aries_cloudagent.messaging.jsonld.routes.register(app:
                                                <sphinx.ext.autodoc.importer._MockObject
                                                object at 0x7f760ddd9c50>)

    Register routes.

aries_cloudagent.messaging.jsonld.routes.sign(request: <sphinx.ext.autodoc.importer._MockObject
                                                object at 0x7f760ddd9c50>)

    Request handler for signing a jsonld doc.

    Parameters request – aiohttp request object

aries_cloudagent.messaging.jsonld.routes.verify(request:
                                                <sphinx.ext.autodoc.importer._MockObject
                                                object at 0x7f760ddd9c50>)

    Request handler for signing a jsonld doc.

    Parameters request – aiohttp request object

```

aries_cloudagent.messaging.models package

Submodules

aries_cloudagent.messaging.models.base module

Base classes for Models and Schemas.

class aries_cloudagent.messaging.models.base.**BaseModel**

Bases: `abc.ABC`

Base model that provides convenience methods.

class **Meta**

Bases: `object`

BaseModel meta data.

schema_class = `None`

Schema

Accessor for the model's schema class.

Returns The schema class

classmethod **deserialize** (*obj*)

Convert from JSON representation to a model instance.

Parameters **obj** – The dict to load into a model instance

Returns A model instance for this data

classmethod **from_json** (*json_repr: Union[str, bytes]*)

Parse a JSON string into a model instance.

Parameters **json_repr** – JSON string

Returns A model instance representation of this JSON

serialize (*as_string=False*) → dict

Create a JSON-compatible dict representation of the model instance.

Parameters **as_string** – Return a string of JSON instead of a dict

Returns A dict representation of this model, or a JSON string if *as_string* is True

to_json () → str

Create a JSON representation of the model instance.

Returns A JSON representation of this message

validate ()

Validate a constructed model.

exception aries_cloudagent.messaging.models.base.**BaseModelError** (**args, error_code: str = None, **kwargs*)

Bases: `aries_cloudagent.core.error.BaseError`

Base exception class for base model errors.

```

class aries_cloudagent.messaging.models.base.BaseModelSchema(*args, **kwargs)
    Bases: sphinx.ext.autodoc.importer._MockObject

    BaseModel schema.

    class Meta
        Bases: object

        BaseModelSchema metadata.

        model_class = None

        ordered = True

        skip_values = [None]

    Model
        Accessor for the schema's model class.

        Returns The model class

    make_model(data: dict, **kwargs)
        Return model instance after loading.

        Returns A model instance

    remove_skipped_values(data, **kwargs)
        Remove values that are are marked to skip.

        Returns Returns this modified data

    skip_dump_only(data, **kwargs)
        Skip fields that are only expected during serialization.

        Parameters data – The incoming data to clean

        Returns The modified data

class aries_cloudagent.messaging.models.base.OpenAPISchema(*args, **kwargs)
    Bases: sphinx.ext.autodoc.importer._MockObject

    Schema for OpenAPI artifacts: excluding unknown fields, not raising exception.

    class Meta
        Bases: object

        BaseModelSchema metadata.

        unknown
            Used by autodoc_mock_imports.

aries_cloudagent.messaging.models.base.resolve_class(the_cls, relative_cls: type =
                                                    None)

    Resolve a class.

    Parameters

        • the_cls – The class to resolve

        • relative_cls – Relative class to resolve from

    Returns The resolved class

    Raises ClassNotFoundError – If the class could not be loaded

aries_cloudagent.messaging.models.base.resolve_meta_property(obj, prop_name:
                                                            str, defval=None)

    Resolve a meta property.

```

Parameters

- **prop_name** – The property to resolve
- **defval** – The default value

Returns The meta property

aries_cloudagent.messaging.models.base_record module

Classes for BaseStorage-based record management.

```
class aries_cloudagent.messaging.models.base_record.BaseExchangeRecord (id:
                                                                    str
                                                                    =
                                                                    None,
                                                                    state:
                                                                    str
                                                                    =
                                                                    None,
                                                                    *,
                                                                    trace:
                                                                    bool
                                                                    =
                                                                    False,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Represents a base record with event tracing capability.

```
class aries_cloudagent.messaging.models.base_record.BaseExchangeSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

Base schema for exchange records.

class Meta

Bases: *object*

BaseExchangeSchema metadata.

model_class

alias of *BaseExchangeRecord*

trace

Used by autodoc_mock_imports.

```
class aries_cloudagent.messaging.models.base_record.BaseRecord (id:      str =
                                                                    None,      state:
                                                                    str = None,
                                                                    *,      created_at:
                                                                    Union[str, date-
                                                                    time.datetime]
                                                                    = None,      up-
                                                                    dated_at:
                                                                    Union[str, date-
                                                                    time.datetime]
                                                                    = None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Represents a single storage record.

DEFAULT_CACHE_TTL = 60

LOG_STATE_FLAG = None

class Meta

Bases: `object`

BaseRecord metadata.

RECORD_ID_NAME = 'id'

RECORD_TYPE = None

TAG_NAMES = {'state'}

WEBHOOK_TOPIC = None

classmethod clear_cached_key (*session*: *aries_cloudagent.core.profile.ProfileSession*,
cache_key: *str*)

Shortcut method to clear a cached key value, if any.

Parameters

- **session** – The profile session to use
- **cache_key** – The unique cache identifier

delete_record (*session*: *aries_cloudagent.core.profile.ProfileSession*)

Remove the stored record.

Parameters session – The profile session to use

classmethod from_storage (*record_id*: *str*, *record*: *Mapping[str, Any]*)

Initialize a record from its stored representation.

Parameters

- **record_id** – The unique record identifier
- **record** – The stored representation

classmethod get_cached_key (*session*: *aries_cloudagent.core.profile.ProfileSession*, *cache_key*:
str)

Shortcut method to fetch a cached key value.

Parameters

- **session** – The profile session to use
- **cache_key** – The unique cache identifier

classmethod get_tag_map () → *Mapping[str, str]*

Accessor for the set of defined tags.

classmethod log_state (*msg*: *str*, *params*: *dict* = *None*, *settings*:
aries_cloudagent.config.base.BaseSettings = *None*, *override*: *bool*
= *False*)

Print a message with increased visibility (for testing).

post_save (*session*: *aries_cloudagent.core.profile.ProfileSession*, *new_record*: *bool*, *last_state*: *str*,
webhook: *bool* = *None*)

Perform post-save actions.

Parameters

- **session** – The profile session to use

- **new_record** – Flag indicating if the record was just created
- **last_state** – The previous state value
- **webhook** – Adjust whether the webhook is called

classmethod prefix_tag_filter (*tag_filter: dict*)

Prefix unencrypted tags used in the tag filter.

classmethod query (*session: aries_cloudagent.core.profile.ProfileSession, tag_filter: dict = None, *, post_filter_positive: dict = None, post_filter_negative: dict = None, alt: bool = False*) → Sequence[aries_cloudagent.messaging.models.base_record.BaseRecord]

Query stored records.

Parameters

- **session** – The profile session to use
- **tag_filter** – An optional dictionary of tag filter clauses
- **post_filter_positive** – Additional value filters to apply matching positively
- **post_filter_negative** – Additional value filters to apply matching negatively
- **alt** – set to match any (positive=True) value or miss all (positive=False) values in post_filter

record_tags

Accessor to define implementation-specific tags.

record_value

Accessor to define custom properties for the JSON record value.

classmethod retrieve_by_id (*session: aries_cloudagent.core.profile.ProfileSession, record_id: str*) → aries_cloudagent.messaging.models.base_record.BaseRecord

Retrieve a stored record by ID.

Parameters

- **session** – The profile session to use
- **record_id** – The ID of the record to find

classmethod retrieve_by_tag_filter (*session: aries_cloudagent.core.profile.ProfileSession, tag_filter: dict, post_filter: dict = None*) → aries_cloudagent.messaging.models.base_record.BaseRecord

Retrieve a record by tag filter.

Parameters

- **session** – The profile session to use
- **tag_filter** – The filter dictionary to apply
- **post_filter** – Additional value filters to apply matching positively, with sequence values specifying alternatives to match (hit any)

save (*session: aries_cloudagent.core.profile.ProfileSession, *, reason: str = None, log_params: Mapping[str, Any] = None, log_override: bool = False, webhook: bool = None*) → str

Persist the record to storage.

Parameters

- **session** – The profile session to use
- **reason** – A reason to add to the log

- **log_params** – Additional parameters to log
- **webhook** – Flag to override whether the webhook is sent

send_webhook (*session: aries_cloudagent.core.profile.ProfileSession, payload: Any, topic: str = None*)

Send a standard webhook.

Parameters

- **session** – The profile session to use
- **payload** – The webhook payload
- **topic** – The webhook topic, defaulting to WEBHOOK_TOPIC

classmethod set_cached_key (*session: aries_cloudagent.core.profile.ProfileSession, cache_key: str, value: Any, ttl=None*)

Shortcut method to set a cached key value.

Parameters

- **session** – The profile session to use
- **cache_key** – The unique cache identifier
- **value** – The value to cache
- **ttl** – The cache ttl

storage_record

Accessor for a *StorageRecord* representing this record.

classmethod strip_tag_prefix (*tags: dict*)

Strip tilde from unencrypted tag names.

tags

Accessor for the record tags generated for this record.

value

Accessor for the JSON record value generated for this record.

webhook_payload

Return a JSON-serialized version of the record for the webhook.

webhook_topic

Return the webhook topic value.

class aries_cloudagent.messaging.models.base_record.**BaseRecordSchema** (**args, **kwargs*)

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Schema to allow serialization/deserialization of base records.

class Meta

Bases: *object*

BaseRecordSchema metadata.

model_class = None

created_at

Used by autodoc_mock_imports.

state

Used by autodoc_mock_imports.

updated_at

Used by autodoc_mock_imports.

`aries_cloudagent.messaging.models.base_record.match_post_filter` (*record: dict, post_filter: dict, positive: bool = True, alt: bool = False*) → bool

Determine if a record value matches the post-filter.

Parameters

- **record** – record to check
- **post_filter** – filter to apply (empty or None filter matches everything)
- **positive** – whether matching all filter criteria positively or negatively
- **alt** – set to match any (positive=True) value or miss all (positive=False) values in post_filter

aries_cloudagent.messaging.models.openapi module

Base class for OpenAPI artifact schema.

class `aries_cloudagent.messaging.models.openapi.OpenAPISchema` (*args, **kwargs)
Bases: `sphinx.ext.autodoc.importer._MockObject`

Schema for OpenAPI artifacts: excluding unknown fields, not raising exception.

class Meta

Bases: `object`

OpenAPISchema metadata.

model_class = None

unknown

Used by autodoc_mock_imports.

aries_cloudagent.messaging.schemas package**Submodules****aries_cloudagent.messaging.schemas.routes module**

Credential schema admin routes.

class `aries_cloudagent.messaging.schemas.routes.SchemaGetResultsSchema` (*args, **kwargs)
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Results schema for schema get request.

schema = `<fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, ...)`

class `aries_cloudagent.messaging.schemas.routes.SchemaIdMatchInfoSchema` (*args, **kwargs)
Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Path parameters and validators for request taking schema id.

```
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<Re
```

```
class aries_cloudagent.messaging.schemas.routes.SchemaSchema(*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Content for returned schema.

```
attr_names = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None
```

```
ident = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
```

```
name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
```

```
seqNo = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=<aries
```

```
ver = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
```

```
version = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<arie
```

```
class aries_cloudagent.messaging.schemas.routes.SchemaSendRequestSchema(*args,
                                                                           **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for schema send request.

```
attributes = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None
```

```
schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=N
```

```
schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validat
```

```
class aries_cloudagent.messaging.schemas.routes.SchemaSendResultsSchema(*args,
                                                                           **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Results schema for schema send request.

```
schema = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, re
```

```
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
class aries_cloudagent.messaging.schemas.routes.SchemasCreatedResultsSchema(*args,
                                                                               **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Results schema for a schemas-created request.

```
schema_ids = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None
```

```
aries_cloudagent.messaging.schemas.routes.post_process_routes(app:
                                                                <sphinx.ext.autodoc.importer._MockObject
                                                                object          at
                                                                0x7f760dde1510>)
```

Amend swagger API.

```
aries_cloudagent.messaging.schemas.routes.register(app:
                                                    <sphinx.ext.autodoc.importer._MockObject
                                                    object at 0x7f760dde1510>)
```

Register routes.

```
aries_cloudagent.messaging.schemas.routes.schemas_created(request:
                                                            <sphinx.ext.autodoc.importer._MockObject
                                                            object          at
                                                            0x7f760dde1510>)
```

Request handler for retrieving schemas that current agent created.

Parameters **request** – aiohttp request object

Returns The identifiers of matching schemas

`aries_cloudagent.messaging.schemas.routes.schemas_get_schema` (*request:*

*<sphinx.ext.autodoc.importer._MockObject
object at
0x7f760dde1510>*)

Request handler for sending a credential offer.

Parameters **request** – aiohttp request object

Returns The schema details.

`aries_cloudagent.messaging.schemas.routes.schemas_send_schema` (*request:*

*<sphinx.ext.autodoc.importer._MockObject
object at
0x7f760dde1510>*)

Request handler for sending a credential offer.

Parameters **request** – aiohttp request object

Returns The schema id sent

aries_cloudagent.messaging.schemas.util module

Schema utilities.

class `aries_cloudagent.messaging.schemas.util.SchemaQueryStringSchema` (**args,*
***kwargs*)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Query string parameters for schema searches.

schema_id = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar`

schema_issuer_did = `<fields.String(default=<marshmallow.missing>, attribute=None, vali`

schema_name = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=N`

schema_version = `<fields.String(default=<marshmallow.missing>, attribute=None, validat`

Submodules

aries_cloudagent.messaging.agent_message module

Agent message base class and schema.

class `aries_cloudagent.messaging.agent_message.AgentMessage` (*_id: str = None,*
_decorators:
aries_cloudagent.messaging.decorators.base
= None)

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Agent message base class.

Handler

Accessor for the agent message's handler class.

Returns Handler class

class Meta

Bases: `object`

AgentMessage metadata.

handler_class = None

message_type = None

schema_class = None

add_trace_decorator (*target: str = 'log', full_thread: bool = True*)

Create a new trace decorator.

Parameters

- **target** – The trace target
- **full_thread** – Full thread flag

add_trace_report (*val: Union[aries_cloudagent.messaging.decorators.trace_decorator.TraceReport, dict]*)

Append a new trace report.

Parameters **val** – The trace target

assign_thread_from (*msg: aries_cloudagent.messaging.agent_message.AgentMessage*)

Copy thread information from a previous message.

Parameters **msg** – The received message containing optional thread information

assign_thread_id (*thid: str, pthid: str = None*)

Assign a specific thread ID.

Parameters

- **thid** – The thread identifier
- **pthid** – The parent thread identifier

assign_trace_decorator (*context, trace*)

Copy trace from a json structure.

Parameters **trace** – string containing trace json stucture

assign_trace_from (*msg: aries_cloudagent.messaging.agent_message.AgentMessage*)

Copy trace information from a previous message.

Parameters **msg** – The received message containing optional trace information

get_signature (*field_name: str*) → `aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator`

Get the signature for a named field.

Parameters **field_name** – Field name to get the signature for

Returns A SignatureDecorator for the requested field name

set_signature (*field_name: str, signature: aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator*)

Add or replace the signature for a named field.

Parameters

- **field_name** – Field to set signature on
- **signature** – Signature for the field

sign_field (*field_name: str, signer_verkey: str, wallet: aries_cloudagent.wallet.base.BaseWallet, timestamp=None*) → *aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator*
Create and store a signature for a named field.

Parameters

- **field_name** – Field to sign
- **signer_verkey** – Verkey of signer
- **wallet** – Wallet to use for signature
- **timestamp** – Optional timestamp for signature

Returns A SignatureDecorator for newly created signature

Raises `ValueError` – If field_name doesn't exist on this message

verify_signatures (*wallet: aries_cloudagent.wallet.base.BaseWallet*) → *bool*
Verify all associated field signatures.

Parameters **wallet** – Wallet to use in verification

Returns True if all signatures verify, else false

verify_signed_field (*field_name: str, wallet: aries_cloudagent.wallet.base.BaseWallet, signer_verkey: str = None*) → *str*
Verify a specific field signature.

Parameters

- **field_name** – The field name to verify
- **wallet** – Wallet to use for the verification
- **signer_verkey** – Verkey of signer to use

Returns The verkey of the signer

Raises

- `ValueError` – If field_name does not exist on this message
- `ValueError` – If the verification fails
- `ValueError` – If the verkey of the signature does not match the provided verkey

exception *aries_cloudagent.messaging.agent_message.AgentMessageError* (**args, error_code: str = None, **kwargs*)

Bases: *aries_cloudagent.messaging.models.base.BaseModelError*

Base exception for agent message issues.

class *aries_cloudagent.messaging.agent_message.AgentMessageSchema* (**args, **kwargs*)

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

AgentMessage schema.

class **Meta**

Bases: `object`

AgentMessageSchema metadata.

model_class = None

signed_fields = None

unknown
Used by autodoc_mock_imports.

check_dump_decorators (*obj*, ***kwargs*)
Pre-dump hook to validate and load the message decorators.

Parameters *obj* – The AgentMessage object

Raises *BaseModelError* – If a decorator does not validate

dump_decorators (*data*, ***kwargs*)
Post-dump hook to write the decorators to the serialized output.

Parameters *obj* – The serialized data

Returns The modified data

extract_decorators (*data: Mapping[KT, VT_co]*, ***kwargs*)
Pre-load hook to extract the decorators and check the signed fields.

Parameters *data* – Incoming data to parse

Returns Parsed and modified data

Raises

- *ValidationError* – If a field signature does not correlate
- to a field in the message
- *ValidationError* – If the message defines both a field signature
- and a value for the same field
- *ValidationError* – If there is a missing field signature

populate_decorators (*obj*, ***kwargs*)
Post-load hook to populate decorators on the message.

Parameters *obj* – The AgentMessage object

Returns The AgentMessage object with populated decorators

replace_signatures (*data*, ***kwargs*)
Post-dump hook to write the signatures to the serialized output.

Parameters *obj* – The serialized data

Returns The modified data

aries_cloudagent.messaging.base_handler module

A Base handler class for all message handlers.

class `aries_cloudagent.messaging.base_handler.BaseHandler`
Bases: `abc.ABC`

Abstract base class for handlers.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Abstract method for handler logic.

Parameters

- **context** – Request context object
- **responder** – A responder object

exception *aries_cloudagent.messaging.base_handler.HandlerException* (**args*, *error_code:* *str* = *None*, ***kwargs*)

Bases: *aries_cloudagent.core.error.BaseError*

Exception base class for generic handler errors.

aries_cloudagent.messaging.error module

Messaging-related error classes and codes.

exception *aries_cloudagent.messaging.error.MessageParseError* (**args*, *error_code:* *str* = *None*, ***kwargs*)

Bases: *aries_cloudagent.core.error.BaseError*

Message parse error.

error_code = 'message_parse_error'

exception *aries_cloudagent.messaging.error.MessagePrepareError* (**args*, *error_code:* *str* = *None*, ***kwargs*)

Bases: *aries_cloudagent.core.error.BaseError*

Message preparation error.

error_code = 'message_prepare_error'

aries_cloudagent.messaging.request_context module

Request context class.

A request context provides everything required by handlers and other parts of the system to process a message.

class *aries_cloudagent.messaging.request_context.RequestContext* (*profile:* *aries_cloudagent.core.profile.Profile*, ***, *context:* *aries_cloudagent.config.injection_container.InjectionContainer* = *None*, *settings:* *Mapping[str, object]* = *None*)

Bases: *object*

Context established by the Conductor and passed into message handlers.

connection_ready

Accessor for the flag indicating an active connection with the sender.

Returns True if the connection is active, else False

connection_record

Accessor for the related connection record.

default_endpoint

Accessor for the default agent endpoint (from agent config).

Returns The default agent endpoint

default_label

Accessor for the default agent label (from agent config).

Returns The default label

inject (*base_cls: Type[Inject], settings: Mapping[str, object] = None, *, required: bool = True*) →

Optional[Inject]

Get the provided instance of a given class identifier.

Parameters

- **cls** – The base class to retrieve an instance of
- **settings** – An optional mapping providing configuration to the provider

Returns An instance of the base class, or None

injector

Accessor for the associated *Injector* instance.

message

Accessor for the deserialized message instance.

Returns This context's agent message

message_receipt

Accessor for the message receipt information.

Returns This context's message receipt information

profile

Accessor for the associated *Profile* instance.

session () → aries_cloudagent.core.profile.ProfileSession

Start a new interactive session with no transaction support requested.

settings

Accessor for the context settings.

classmethod test_context () → aries_cloudagent.messaging.request_context.RequestContext

Quickly set up a new request context for tests.

transaction () → aries_cloudagent.core.profile.ProfileSession

Start a new interactive session with commit and rollback support.

If the current backend does not support transactions, then commit and rollback operations of the session will not have any effect.

update_settings (*settings: Mapping[str, object]*)

Update the scope with additional settings.

aries_cloudagent.messaging.responder module

A message responder.

The responder is provided to message handlers to enable them to send a new message in response to the message being handled.

```
class aries_cloudagent.messaging.responder.BaseResponder(* , connection_id: str =  
                                                         None, reply_session_id:  
                                                         str = None, re-  
                                                         ply_to_verkey: str =  
                                                         None)
```

Bases: `abc.ABC`

Interface for message handlers to send responses.

```
create_outbound (message: Union[aries_cloudagent.messaging.agent_message.AgentMessage,  
                                str, bytes], *, connection_id: str = None, reply_session_id: str = None,  
                                reply_thread_id: str = None, reply_to_verkey: str = None, target:  
                                aries_cloudagent.connections.models.connection_target.ConnectionTarget =  
                                None, target_list: Sequence[aries_cloudagent.connections.models.connection_target.ConnectionTarget]  
                                = None, to_session_only: bool = False) →  
                                aries_cloudagent.transport.outbound.message.OutboundMessage  
Create an OutboundMessage from a message payload.
```

```
send (message: Union[aries_cloudagent.messaging.agent_message.AgentMessage, str, bytes],  
      **kwargs)  
Convert a message to an OutboundMessage and send it.
```

```
send_outbound (message: aries_cloudagent.transport.outbound.message.OutboundMessage)  
Send an outbound message.
```

Parameters **message** – The *OutboundMessage* to be sent

```
send_reply (message: Union[aries_cloudagent.messaging.agent_message.AgentMessage,  
                             str, bytes], *, connection_id: str = None, target:  
                             aries_cloudagent.connections.models.connection_target.ConnectionTarget = None, tar-  
                             get_list: Sequence[aries_cloudagent.connections.models.connection_target.ConnectionTarget]  
                             = None)  
Send a reply to an incoming message.
```

Parameters

- **message** – the *AgentMessage*, or pre-packed str or bytes to reply with
- **connection_id** – optionally override the target connection ID
- **target** – optionally specify a *ConnectionTarget* to send to

Raises *ResponderError* – If there is no active connection

```
send_webhook (topic: str, payload: dict)  
Dispatch a webhook.
```

Parameters

- **topic** – the webhook topic identifier
- **payload** – the webhook payload value

```
class aries_cloudagent.messaging.responder.MockResponder  
Bases: aries_cloudagent.messaging.responder.BaseResponder
```

Mock responder implementation for use by tests.

send (message: Union[aries_cloudagent.messaging.agent_message.AgentMessage, str, bytes], **kwargs)

Convert a message to an OutboundMessage and send it.

send_outbound (message: aries_cloudagent.transport.outbound.message.OutboundMessage)

Send an outbound message.

send_reply (message: Union[aries_cloudagent.messaging.agent_message.AgentMessage, str, bytes], **kwargs)

Send a reply to an incoming message.

send_webhook (topic: str, payload: dict)

Send an outbound message.

exception aries_cloudagent.messaging.responder.ResponderError (*args, error_code: str = None, **kwargs)

Bases: aries_cloudagent.core.error.BaseError

Responder error.

aries_cloudagent.messaging.util module

Utils for messages.

aries_cloudagent.messaging.util.**canon** (raw_attr_name: str) → str

Canonicalize input attribute name for indy proofs and credential offers.

Parameters raw_attr_name – raw attribute name

Returns canonicalized attribute name

aries_cloudagent.messaging.util.**datetime_now** () → datetime.datetime

Timestamp in UTC.

aries_cloudagent.messaging.util.**datetime_to_str** (dt: Union[str, datetime.datetime]) → str

Convert a datetime object to an indy-standard datetime string.

Parameters dt – May be a string or datetime to allow automatic conversion

aries_cloudagent.messaging.util.**encode** (orig: Any) → str

Encode a credential value as an int.

Encode credential attribute value, purely stringifying any int32 and leaving numeric int32 strings alone, but mapping any other input to a stringified 256-bit (but not 32-bit) integer. Predicates in indy-sdk operate on int32 values properly only when their encoded values match their raw values.

Parameters orig – original value to encode

Returns encoded value

aries_cloudagent.messaging.util.**epoch_to_str** (epoch: int) → str

Convert epoch seconds to indy-standard datetime string.

Parameters epoch – epoch seconds

aries_cloudagent.messaging.util.**str_to_datetime** (dt: Union[str, datetime.datetime]) → datetime.datetime

Convert an indy-standard datetime string to a datetime.

Using a fairly lax regex pattern to match slightly different formats. In Python 3.7 datetime.fromisoformat might be used.

Parameters `dt` – May be a string or datetime to allow automatic conversion

`aries_cloudagent.messaging.util.str_to_epoch(dt: Union[str, datetime.datetime]) → int`
 Convert an indy-standard datetime string to epoch seconds.

Parameters `dt` – May be a string or datetime to allow automatic conversion

`aries_cloudagent.messaging.util.time_now() → str`
 Timestamp in ISO format.

aries_cloudagent.messaging.valid module

Validators for schema fields.

```
class aries_cloudagent.messaging.valid.Base58SHA256Hash
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against base58 encoding of SHA-256 hash.
    EXAMPLE = 'H3C2AVvLMv6gmMnam3uVAjZpfkcJCwDwnZn6z3wXmqPV'
    PATTERN = '^ [<sphinx.ext.autodoc.importer._MockObject object>] {43,44}$'

class aries_cloudagent.messaging.valid.Base64
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate base64 value.
    EXAMPLE = 'ey4uLn0='
    PATTERN = '^ [a-zA-Z0-9+/]*={0,2}$'

class aries_cloudagent.messaging.valid.Base64URL
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate base64 value.
    EXAMPLE = 'ey4uLn0='
    PATTERN = '^ [-_a-zA-Z0-9]*={0,2}$'

class aries_cloudagent.messaging.valid.Base64URLNoPad
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate base64 value.
    EXAMPLE = 'ey4uLn0'
    PATTERN = '^ [-_a-zA-Z0-9]*$'

class aries_cloudagent.messaging.valid.DIDKey
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against DID key specification.
    EXAMPLE = 'did:key:z6MkpTHR8VNsBxYAAWHut2Geadd9jSwuBV8xRoAnwWsdvktH'
    PATTERN = '^ did:key:z [<sphinx.ext.autodoc.importer._MockObject object>]+ $'

class aries_cloudagent.messaging.valid.DIDPosture
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against defined DID postures.
    EXAMPLE = 'wallet_only'
```

```

class aries_cloudagent.messaging.valid.Endpoint
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against endpoint URL on any scheme.

    EXAMPLE = 'https://myhost:8021'

    PATTERN = '^([A-Za-z0-9\\.|\\-|\\+|:|/|([A-Za-z0-9|\\.A-Za-z0-9-|+|A-Za-z0-9])+(:[1-9][0-9]

class aries_cloudagent.messaging.valid.EndpointType
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against allowed endpoint/service types.

    EXAMPLE = 'Endpoint'

class aries_cloudagent.messaging.valid.IndyCredDefId
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against indy credential definition identifier specification.

    EXAMPLE = 'WgWxqztrNooG92RXvxSTWv:3:CL:20:tag'

    PATTERN = '^(<sphinx.ext.autodoc.importer._MockObject object>]{21,22}):3:CL:(([1-9][0-9]

class aries_cloudagent.messaging.valid.IndyCredRevId
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against indy credential revocation identifier specification.

    EXAMPLE = '12345'

    PATTERN = '^([1-9][0-9]*$'

class aries_cloudagent.messaging.valid.IndyDID
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against indy DID.

    EXAMPLE = 'WgWxqztrNooG92RXvxSTWv'

    PATTERN = '^((did:sov:)?<sphinx.ext.autodoc.importer._MockObject object>]{21,22}$'

class aries_cloudagent.messaging.valid.IndyExtraWQL
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value as potential extra WQL query in cred search for proof req.

    EXAMPLE = '{"0_drink_uuid": {"attr::drink::value": "martini"}}'

    PATTERN = '^{\s*".*?"\s*:\s*{\s*.*?\s*[, \s*".*?"\s*:\s*{\s*.*?\s*}\s*}\s*$'

class aries_cloudagent.messaging.valid.IndyISO8601DateTime
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against ISO 8601 datetime format, indy profile.

    EXAMPLE = '2021-02-25 23:41:27Z'

    PATTERN = '^\\d{4}-\\d{2}-\\d{2}[T ]\\d{2}:\\d{2}:\\d{2}(?:\\.\\d{1,6})?(?:[+-]?

class aries_cloudagent.messaging.valid.IndyPredicate
    Bases: sphinx.ext.autodoc.importer._MockObject

    Validate value against indy predicate.

    EXAMPLE = '>='

```

```
class aries_cloudagent.messaging.valid.IndyRawPublicKey
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against indy (Ed25519VerificationKey2018) raw public key.
    EXAMPLE = 'H3C2AVvLMv6gmMnam3uVAjZpfkcJCwDwnZn6z3wXmqPV'
    PATTERN = '^(<sphinx.ext.autodoc.importer._MockObject object>){43,44}$'

class aries_cloudagent.messaging.valid.IndyRevRegId
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against indy revocation registry identifier specification.
    EXAMPLE = 'WgWxqztrNooG92RXvxSTWv:4:WgWxqztrNooG92RXvxSTWv:3:CL:20:tag:CL_ACCUM:0'
    PATTERN = '^([<sphinx.ext.autodoc.importer._MockObject object>]{21,22}):4:([<sphinx.ex

class aries_cloudagent.messaging.valid.IndyRevRegSize
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value as indy revocation registry size.
    EXAMPLE = 1000

class aries_cloudagent.messaging.valid.IndySchemaId
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against indy schema identifier specification.
    EXAMPLE = 'WgWxqztrNooG92RXvxSTWv:2:schema_name:1.0'
    PATTERN = '^(<sphinx.ext.autodoc.importer._MockObject object>){21,22}:2:.+: [0-9.]+$'

class aries_cloudagent.messaging.valid.IndyVersion
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against indy version specification.
    EXAMPLE = '1.0'
    PATTERN = '^ [0-9.]+$'

class aries_cloudagent.messaging.valid.IndyWQL
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value as potential WQL query.
    EXAMPLE = '{"attr::name::value": "Alex"}'
    PATTERN = '^ {.*}$'

class aries_cloudagent.messaging.valid.IntEpoch
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against (integer) epoch format.
    EXAMPLE = 1614296487

class aries_cloudagent.messaging.valid.JSONWebToken
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate JSON Web Token.
    EXAMPLE = 'eyJhbGciOiJFZERTQSJ9.eyJhIjogIjAifQ.dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1lgFWFO
    PATTERN = '^[_a-zA-Z0-9]*\\. [_a-zA-Z0-9]*\\. [_a-zA-Z0-9]*$'
```



```

class aries_cloudagent.messaging.valid.JWSHeaderKid
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against JWS header kid.
    EXAMPLE = 'did:sov:LjgpST2rjsoxYegQDRm7EL#keys-4'
    PATTERN = '^did:(?:key:z[<sphinx.ext.autodoc.importer._MockObject object>]+|sov:[<sphinx.ext.autodoc.importer._MockObject object>]+'

class aries_cloudagent.messaging.valid.NaturalNumber
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value as positive integer.
    EXAMPLE = 10

class aries_cloudagent.messaging.valid.NumericStrNatural
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against natural number numeric string.
    EXAMPLE = '1'
    PATTERN = '^([1-9][0-9]*)$'

class aries_cloudagent.messaging.valid.NumericStrWhole
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value against whole number numeric string.
    EXAMPLE = '0'
    PATTERN = '^([0-9]*)$'

class aries_cloudagent.messaging.valid.SHA256Hash
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate (binhex-encoded) SHA256 value.
    EXAMPLE = '617a48c7c8afe0521efdc03e5bb0ad9e655893e6b4b51f0e794d70fba132aacb'
    PATTERN = '^([a-fA-F0-9+/]){64}$'

class aries_cloudagent.messaging.valid.UUIDFour
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate UUID4: 8-4-4-4-12 hex digits, the 13th of which being 4.
    EXAMPLE = '3fa85f64-5717-4562-b3fc-2c963f66afa6'
    PATTERN = '([a-fA-F0-9]{8}-[a-fA-F0-9]{4}-4[a-fA-F0-9]{3}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12})$'

class aries_cloudagent.messaging.valid.WholeNumber
    Bases: sphinx.ext.autodoc.importer._MockObject
    Validate value as non-negative integer.
    EXAMPLE = 0

```

aries_cloudagent.multitenant package

Subpackages

aries_cloudagent.multitenant.admin package

Submodules

aries_cloudagent.multitenant.admin.routes module

Multitenant admin routes.

```
class aries_cloudagent.multitenant.admin.routes.CreateWalletRequestSchema (*args,  
                                                                           **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for adding a new wallet which will be registered by the agent.

image_url = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

key_management_mode = <fields.String(default='managed', attribute=None, validate=<OneOf

label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

validate_fields (data, **kwargs)

Validate schema fields.

Parameters data – The data to validate

Raises ValidationError – If any of the fields do not validate

wallet_dispatch_type = <fields.String(default='default', attribute=None, validate=<OneOf

wallet_key = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

wallet_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

wallet_type = <fields.String(default='in_memory', attribute=None, validate=<OneOf(choi

wallet_webhook_urls = <fields.List(default=<marshmallow.missing>, attribute=None, vali

```
class aries_cloudagent.multitenant.admin.routes.CreateWalletResponseSchema (*args,  
                                                                           **kwargs)
```

Bases: *aries_cloudagent.wallet.models.wallet_record.WalletRecordSchema*

Response schema for creating a wallet.

token = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```
class aries_cloudagent.multitenant.admin.routes.CreateWalletTokenRequestSchema (*args,  
                                                                                **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for creating a wallet token.

wallet_key = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```
class aries_cloudagent.multitenant.admin.routes.CreateWalletTokenResponseSchema (*args,  
                                                                                **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Response schema for creating a wallet token.

token = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```
class aries_cloudagent.multitenant.admin.routes.MultitenantModuleResponseSchema (*args,  
                                                                                **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Response schema for multitenant module.

```
class aries_cloudagent.multitenant.admin.routes.RemoveWalletRequestSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for removing a wallet.

```
wallet_key = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.multitenant.admin.routes.UpdateWalletRequestSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for updating a existing wallet.

```
image_url = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
wallet_dispatch_type = <fields.String(default='default', attribute=None, validate=<One
```

```
wallet_webhook_urls = <fields.List(default=<marshmallow.missing>, attribute=None, vali
```

```
class aries_cloudagent.multitenant.admin.routes.WalletIdMatchInfoSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Path parameters and validators for request taking wallet id.

```
wallet_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.multitenant.admin.routes.WalletListQueryStringSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for wallet list request query string.

```
wallet_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.multitenant.admin.routes.WalletListSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for wallet list.

```
results = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
aries_cloudagent.multitenant.admin.routes.format_wallet_record(wallet_record:
                                                                    aries_cloudagent.wallet.models.wallet_re
```

Serialize a WalletRecord object.

```
aries_cloudagent.multitenant.admin.routes.post_process_routes(app:
                                                                    <sphinx.ext.autodoc.importer._MockObject
                                                                    object          at
                                                                    0x7f760de377d0>)
```

Amend swagger API.

```
aries_cloudagent.multitenant.admin.routes.register(app:
                                                                    <sphinx.ext.autodoc.importer._MockObject
                                                                    object at 0x7f760de377d0>)
```

Register routes.

```
aries_cloudagent.multitenant.admin.routes.wallet_create(request:
                                                                    <sphinx.ext.autodoc.importer._MockObject
                                                                    object          at
                                                                    0x7f760de377d0>)
```

Request handler for adding a new subwallet for handling by the agent.

Parameters **request** – aiohttp request object

```
aries_cloudagent.multitenant.admin.routes.wallet_create_token(request:
                                                                <sphinx.ext.autodoc.importer._MockObject
                                                                object          at
                                                                0x7f760de377d0>)
```

Request handler for creating an authorization token for a specific subwallet.

Parameters **request** – aiohttp request object

```
aries_cloudagent.multitenant.admin.routes.wallet_get(request:
                                                       <sphinx.ext.autodoc.importer._MockObject
                                                       object at 0x7f760de377d0>)
```

Request handler for getting a single subwallet.

Parameters **request** – aiohttp request object

Raises HTTPNotFound – if wallet_id does not match any known wallets

```
aries_cloudagent.multitenant.admin.routes.wallet_remove(request:
                                                         <sphinx.ext.autodoc.importer._MockObject
                                                         object          at
                                                         0x7f760de377d0>)
```

Request handler to remove a subwallet from agent and storage.

Parameters **request** – aiohttp request object.

```
aries_cloudagent.multitenant.admin.routes.wallet_update(request:
                                                         <sphinx.ext.autodoc.importer._MockObject
                                                         object          at
                                                         0x7f760de377d0>)
```

Request handler for updating a existing subwallet for handling by the agent.

Parameters **request** – aiohttp request object

```
aries_cloudagent.multitenant.admin.routes.wallets_list(request:
                                                         <sphinx.ext.autodoc.importer._MockObject
                                                         object          at
                                                         0x7f760de377d0>)
```

Request handler for listing all internal subwallets.

Parameters **request** – aiohttp request object

Submodules

aries_cloudagent.multitenant.error module

Multitenant error classes.

```
exception aries_cloudagent.multitenant.error.WalletKeyMissingError(*args, error_code:
                                                                    str          =
                                                                    None,
                                                                    **kwargs)
```

Bases: `aries_cloudagent.core.error.BaseError`

Wallet key missing exception.

aries_cloudagent.multitenant.manager module

Manager for multitenancy.

class aries_cloudagent.multitenant.manager.**MultitenantManager** (profile: *aries_cloudagent.core.profile.Profile*)

Bases: `object`

Class for handling multitenancy.

add_key (wallet_id: str, recipient_key: str, *, skip_if_exists: bool = False)

Add a wallet key to map incoming messages to specific subwallets.

Parameters

- **wallet_id** – The wallet id the key corresponds to
- **recipient_key** – The recipient key belonging to the wallet
- **skip_if_exists** – Whether to skip the action if the key is already registered for relaying / mediation

create_auth_token (wallet_record: *aries_cloudagent.wallet.models.wallet_record.WalletRecord*, wallet_key: str = None) → str

Create JWT auth token for specified wallet record.

Parameters

- **wallet_record** – The wallet record to create the token for
- **wallet_key** – The wallet key to include in the token. Only required for “unmanaged” wallets

Raises `WalletKeyMissingError` – If the wallet key is missing. Only thrown for “unmanaged” wallets

Returns JWT auth token

Return type str

create_wallet (settings: dict, key_management_mode: str) → *aries_cloudagent.wallet.models.wallet_record.WalletRecord*

Create new wallet and wallet record.

Parameters

- **settings** – The context settings for this wallet
- **key_management_mode** – The mode to use for key management. Either “unmanaged” to not store the wallet key, or “managed” to store the wallet key

Raises `MultitenantManagerError` – If the wallet name already exists

Returns The newly created wallet record

Return type `WalletRecord`

get_default_mediator () → Optional[aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord]

Retrieve the default mediator used for subwallet routing.

Returns retrieved default mediator or None if not set

Return type Optional[`MediationRecord`]

get_profile_for_token (context: aries_cloudagent.config.injection_context.InjectionContext, token: str) → aries_cloudagent.core.profile.Profile

Get the profile associated with a JWT header token.

Parameters

- **context** – The context to use for profile creation
- **token** – The token

Raises

- `WalletKeyMissingError` – If the `wallet_key` is missing for an unmanaged wallet
- `InvalidTokenError` – If there is an exception while decoding the token

Returns Profile associated with the token

get_wallet_profile (*base_context: aries_cloudagent.config.injection_context.InjectionContext*,
wallet_record: aries_cloudagent.wallet.models.wallet_record.WalletRecord,
*extra_settings: dict = {}, *, provision=False*) →
`aries_cloudagent.core.profile.Profile`
Get profile for a wallet record.

Parameters

- **base_context** – Base context to extend from
- **wallet_record** – Wallet record to get the context for
- **extra_settings** – Any extra context settings

Returns Profile for the wallet record

Return type *Profile*

get_wallets_by_message (*message_body, wire_format: aries_cloudagent.transport.wire_format.BaseWireFormat*
= None) → `List[aries_cloudagent.wallet.models.wallet_record.WalletRecord]`
Get the wallet records associated with the message body.

Parameters

- **message_body** – The body of the message
- **wire_format** – Wire format to use for recipient detection

Returns Wallet records associated with the message body

get_webhook_urls (*base_context: aries_cloudagent.config.injection_context.InjectionContext*, *wallet_record: aries_cloudagent.wallet.models.wallet_record.WalletRecord*) → `list`
Get the webhook urls according to `dispatch_type`.

Parameters

- **base_context** – Base context to get `base_webhook_urls`
- **wallet_record** – Wallet record to get `dispatch_type` and `webhook_urls`

Returns webhook urls according to `dispatch_type`

remove_wallet (*wallet_id: str, wallet_key: str = None*)
Remove the wallet with specified wallet id.

Parameters

- **wallet_id** – The wallet id of the wallet record
- **wallet_key** – The wallet key to open the wallet. Only required for “unmanaged” wallets

Raises `WalletKeyMissingError` – If the wallet key is missing. Only thrown for “unmanaged” wallets

update_wallet (*wallet_id: str, new_settings: dict*) → aries_cloudagent.wallet.models.wallet_record.WalletRecord
Update a existing wallet and wallet record.

Parameters

- **wallet_id** – The wallet id of the wallet record
- **new_settings** – The context settings to be updated for this wallet

Returns The updated wallet record

Return type *WalletRecord*

exception aries_cloudagent.multitenant.manager.MultitenantManagerError (*args,
er-
ror_code:
str
=
None,
**kwargs)

Bases: *aries_cloudagent.core.error.BaseError*

Generic multitenant error.

aries_cloudagent.protocols package

Subpackages

aries_cloudagent.protocols.actionmenu package

Subpackages

aries_cloudagent.protocols.actionmenu.v1_0 package

Subpackages

aries_cloudagent.protocols.actionmenu.v1_0.handlers package

Submodules

aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_handler module

Action menu message handler.

class aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_handler.MenuHandler
Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Message handler class for action menus.

handle (*context: aries_cloudagent.messaging.request_context.RequestContext, responder: aries_cloudagent.messaging.responder.BaseResponder*)
Message handler logic for action menus.

Parameters

- **context** – request context

- **responder** – responder callback

`aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_request_handler` module

Action menu request message handler.

class `aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_request_handler.MenuRequestHandler`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for action menu requests.

handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
Message handler logic for action menu requests.

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.actionmenu.v1_0.handlers.perform_handler` module

Action menu perform request message handler.

class `aries_cloudagent.protocols.actionmenu.v1_0.handlers.perform_handler.PerformHandler`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for action menu perform requests.

handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
Message handler logic for action menu perform requests.

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.actionmenu.v1_0.messages` package

Submodules

`aries_cloudagent.protocols.actionmenu.v1_0.messages.menu` module

Represents an action menu.


```

class aries_cloudagent.protocols.actionmenu.v1_0.messages.menu.Menu(*, title:
                                                                    str =
                                                                    None,
                                                                    descrip-
                                                                    tion: str
                                                                    = None,
                                                                    er-
                                                                    rormsg:
                                                                    str =
                                                                    None,
                                                                    options:
                                                                    Se-
                                                                    quence[aries_cloudagent.protoco
                                                                    = None,
                                                                    **kwargs)

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing an action menu.

```

class Meta
    Bases: object

    Metadata for an action menu.

    handler_class = 'aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_handler.M
    message_type = 'action-menu/1.0/menu'
    schema_class = 'MenuSchema'

```

```

class aries_cloudagent.protocols.actionmenu.v1_0.messages.menu.MenuSchema(*args,
                                                                              **kwargs)

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Menu schema class.

```

class Meta
    Bases: object

    Menu schema metadata.

    model_class
        alias of Menu

    unknown = 'exclude'

    description = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    errormsg = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    options = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, r
    title = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```

aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request module

Represents a request for an action menu.

```

class aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequest(**kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage

```

Class representing a request for an action menu.

```
class Meta
    Bases: object
    Metadata for action menu request.
    handler_class = 'aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_request_handler'
    message_type = 'action-menu/1.0/menu-request'
    schema_class = 'MenuRequestSchema'

class aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema(*args, **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    MenuRequest schema class.

class Meta
    Bases: object
    MenuRequest schema metadata.
    model_class
        alias of MenuRequest
    unknown = 'exclude'
```

aries_cloudagent.protocols.actionmenu.v1_0.messages.perform module

Represents a request to perform a menu action.

```
class aries_cloudagent.protocols.actionmenu.v1_0.messages.perform.Perform(*args, **kwargs)
    name: str
    = None,
    params: Mapping[str, str]
    = None,
    **kwargs)

Bases: aries_cloudagent.messaging.agent_message.AgentMessage
Class representing a request to perform a menu action.

class Meta
    Bases: object
    Perform metadata.
    handler_class = 'aries_cloudagent.protocols.actionmenu.v1_0.handlers.perform_handler'
    message_type = 'action-menu/1.0/perform'
    schema_class = 'PerformSchema'

class aries_cloudagent.protocols.actionmenu.v1_0.messages.perform.PerformSchema(*args, **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Perform schema class.
```

```

class Meta
    Bases: object

    Perform schema metadata.

    model_class
        alias of Perform

    unknown = 'exclude'

    name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
    params = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, re

```

aries_cloudagent.protocols.actionmenu.v1_0.models package

Submodules

aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form module

Record used to represent the form associated with an action menu option.

```

class aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form.MenuForm(*,
                                                                            ti-
                                                                            tle:
                                                                            str
                                                                            =
                                                                            None,
                                                                            de-
                                                                            scrip-
                                                                            tion:
                                                                            str
                                                                            =
                                                                            None,
                                                                            params:
                                                                            Se-
                                                                            quence[aries_cloudage
                                                                            =
                                                                            None,
                                                                            sub-
                                                                            mit_label:
                                                                            str
                                                                            =
                                                                            None)

```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Instance of a form associated with an action menu item.

```

class Meta
    Bases: object

    Menu form metadata.

    schema_class = 'MenuFormSchema'

```

```

class aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form.MenuFormSchema(*args,
                                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema

```

MenuForm schema.

```
class Meta
    Bases: object
    MenuFormSchema metadata.
    model_class
        alias of MenuForm
    unknown = 'exclude'
    description = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    params = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, re
    submit_label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
    title = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
```

aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form_param module

Record used to represent a parameter in a menu form.

```
class aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form_param.MenuFormParam(*,
    name:
    str
    =
    None,
    ti-
    tle:
    str
    =
    None,
    de-
    fault:
    str
    =
    None,
    de-
    scrip-
    tion:
    str
    =
    None,
    in-
    put_ty
    str
    =
    None,
    re-
    quired
    bool
    =
    None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Instance of a menu form param associated with an action menu option.

```
class Meta
    Bases: object
    Menu form param metadata.
    schema_class = 'MenuFormParamSchema'

class aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form_param.MenuFormParamSchema
    Bases: aries_cloudagent.messaging.models.base.BaseModelSchema
    MenuFormParam schema.
    class Meta
        Bases: object
        MenuFormParamSchema metadata.
        model_class
            alias of MenuFormParam
        unknown = 'exclude'
    default = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    description = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    input_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
    required = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None,
    title = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
```

`aries_cloudagent.protocols.actionmenu.v1_0.models.menu_option` module

Record used to represent individual menu options in an action menu.

```
class aries_cloudagent.protocols.actionmenu.v1_0.models.menu_option.MenuOption(*,
                                                                              name:
                                                                              str
                                                                              =
                                                                              None,
                                                                              ti-
                                                                              tle:
                                                                              str
                                                                              =
                                                                              None,
                                                                              de-
                                                                              scrip-
                                                                              tion:
                                                                              str
                                                                              =
                                                                              None,
                                                                              dis-
                                                                              abled:
                                                                              bool
                                                                              =
                                                                              None,
                                                                              form:
                                                                              aries_cloudagent
                                                                              =
                                                                              None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Instance of a menu option associated with an action menu.

```
class Meta
```

Bases: *object*

Menu option metadata.

```
    schema_class = 'MenuOptionSchema'
```

```
class aries_cloudagent.protocols.actionmenu.v1_0.models.menu_option.MenuOptionSchema(*args,
                                                                                      **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

MenuOption schema.

```
class Meta
```

Bases: *object*

MenuOptionSchema metadata.

```
    model_class
```

alias of *MenuOption*

```
    unknown = 'exclude'
```

```
description = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
disabled = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
form = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, re
```

```
name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
```

```
title = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
```

Submodules

aries_cloudagent.protocols.actionmenu.v1_0.base_service module

Base action menu service classes.

class aries_cloudagent.protocols.actionmenu.v1_0.base_service.**BaseMenuService** (*context:* aries_cloudagent.c

Bases: abc.ABC

Base action menu service interface.

get_active_menu (*connection:* aries_cloudagent.connections.models.conn_record.ConnRecord = None, *thread_id:* str = None) → aries_cloudagent.protocols.actionmenu.v1_0.messages.menu.Menu
Render the current menu.

Parameters

- **connection** – The active connection record
- **thread_id** – The thread identifier from the requesting message.

perform_menu_action (*action_name:* str, *action_params:* dict, *connection:* aries_cloudagent.connections.models.conn_record.ConnRecord = None, *thread_id:* str = None) → aries_cloudagent.messaging.agent_message.AgentMessage
Perform an action defined by the active menu.

Parameters

- **action_name** – The unique name of the action being performed
- **action_params** – A collection of parameters for the action
- **connection** – The active connection record
- **thread_id** – The thread identifier from the requesting message.

classmethod service_handler ()
Quick accessor for conductor to use.

aries_cloudagent.protocols.actionmenu.v1_0.controller module

Protocol controller for the action menu message family.

class aries_cloudagent.protocols.actionmenu.v1_0.controller.**Controller** (*protocol:* str)

Bases: object

Action menu protocol controller.

determine_roles (*context:* aries_cloudagent.config.injection_context.InjectionContext) → Sequence[str]
Determine what action menu roles are defined.

aries_cloudagent.protocols.actionmenu.v1_0.driver_service module

Driver-based action menu service classes.

```
class aries_cloudagent.protocols.actionmenu.v1_0.driver_service.DriverMenuService(context:  
    aries_cloudagent.context.Context)
```

Bases: `aries_cloudagent.protocols.actionmenu.v1_0.base_service.BaseMenuService`

Driver-based action menu service.

```

get_active_menu (connection: aries_cloudagent.connections.models.conn_record.ConnRecord =
    None, thread_id: str = None) → aries_cloudagent.protocols.actionmenu.v1_0.messages.menu.Menu
    Render the current menu.

```

Parameters

- **connection** – The active connection record
- **thread_id** – The thread identifier from the requesting message.

```
perform_menu_action (action_name: str, action_params: dict, connection:
    aries_cloudagent.connections.models.conn_record.ConnRecord
    = None, thread_id: str = None) →
    aries_cloudagent.messaging.agent_message.AgentMessage
```

Perform an action defined by the active menu.

Parameters

- **action_name** – The unique name of the action being performed
- **action_params** – A collection of parameters for the action
- **connection** – The active connection record
- **thread_id** – The thread identifier from the requesting message.

send_webhook (*topic: str, payload: dict*)
Dispatch a webhook through the registered responder.

aries_cloudagent.protocols.actionmenu.v1_0.message_types module

Message type identifiers for Action Menus.

aries_cloudagent.protocols.actionmenu.v1_0.routes module

Action menu admin routes.

[illegible]

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Schema for the modules endpoint.

[illegible]

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Path parameters and validators for request taking connection id.

```
conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

[illegible]

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Matches MenuSchema but without the inherited AgentMessage properties.


```

description = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
errormsg = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
options = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, r
title = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
class aries_cloudagent.protocols.actionmenu.v1_0.routes.PerformRequestSchema(*args,
                                                                    **kwargs)
Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
Request schema for performing a menu action.
name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
params = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, re
class aries_cloudagent.protocols.actionmenu.v1_0.routes.SendMenuSchema(*args,
                                                                    **kwargs)
Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
Request schema for sending a menu to a connection.
menu = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, re
aries_cloudagent.protocols.actionmenu.v1_0.routes.actionmenu_close(request:
                                                                    <sphinx.ext.autodoc.importer._Mock
                                                                    object at
                                                                    0x7f760de2e710>)
Request handler for closing the menu associated with a connection.
Parameters request – aiohttp request object
aries_cloudagent.protocols.actionmenu.v1_0.routes.actionmenu_fetch(request:
                                                                    <sphinx.ext.autodoc.importer._Mock
                                                                    object at
                                                                    0x7f760de2e710>)
Request handler for fetching the previously-received menu for a connection.
Parameters request – aiohttp request object
aries_cloudagent.protocols.actionmenu.v1_0.routes.actionmenu_perform(request:
                                                                    <sphinx.ext.autodoc.importer._Mock
                                                                    object
                                                                    at
                                                                    0x7f760de2e710>)
Request handler for performing a menu action.
Parameters request – aiohttp request object
aries_cloudagent.protocols.actionmenu.v1_0.routes.actionmenu_request(request:
                                                                    <sphinx.ext.autodoc.importer._Mock
                                                                    object
                                                                    at
                                                                    0x7f760de2e710>)
Request handler for requesting a menu from the connection target.
Parameters request – aiohttp request object
aries_cloudagent.protocols.actionmenu.v1_0.routes.actionmenu_send(request:
                                                                    <sphinx.ext.autodoc.importer._Mock
                                                                    object at
                                                                    0x7f760de2e710>)
Request handler for requesting a menu from the connection target.

```

Parameters **request** – aiohttp request object

```
aries_cloudagent.protocols.actionmenu.v1_0.routes.post_process_routes (app:
                                                                    <sphinx.ext.autodoc.importer.
                                                                    ob-
                                                                    ject at
                                                                    0x7f760de2e710>)
```

Amend swagger API.

```
aries_cloudagent.protocols.actionmenu.v1_0.routes.register (app:
                                                                    <sphinx.ext.autodoc.importer._MockObject
                                                                    object at
                                                                    0x7f760de2e710>)
```

Register routes.

aries_cloudagent.protocols.actionmenu.v1_0.util module

Action menu utility methods.

```
aries_cloudagent.protocols.actionmenu.v1_0.util.retrieve_connection_menu (connection_id:
                                                                    str,
                                                                    con-
                                                                    text:
                                                                    aries_cloudagent.admin.r
                                                                    →
                                                                    aries_cloudagent.protocol
```

Retrieve the previously-received action menu.

```
aries_cloudagent.protocols.actionmenu.v1_0.util.save_connection_menu (menu:
                                                                    aries_cloudagent.protocols.acti
                                                                    con-
                                                                    nec-
                                                                    tion_id:
                                                                    str,
                                                                    con-
                                                                    text:
                                                                    aries_cloudagent.admin.request
```

Save a received action menu.

Submodules

aries_cloudagent.protocols.actionmenu.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.basicmessage package

Subpackages

aries_cloudagent.protocols.basicmessage.v1_0 package

Subpackages

aries_cloudagent.protocols.basicmessage.v1_0.handlers package

Submodules

aries_cloudagent.protocols.basicmessage.v1_0.handlers.basicmessage_handler module

Basic message handler.

```
class aries_cloudagent.protocols.basicmessage.v1_0.handlers.basicmessage_handler.BasicMessageHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for basic messages.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Message handler logic for basic messages.
```

Parameters

- **context** – request context
- **responder** – responder callback

aries_cloudagent.protocols.basicmessage.v1_0.messages package

Submodules

aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage module

Basic message.

```
class aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage.BasicMessage (*,
    sent_time: Union[datetime.datetime, None],
    content: str,
    localization: str,
    **kwargs)

    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
```

Class defining the structure of a basic message.

```
class Meta
    Bases: object
    Basic message metadata class.
    handler_class = 'aries_cloudagent.protocols.basicmessage.v1_0.handlers.basicmessage'
    message_type = 'basicmessage/1.0/message'
    schema_class = 'BasicMessageSchema'

class aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage.BasicMessageSchema
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Basic message schema class.
    class Meta
        Bases: object
        Basic message schema metadata.
        model_class
            alias of BasicMessage
        unknown = 'exclude'
    content = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    sent_time = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
```

Submodules

aries_cloudagent.protocols.basicmessage.v1_0.message_types module

Message type identifiers for Connections.

aries_cloudagent.protocols.basicmessage.v1_0.routes module

Basic message admin routes.

```
class aries_cloudagent.protocols.basicmessage.v1_0.routes.BasicMessageModuleResponseSchema
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Response schema for Basic Message Module.
class aries_cloudagent.protocols.basicmessage.v1_0.routes.ConnIdMatchInfoSchema(*args,
                                                                              **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Path parameters and validators for request taking connection id.
    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.protocols.basicmessage.v1_0.routes.SendMessageSchema(*args,
                                                                              **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for sending a message.
    content = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
aries_cloudagent.protocols.basicmessage.v1_0.routes.connections_send_message(request:
    <sphinx.ext.autodoc
    ob-
    ject
    at
    0x7f760dc84e10>)
```

Request handler for sending a basic message to a connection.

Parameters **request** – aiohttp request object

```
aries_cloudagent.protocols.basicmessage.v1_0.routes.post_process_routes(app:
    <sphinx.ext.autodoc.import
    ob-
    ject
    at
    0x7f760dc84e10>)
```

Amend swagger API.

```
aries_cloudagent.protocols.basicmessage.v1_0.routes.register(app:
    <sphinx.ext.autodoc.importer._MockObject
    object          at
    0x7f760dc84e10>)
```

Register routes.

Submodules

aries_cloudagent.protocols.basicmessage.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.connections package

Subpackages

aries_cloudagent.protocols.connections.v1_0 package

Subpackages

aries_cloudagent.protocols.connections.v1_0.handlers package

Submodules

aries_cloudagent.protocols.connections.v1_0.handlers.connection_invitation_handler module

Connect invitation handler.

```
class aries_cloudagent.protocols.connections.v1_0.handlers.connection_invitation_handler.C
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler class for connection invitations.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle connection invitation.

Parameters

- **context** – Request context
- **responder** – Responder callback

aries_cloudagent.protocols.connections.v1_0.handlers.connection_request_handler module

Connection request handler.

class `aries_cloudagent.protocols.connections.v1_0.handlers.connection_request_handler.ConnectionRequestHandler`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Handler class for connection requests.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle connection request.

Parameters

- **context** – Request context
- **responder** – Responder callback

aries_cloudagent.protocols.connections.v1_0.handlers.connection_response_handler module

Connection response handler.

class `aries_cloudagent.protocols.connections.v1_0.handlers.connection_response_handler.ConnectionResponseHandler`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Handler class for connection responses.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle connection response.

Parameters

- **context** – Request context
- **responder** – Responder callback

aries_cloudagent.protocols.connections.v1_0.messages package

Submodules

aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation module

Represents an invitation message for establishing connection.

```
class aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a connection invitation.

```
class Meta
```

Bases: *object*

Metadata for a connection invitation.

```
    handler_class = 'aries_cloudagent.protocols.connections.v1_0.handlers.connection_invitation_handler.ConnectionInvitationHandler'
```

```
    message_type = 'connections/1.0/invitation'
```

```
    schema_class = 'ConnectionInvitationSchema'
```

```
classmethod from_url(url: str) → aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation
```

Parse a URL-encoded invitation into a *ConnectionInvitation* message.

Parameters *url* – Url to decode

Returns A *ConnectionInvitation* object.

```
to_url(base_url: str = None) → str
```

Convert an invitation to URL format for sharing.

Returns An invite url

class aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.**ConnectionInvitation**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Connection invitation schema class.

class Meta

Bases: *object*

Connection invitation schema metadata.

model_class

alias of *ConnectionInvitation*

unknown = 'exclude'

did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation.validate_fields>)>

endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

image_url = <fields.Url(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>

label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>

recipient_keys = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>

routing_keys = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>

validate_fields (data, **kwargs)

Validate schema fields.

Parameters **data** – The data to validate

Raises *ValidationError* – If any of the fields do not validate

aries_cloudagent.protocols.connections.v1_0.messages.connection_request module

Represents a connection request message.

class aries_cloudagent.protocols.connections.v1_0.messages.connection_request.**ConnectionRequest**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a connection request.


```

class Meta
    Bases: object
    Metadata for a connection request.
    handler_class = 'aries_cloudagent.protocols.connections.v1_0.handlers.connection_re
    message_type = 'connections/1.0/request'
    schema_class = 'ConnectionRequestSchema'

```

```

class aries_cloudagent.protocols.connections.v1_0.messages.connection_request.ConnectionRe

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Connection request schema class.

```

class Meta
    Bases: object
    Connection request schema metadata.
    model_class
        alias of ConnectionRequest
    unknown = 'exclude'
    connection = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,
    image_url = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```

aries_cloudagent.protocols.connections.v1_0.messages.connection_response module

Represents a connection response message.

```

class aries_cloudagent.protocols.connections.v1_0.messages.connection_response.ConnectionRe

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a connection response.

```

class Meta
    Bases: object
    Metadata for a connection response.
    handler_class = 'aries_cloudagent.protocols.connections.v1_0.handlers.connection_re
    message_type = 'connections/1.0/response'
    schema_class = 'ConnectionResponseSchema'

```

```

class aries_cloudagent.protocols.connections.v1_0.messages.connection_response.ConnectionRe

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Connection response schema class.

```
class Meta
    Bases: object
    Connection response schema metadata.
    model_class
        alias of ConnectionResponse
    signed_fields = ('connection',)
    unknown = 'exclude'
    connection = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

aries_cloudagent.protocols.connections.v1_0.messages.problem_report module

Represents a connection problem report message.

```
class aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReport(*,
    problem_report_id: str,
    problem_report_text: str,
    explanation: str,
    **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
```

Base class representing a connection problem report message.

```
class Meta
    Bases: object
    Connection problem report metadata.
    handler_class = 'aries_cloudagent.messaging.problem_report.handler.ProblemReportHandler'
    message_type = 'connections/1.0/problem_report'
    schema_class = 'ProblemReportSchema'
```

```
class aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportReasonCode
    Bases: str, enum.Enum
```

Supported reason codes.

```
INVITATION_NOT_ACCEPTED = 'invitation_not_accepted'
REQUEST_NOT_ACCEPTED = 'request_not_accepted'
REQUEST_PROCESSING_ERROR = 'request_processing_error'
RESPONSE_NOT_ACCEPTED = 'response_not_accepted'
RESPONSE_PROCESSING_ERROR = 'response_processing_error'
```

```
class aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Schema for ProblemReport base class.

```
class Meta
    Bases: object

    Metadata for problem report schema.

    model_class
        alias of ProblemReport

    unknown = 'exclude'

    explain = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    problem_code = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

aries_cloudagent.protocols.connections.v1_0.models package

Submodules

aries_cloudagent.protocols.connections.v1_0.models.connection_detail module

An object for containing the connection request/response DID information.

```
class aries_cloudagent.protocols.connections.v1_0.models.connection_detail.ConnectionDetail
```

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Class representing the details of a connection.

```
class Meta
    Bases: object

    ConnectionDetail metadata.

    schema_class = 'ConnectionDetailSchema'

did
    Accessor for the connection DID.

    Returns The DID for this connection

did_doc
    Accessor for the connection DID Document.

    Returns The DIDDoc for this connection
```

```
class aries_cloudagent.protocols.connections.v1_0.models.connection_detail.ConnectionDetail
```

Bases: `aries_cloudagent.messaging.models.base.BaseModelSchema`

ConnectionDetail schema.

```
class Meta
    Bases: object
    ConnectionDetailSchema metadata.
    model_class
        alias of ConnectionDetail
    unknown = 'exclude'
    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    did_doc = <fields.DIDDocWrapper(default=<marshmallow.missing>, attribute=None, validate=
```

```
class aries_cloudagent.protocols.connections.v1_0.models.connection_detail.DIDDocWrapper(*,
def
fa
Ar
=
<
me
lo
m.
in
Ar
=
<
me
lo
da
st
=
Ne
at
tr
st
=
Ne
va
i-
da
U
Ar
It-
er
ab
Ar
=
Ne
re
qu
be
=
Fe
al
lo
be
=
Ne
lo
be
=
Fe
du
be
=
Fe
er
ro
D
st
=
Ne
**
```

Bases: `marshmallow.fields.Field`

Field that loads and serializes DIDDoc.

Submodules

`aries_cloudagent.protocols.connections.v1_0.manager` module

Classes to manage connections.

class `aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager` (*session:* `aries_cloudagent.con`

Bases: `aries_cloudagent.connections.base_manager.BaseConnectionManager`

Class for managing connections.

accept_response (*response:* `aries_cloudagent.protocols.connections.v1_0.messages.connection_response.ConnectionResponse`, *receipt:* `aries_cloudagent.transport.inbound.receipt.MessageReceipt`) → `aries_cloudagent.connections.models.conn_record.ConnRecord`

Accept a connection response.

Process a `ConnectionResponse` message by looking up the connection request and setting up the pairwise connection.

Parameters

- **response** – The `ConnectionResponse` to accept
- **receipt** – The message receipt

Returns The updated `ConnRecord` representing the connection

Raises

- `ConnectionManagerError` – If there is no DID associated with the connection response
- `ConnectionManagerError` – If the corresponding connection is not at the request or response stage

create_invitation (*my_label:* `str = None`, *my_endpoint:* `str = None`, *auto_accept:* `bool = None`, *public:* `bool = False`, *multi_use:* `bool = False`, *alias:* `str = None`, *routing_keys:* `Sequence[str] = None`, *recipient_keys:* `Sequence[str] = None`, *metadata:* `dict = None`, *mediation_id:* `str = None`) → `Tuple[aries_cloudagent.connections.models.conn_record.ConnRecord, aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation]`

Generate new connection invitation.

This interaction represents an out-of-band communication channel. In the future and in practice, these sort of invitations will be received over any number of channels such as SMS, Email, QR Code, NFC, etc.

Structure of an invite message:

```
{
  "@type": "https://didcomm.org/connections/1.0/invitation",
  "label": "Alice",
  "did": "did:sov:QmWbsNYhMrjHiqZDTUTEJs"
}
```

Or, in the case of a peer DID:

```
{
  "@type": "https://didcomm.org/connections/1.0/invitation",
  "label": "Alice",
  "did": "did:peer:oiSqsNYhMrjHiqZDTUthsw",
  "recipient_keys": ["8HH5gYEeNc3z7PYXmd54d4x6qAfCNrqQqEB3nS7Zfu7K"],
  "service_endpoint": "https://example.com/endpoint",
  "routing_keys": ["9EH5gYEeNc3z7PYXmd53d5x6qAfCNrqQqEB4nS7Zfu6K"],
}
```

Parameters

- **my_label** – label for this connection
- **my_endpoint** – endpoint where other party can reach me
- **auto_accept** – auto-accept a corresponding connection request (None to use config)
- **public** – set to create an invitation from the public DID
- **multi_use** – set to True to create an invitation for multiple use
- **alias** – optional alias to apply to connection for later use

Returns A tuple of the new *ConnRecord* and *ConnectionInvitation* instances

create_request (*connection*: *aries_cloudagent.connections.models.conn_record.ConnRecord*,
my_label: *str = None*, *my_endpoint*: *str = None*, *mediation_id*: *str = None*) →
aries_cloudagent.protocols.connections.v1_0.messages.connection_request.ConnectionRequest
 Create a new connection request for a previously-received invitation.

Parameters

- **connection** – The *ConnRecord* representing the invitation to accept
- **my_label** – My label
- **my_endpoint** – My endpoint

Returns A new *ConnectionRequest* message to send to the other agent

create_response (*connection*: *aries_cloudagent.connections.models.conn_record.ConnRecord*,
my_endpoint: *str = None*, *mediation_id*: *str = None*) →
aries_cloudagent.protocols.connections.v1_0.messages.connection_response.ConnectionResponse
 Create a connection response for a received connection request.

Parameters

- **connection** – The *ConnRecord* with a pending connection request
- **my_endpoint** – The endpoint I can be reached at
- **mediation_id** – The record id for mediation that contains routing_keys and
- **endpoint** (*service*) –

Returns A tuple of the updated *ConnRecord* new *ConnectionResponse* message

create_static_connection (*my_did*: *str = None*, *my_seed*: *str = None*, *their_did*: *str = None*,
their_seed: *str = None*, *their_verkey*: *str = None*,
their_endpoint: *str = None*, *their_label*: *str = None*, *alias*:
str = None) → *Tuple*[*aries_cloudagent.wallet.base.DIDInfo*,
aries_cloudagent.wallet.base.DIDInfo,
aries_cloudagent.connections.models.conn_record.ConnRecord]
 Register a new static connection (for use by the test suite).

Parameters

- **my_did** – override the DID used in the connection
- **my_seed** – provide a seed used to generate our DID and keys
- **their_did** – provide the DID used by the other party
- **their_seed** – provide a seed used to generate their DID and keys
- **their_verkey** – provide the verkey used by the other party
- **their_endpoint** – their URL endpoint for routing messages
- **alias** – an alias for this connection record

Returns my DIDInfo, their DIDInfo, new *ConnRecord* instance

Return type Tuple

establish_inbound (*connection: aries_cloudagent.connections.models.conn_record.ConnRecord, inbound_connection_id: str, outbound_handler: Coroutine[T_co, T_contra, V_co]*) → str

Assign the inbound routing connection for a connection record.

Returns: the current routing state (request or done)

find_connection (*their_did: str, my_did: str = None, my_verkey: str = None, auto_complete=False*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*
Look up existing connection information for a sender verkey.

Parameters

- **their_did** – Their DID
- **my_did** – My DID
- **my_verkey** – My verkey
- **auto_complete** – Should this connection automatically be promoted to active

Returns The located *ConnRecord*, if any

find_inbound_connection (*receipt: aries_cloudagent.transport.inbound.receipt.MessageReceipt*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Deserialize an incoming message and further populate the request context.

Parameters **receipt** – The message receipt

Returns The *ConnRecord* associated with the expanded message, if any

get_connection_targets (*, *connection_id: str = None, connection: aries_cloudagent.connections.models.conn_record.ConnRecord = None*)

Create a connection target from a *ConnRecord*.

Parameters

- **connection_id** – The connection ID to search for
- **connection** – The connection record itself, if already available

get_endpoints (*conn_id: str*) → Tuple[str, str]

Get connection endpoints.

Parameters **conn_id** – connection identifier

Returns Their endpoint for this connection

receive_invitation (*invitation: aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation*, *their_public_did: str = None*, *auto_accept: bool = None*, *alias: str = None*, *mediation_id: str = None*, *mediation_record: aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord = None*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Create a new connection record to track a received invitation.

Parameters

- **invitation** – The *ConnectionInvitation* to store
- **auto_accept** – set to auto-accept the invitation (None to use config)
- **alias** – optional alias to set on the record

Returns The new *ConnRecord* instance

receive_request (*request: aries_cloudagent.protocols.connections.v1_0.messages.connection_request.ConnectionRequest*, *receipt: aries_cloudagent.transport.inbound.receipt.MessageReceipt*, *mediation_id: str = None*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Receive and store a connection request.

Parameters

- **request** – The *ConnectionRequest* to accept
- **receipt** – The message receipt

Returns The new or updated *ConnRecord* instance

resolve_inbound_connection (*receipt: aries_cloudagent.transport.inbound.receipt.MessageReceipt*) → *aries_cloudagent.connections.models.conn_record.ConnRecord*

Populate the receipt DID information and find the related *ConnRecord*.

Parameters **receipt** – The message receipt

Returns The *ConnRecord* associated with the expanded message, if any

session

Accessor for the current profile session.

Returns The profile session for this connection manager

update_inbound (*inbound_connection_id: str*, *recip_verkey: str*, *routing_state: str*)

Activate connections once a route has been established.

Looks up pending connections associated with the inbound routing connection and marks the routing as complete.

exception *aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManagerError* (*args, error_code: str, **kwargs)

Bases: *aries_cloudagent.core.error.BaseError*

Connection error.

aries_cloudagent.protocols.connections.v1_0.message_types module

Message type identifiers for Connections.

aries_cloudagent.protocols.connections.v1_0.routes module

Connection handling admin routes.

```
class aries_cloudagent.protocols.connections.v1_0.routes.AcceptInvitationQueryStringSchema
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for accept invitation request query string.

```
    mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

```
    my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
```

```
    my_label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.AcceptRequestQueryStringSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for accept conn-request web-request query string.

```
    my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnIdMatchInfoSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Path parameters and validators for request taking connection id.

```
    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnIdRefIdMatchInfoSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Path parameters and validators for request taking connection and ref ids.

```
    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
    ref_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionListSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for connection list.

```
    results = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionMetadataQuerySchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Query schema for metadata.

```
    key = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, req
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionMetadataSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for connection metadata.

```
    results = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionMetadataSetRequestSchema
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request Schema for set metadata.

```
    metadata = <fields.Dict (default=<marshmallow.missing>, attribute=None, validate=None,
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionModuleResponseSchema (*args,
**kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Response schema for connection module.

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionStaticRequestSchema (*args,
**kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for a new static connection.

```
    alias = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
    my_did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries
```

```
    my_seed = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None,
```

```
    their_did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
    their_endpoint = <fields.String (default=<marshmallow.missing>, attribute=None, validat
```

```
    their_label = <fields.String (default=<marshmallow.missing>, attribute=None, validate=N
```

```
    their_seed = <fields.String (default=<marshmallow.missing>, attribute=None, validate=No
```

```
    their_verkey = <fields.String (default=<marshmallow.missing>, attribute=None, validate=
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionStaticResultSchema (*args,
**kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for new static connection.

```
    mv_verkey = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
    my_did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries
```

```
    my_endpoint = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<
```

```
    record = <fields.Nested (default=<marshmallow.missing>, attribute=None, validate=None,
```

```
    their_did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
    their_verkey = <fields.String (default=<marshmallow.missing>, attribute=None, validate=
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ConnectionsListQueryStringSchema (*args,
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for connections list request query string.

```
    alias = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
    invitation_key = <fields.String (default=<marshmallow.missing>, attribute=None, validat
```

```
    my_did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<aries
```

```
    state = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<OneOf (
```

```
    their_did = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
    their_role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<O
```

class aries_cloudagent.protocols.connections.v1_0.routes.CreateInvitationQueryStringSchema

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for create invitation request query string.

```
    alias = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
    auto_accept = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=
    multi_use = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=No
    public = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None,
```

class aries_cloudagent.protocols.connections.v1_0.routes.CreateInvitationRequestSchema (*args, **kw

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for invitation connection target.

```
    mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
    metadata = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None,
    recipient_keys = <fields.List(default=<marshmallow.missing>, attribute=None, validate=
    routing_keys = <fields.List(default=<marshmallow.missing>, attribute=None, validate=No
    service_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, valid
```

class aries_cloudagent.protocols.connections.v1_0.routes.EndpointsResultSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for connection endpoints.

```
    my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
    their_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate
```

class aries_cloudagent.protocols.connections.v1_0.routes.InvitationResultSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Result schema for a new connection invitation.

```
    connection_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
    invitation = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=No
    invitation_url = <fields.String(default=<marshmallow.missing>, attribute=None, validate
```

class aries_cloudagent.protocols.connections.v1_0.routes.ReceiveInvitationQueryStringSchema

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for receive invitation request query string.

```
    alias = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
    auto_accept = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=
    mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

```
class aries_cloudagent.protocols.connections.v1_0.routes.ReceiveInvitationRequestSchema (*args, **kwargs)
```

Bases: `aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitationSchema`

Request schema for receive invitation request.

validate_fields (data, **kwargs)

Bypass middleware field validation: marshmallow has no data yet.

```
aries_cloudagent.protocols.connections.v1_0.routes.connection_sort_key (conn)
```

Get the sorting key for a particular connection.

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_accept_invitation (request:
    <sphinx.ext.autodoc.Object at 0x7f760e55fed0>)
```

Request handler for accepting a stored connection invitation.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_accept_request (request:
    <sphinx.ext.autodoc.Object at 0x7f760e55fed0>)
```

Request handler for accepting a stored connection request.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_create_invitation (request:
    <sphinx.ext.autodoc.Object at 0x7f760e55fed0>)
```

Request handler for creating a new connection invitation.

Parameters **request** – aiohttp request object

Returns The connection invitation details

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_create_static (request:
    <sphinx.ext.autodoc.Object at 0x7f760e55fed0>)
```

Request handler for creating a new static connection.

Parameters **request** – aiohttp request object

Returns The new connection record

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_endpoints(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760e55fed0>)
```

Request handler for fetching connection endpoints.

Parameters **request** – aiohttp request object

Returns The endpoints response

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_establish_inbound(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760e55fed0>)
```

Request handler for setting the inbound connection on a connection record.

Parameters **request** – aiohttp request object

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_list(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760e55fed0>)
```

Request handler for searching connection records.

Parameters **request** – aiohttp request object

Returns The connection list response

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_metadata(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760e55fed0>)
```

Handle fetching metadata associated with a single connection record.

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_metadata_set(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760e55fed0>)
```

Handle fetching metadata associated with a single connection record.

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_receive_invitation(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760e55fed0>)
```

Request handler for receiving a new connection invitation.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_remove(request:
    <sphinx.ext.autodoc.importer.
    ob-
    ject at
    0x7f760e55fed0>)
```

Request handler for removing a connection record.

Parameters **request** – aiohttp request object

```
aries_cloudagent.protocols.connections.v1_0.routes.connections_retrieve(request:
    <sphinx.ext.autodoc.importer.
    ob-
    ject
    at
    0x7f760e55fed0>)
```

Request handler for fetching a single connection record.

Parameters **request** – aiohttp request object

Returns The connection record response

```
aries_cloudagent.protocols.connections.v1_0.routes.post_process_routes(app:
    <sphinx.ext.autodoc.importer.
    ob-
    ject
    at
    0x7f760e55fed0>)
```

Amend swagger API.

```
aries_cloudagent.protocols.connections.v1_0.routes.register(app:
    <sphinx.ext.autodoc.importer._MockObject
    object at
    0x7f760e55fed0>)
```

Register routes.

Submodules

aries_cloudagent.protocols.connections.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.coordinate_mediation package

Subpackages

aries_cloudagent.protocols.coordinate_mediation.v1_0 package

Subpackages

aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers package

Submodules

`aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_handler` module

Handler for keylist message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_handler.KeylistHandler
```

```
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for keylist message.

```
    handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder: aries_cloudagent.messaging.responder.BaseResponder)
        Handle keylist message.
```

`aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_query_handler` module

Handler for keylist-query message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_query_handler.KeylistQueryHandler
```

```
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for keylist-query message.

```
    handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder: aries_cloudagent.messaging.responder.BaseResponder)
        Handle keylist-query message.
```

`aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_update_handler` module

Handler for keylist-update messages.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_update_handler.KeylistUpdateHandler
```

```
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for keylist-update messages.

```
    handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder: aries_cloudagent.messaging.responder.BaseResponder)
        Handle keylist-update messages.
```

`aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_update_response_handler` module

Handler for keylist-update-response message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_update_response_handler.KeylistUpdateResponseHandler
```

```
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for keylist-update-response message.

```
    handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder: aries_cloudagent.messaging.responder.BaseResponder)
        Handle keylist-update-response message.
```


aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediation_deny_handler module

Handler for mediate-deny message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediation_deny_handler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for mediate-deny message.

```
    handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
            aries_cloudagent.messaging.responder.BaseResponder)
        Handle mediate-deny message.
```

aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediation_grant_handler module

Handler for mediate-grant message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediation_grant_handler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for mediate-grant message.

```
    handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
            aries_cloudagent.messaging.responder.BaseResponder)
        Handle mediate-grant message.
```

aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediation_request_handler module

Handler for mediate-request message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediation_request_handler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for mediate-request message.

```
    handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
            aries_cloudagent.messaging.responder.BaseResponder)
        Handle mediate-request message.
```

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages package

Subpackages

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner package

Submodules

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_key module

Inner structure of keylist message. Represents a single item in keylist.keys.

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_key.Key1`

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Inner structure of Keylist keys attribute.

class Meta

Bases: `object`

KeylistKey metadata.

schema_class = `'KeylistKeySchema'`

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_key.Key1`

Bases: `aries_cloudagent.messaging.models.base.BaseModelSchema`

KeylistKey schema.

class Meta

Bases: `object`

KeylistKeySchema metadata.

model_class

alias of `KeylistKey`

unknown

Used by autodoc_mock_imports.

recipient_key

Used by autodoc_mock_imports.

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_query_paginate
module

Inner structure of KeylistQuery. Represents KeylistQuery.paginate.

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_query_pagination

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a keylist query pagination.

class Meta

Bases: *object*

Keylist query pagination metadata.

schema_class = 'KeylistQueryPaginateSchema'

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_query_pagination_schema

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Keylist query pagination schema.

class Meta

Bases: *object*

Keylist query pagination schema metadata.

model_class

alias of *KeylistQueryPaginate*

limit

Used by autodoc_mock_imports.

offset

Used by autodoc_mock_imports.

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_update_rule module

Inner structure of keylist-update message.

Represents single item of keylist-update.updates.

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_update_rule

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a keylist update rule.

class Meta

Bases: *object*

Keylist update metadata.

schema_class = 'KeylistUpdateRuleSchema'

RULE_ADD = 'add'

RULE_REMOVE = 'remove'

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_update_rule

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Keylist update specification schema.

class Meta

Bases: *object*

Keylist update schema metadata.

model_class

alias of *KeylistUpdateRule*

action

Used by autodoc_mock_imports.

recipient_key

Used by autodoc_mock_imports.

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_updated module

Inner structure of keylist-update-response.

Represents single item in keylist-update-response.updated list.

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_updated_item

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a route update response.

class Meta

Bases: *object*

KeylistUpdated metadata.

schema_class = 'KeylistUpdatedSchema'

RESULT_CLIENT_ERROR = 'client_error'

RESULT_NO_CHANGE = 'no_change'

```
RESULT_SERVER_ERROR = 'server_error'
```

```
RESULT_SUCCESS = 'success'
```

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_updated.KeylistUpdated:
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

KeylistUpdated schema.

```
class Meta
```

Bases: *object*

KeylistUpdatedSchema metadata.

```
model_class
```

alias of *KeylistUpdated*

```
unknown
```

Used by autodoc_mock_imports.

```
action
```

Used by autodoc_mock_imports.

```
recipient_key
```

Used by autodoc_mock_imports.

```
result
```

Used by autodoc_mock_imports.

Submodules

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist module

Response to keylist-query message.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist.Keylist(*,
keys: Sequence[str] = None,
page_id: str = None,
**kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a keylist-query response.

```
class Meta
```

Bases: *object*

Metadata for a keylist query response.

```
handler_class = 'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist'
```

```

    message_type = 'coordinate-mediation/1.0/keylist'
    schema_class = 'KeylistSchema'
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist.KeylistSchema (
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Keylist query response schema class.

    class Meta
        Bases: object
        Keylist query response schema metadata.

        model_class
            alias of Keylist

    keys
        Used by autodoc_mock_imports.

    pagination
        Used by autodoc_mock_imports.

```

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_query module

keylist-query message used to request list of keys handled by mediator.

```

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_query.KeylistQuery (
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
    Class representing a keylist query message.

    class Meta
        Bases: object
        Metadata for a keylist query.

        handler_class = 'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_query'
        message_type = 'coordinate-mediation/1.0/keylist-query'
        schema_class = 'KeylistQuerySchema'
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_query.KeylistQuerySchema (
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Keylist query schema class.

```

```
class Meta
    Bases: object
    Keylist query schema metadata.

    model_class
        alias of KeylistQuery

filter
    Used by autodoc_mock_imports.

paginate
    Used by autodoc_mock_imports.
```

`aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update` module

keylist-update message used to notify mediator of keylist changes.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update.KeylistUpdate
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a keylist update message.

```
class Meta
    Bases: object
    Metadata for a keylist update.

    handler_class = 'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keylist_update'
    message_type = 'coordinate-mediation/1.0/keylist-update'
    schema_class = 'KeylistUpdateSchema'
```

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update.KeylistUpdateSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Keylist update schema class.

```
class Meta
    Bases: object
    Keylist update schema metadata.

    model_class
        alias of KeylistUpdate

updates
    Used by autodoc_mock_imports.
```

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response module

Response to keylist-update used to notify mediation client of applied updates.

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a keylist update result message.

class Meta

Bases: *object*

Metadata for a keylist update result.

handler_class = 'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.keyl

message_type = 'coordinate-mediation/1.0/keylist-update-response'

schema_class = 'KeylistUpdateResponseSchema'

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Keylist update result schema class.

class Meta

Bases: *object*

Keylist update result schema metadata.

model_class

alias of *KeylistUpdateResponse*

updated

Used by autodoc_mock_imports.

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_deny module

mediate-deny message used to notify mediation client of a denied mediation request.

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_deny.**MediationDeny**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a mediation deny message.

class Meta

Bases: *object*

Metadata for a mediation deny.

handler_class = 'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediate_deny_handler'

message_type = 'coordinate-mediation/1.0/mediate-deny'

schema_class = 'MediationDenySchema'

class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_deny.**MediationDenySchema**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Mediation grant schema class.

class Meta

Bases: *object*

Mediation deny schema metadata.

model_class

alias of *MediationDeny*

mediator_terms

Used by autodoc_mock_imports.

recipient_terms

Used by autodoc_mock_imports.

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_grant module

mediate-grant message.

Used to notify mediation client of a granted mediation request.

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_grant.MediationGrant`

Bases: `aries_cloudagent.messaging.agent_message.AgentMessage`

Class representing a mediation grant message.

class `Meta`

Bases: `object`

Metadata for a mediation grant.

handler_class = `'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.mediate_grant_handler'`

message_type = `'coordinate-mediation/1.0/mediate-grant'`

schema_class = `'MediationGrantSchema'`

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_grant.MediationGrantSchema`

Bases: `aries_cloudagent.messaging.agent_message.AgentMessageSchema`

Mediation grant schema class.

class `Meta`

Bases: `object`

Mediation grant schema metadata.

model_class

alias of `MediationGrant`

endpoint

Used by `autodoc_mock_imports`.

routing_keys

Used by `autodoc_mock_imports`.

aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_request module

mediate-request message used to request mediation from a mediator.

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_request.Mediat
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Represents a request for mediation.

```
class Meta
```

Bases: *object*

MediationRequest metadata.

```
    handler_class = 'aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.medi
```

```
    message_type = 'coordinate-mediation/1.0/mediate-request'
```

```
    schema_class = 'MediationRequestSchema'
```

```
class aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_request.Mediat
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Mediation request schema class.

```
class Meta
```

Bases: *object*

Mediation request schema metadata.

```
    model_class
```

alias of *MediationRequest*

```
    mediator_terms
```

Used by autodoc_mock_imports.

```
    recipient_terms
```

Used by autodoc_mock_imports.

aries_cloudagent.protocols.coordinate_mediation.v1_0.models package

Submodules

`aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record` module

Store state for Mediation requests.

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord`

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Class representing stored mediation information.

class Meta

Bases: *object*

RouteRecord metadata.

schema_class = 'MediationRecordSchema'

RECORD_ID_NAME = 'mediation_id'

RECORD_TYPE = 'mediation_requests'

ROLE_CLIENT = 'client'

ROLE_SERVER = 'server'

STATE_DENIED = 'denied'

STATE_GRANTED = 'granted'

STATE_REQUEST = 'request'

TAG_NAMES = {'connection_id', 'role', 'state'}

classmethod exists_for_connection_id (*session: aries_cloudagent.core.profile.ProfileSession,*
connection_id: str) → bool

Return whether a mediation record exists for the given connection.

Parameters

- **session** (*ProfileSession*) – session
- **connection_id** (*str*) – connection_id

Returns whether record exists

Return type bool

mediation_id

Get Mediation ID.

record_value

Return values of record as dictionary.

classmethod retrieve_by_connection_id (*session: aries_cloudagent.core.profile.ProfileSession,*
connection_id: str) →

aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord

Retrieve a mediation record by connection ID.

Parameters

- **session** (*ProfileSession*) – session
- **connection_id** (*str*) – connection_id

Returns retrieved record

Return type *MediationRecord*

state

Get Mediation state.

class *aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord*

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

MediationRecordSchema schema.

```

class Meta
    Bases: object

    MediationRecordSchema metadata.

    model_class
        alias of MediationRecord

    unknown
        Used by autodoc_mock_imports.

connection_id
    Used by autodoc_mock_imports.

endpoint
    Used by autodoc_mock_imports.

mediation_id
    Used by autodoc_mock_imports.

mediator_terms
    Used by autodoc_mock_imports.

recipient_terms
    Used by autodoc_mock_imports.

role
    Used by autodoc_mock_imports.

routing_keys
    Used by autodoc_mock_imports.

```

Submodules

`aries_cloudagent.protocols.coordinate_mediation.v1_0.manager` module

Manager for Mediation coordination.

exception `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationAlreadyExists`

```

Bases:
    aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManagerError

```

Raised on mediation record already exists for given connection.

```

class aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager (profile:
    aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager)

```

```

Bases: object

```

Class for handling Mediation.

MediationManager creates or retrieves a routing DID as a means to hand out a consistent routing key to mediation clients.

DEFAULT_MEDIATOR_RECORD_TYPE = 'default_mediator'

ROUTING_DID_RECORD_TYPE = 'routing_did'

SEND_REQ_AFTER_CONNECTION = 'send_mediation_request_on_connection'

SET_TO_DEFAULT_ON_GRANTED = 'set_to_default_on_granted'

add_key (*recipient_key*: str, *message*: Optional[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update.Message] = None) → aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update.KeylistUpdate
Prepare a keylist update add.

Parameters

- **recipient_key** (str) – key to add
- **message** (Optional[KeylistUpdate]) – append update to message

Returns Message to send to mediator to notify of key addition.

Return type *KeylistUpdate*

clear_default_mediator ()

Clear the stored default mediator.

create_keylist_query_response (*keylist*: Sequence[aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord]) → aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist.KeylistQueryResponse
Prepare a keylist message from keylist.

Parameters **keylist** (Sequence[RouteRecord]) – keylist to format into message

Returns message to return to client

Return type *Keylist*

deny_request (*mediation_id*: str, *, *mediator_terms*: Sequence[str] = None, *recipient_terms*: Sequence[str] = None) → Tuple[aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord, aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_deny.MediationDeny]
Deny a mediation request and prepare a deny message.

Parameters

- **mediation_id** – mediation record ID to deny
- **mediator_terms** (Sequence[str]) – updated mediator terms to return to requester. –
- **recipient_terms** (Sequence[str]) – updated recipient terms to return to requester. –

Returns message to return to denied client.

Return type *MediationDeny*

get_default_mediator () → Optional[aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord]
Retrieve default mediator from the store.

Returns retrieved default mediator or None if not set

Return type Optional[*MediationRecord*]

get_default_mediator_id () → Optional[str]

Retrieve default mediator ID from the store.

Returns retrieved default mediator ID or None if not set

Return type `Optional[str]`

get_keylist (*record*: `aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord`)
→ `Sequence[aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord]`

Retrieve keylist for mediation client.

Parameters **record** (`MediationRecord`) – record associated with client keylist

Returns sequence of routes (the keylist)

Return type `Sequence[RouteRecord]`

get_my_keylist (*connection_id*: `Optional[str]` = `None`) → `Sequence[aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord]`

Get my routed keys.

Parameters **connection_id** (`Optional[str]`) – connection id of mediator

Returns list of routes (the keylist)

Return type `Sequence[RouteRecord]`

grant_request (*mediation_id*: `str`) → `Tuple[aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord, aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_grant.MediationGrant]`

Grant a mediation request and prepare grant message.

Parameters **mediation_id** – mediation record ID to grant

Returns updated mediation record and message to return to grantee

Return type (`MediationRecord`, `MediationGrant`)

prepare_keylist_query (*filter_*: `dict` = `None`, *paginate_limit*: `int`
= `-1`, *paginate_offset*: `int` = `0`) →
`aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_query.KeylistQuery`

Prepare keylist query message.

Parameters

- **filter** (`dict`) – filter for keylist query
- **paginate_limit** (`int`) – paginate_limit
- **paginate_offset** (`int`) – paginate_offset

Returns message to send to mediator

Return type `KeylistQuery`

prepare_request (*connection_id*: `str`, *mediator_terms*: `Sequence[str]` =
`None`, *recipient_terms*: `Sequence[str]` = `None`) → `Tuple[aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord, aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_request.MediationRequest]`

Prepare a MediationRequest Message, saving a new mediation record.

Parameters

- **connection_id** (`str`) – ID representing mediator
- **mediator_terms** (`Sequence[str]`) – mediator_terms
- **recipient_terms** (`Sequence[str]`) – recipient_terms

Returns message to send to mediator

Return type `MediationRequest`

receive_request (*connection_id*: str, *request*: aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_request → aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord)
Create a new mediation record to track this request.

Parameters **request** (MediationRequest) – request message

Returns record created during receipt of request.

Return type MediationRecord

remove_key (*recipient_key*: str, *message*: Optional[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update.KeylistUpdate] = None) → aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update.KeylistUpdate
Prepare keylist update remove.

Parameters

- **recipient_key** (str) – key to remove
- **message** (Optional[KeylistUpdate]) – append update to message

Returns Message to send to mediator to notify of key removal.

Return type KeylistUpdate

request_denied (*record*: aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord, *deny*: aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_deny.MediationDeny)
Process mediation denied message.

Parameters **record** (MediationRecord) – record representing the denied mediation request

request_granted (*record*: aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord, *grant*: aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.mediate_grant.MediationGrant)
Process mediation grant message.

Parameters **record** (MediationRecord) – record representing the granted mediation request

set_default_mediator (*record*: aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord)
Set default mediator from record.

set_default_mediator_by_id (*mediation_id*: str)
Set default mediator from ID.

store_update_results (*connection_id*: str, *results*: Sequence[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response.KeylistUpdateResponse])
Store results of keylist update from keylist update response message.

Parameters

- **connection_id** (str) – connection ID of mediator sending results
- **results** (Sequence[KeylistUpdateResponse]) – keylist update results
- **session** – An active profile session

update_keylist (*record*: aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord, *updates*: Sequence[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.inner.keylist_update_rule.KeylistUpdateRule]) → aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response.KeylistUpdateResponse
Update routes defined in keylist update rules.

Parameters

- **record** (MediationRecord) – record associated with client updating keylist
- **updates** (Sequence[KeylistUpdateRule]) – updates to apply

Returns message to return to client

Return type *KeylistUpdateResponse*

exception `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManagerError`

Bases: *aries_cloudagent.core.error.BaseError*

Generic Mediation error.

exception `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationNotGrantedError`

Bases: *aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManagerError*

Raised when mediation state should be granted and is not.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.message_types` module

Message type identifiers for Coordinate Mediation protocol.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes` module

coordinate mediation admin routes.

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.AdminMediationDenySchema`

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for Mediation deny admin request query string.

mediator_terms = `<fields.List (default=<marshmallow.missing>, attribute=None, validate=)`

recipient_terms = `<fields.List (default=<marshmallow.missing>, attribute=None, validate=)`

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.GetKeylistQuerySchema` *(*args, **kwargs)*

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Get keylist query string parameters.

conn_id = `<fields.UUID (default=<marshmallow.missing>, attribute=None, validate=None, r`

role = `<fields.String (default=<marshmallow.missing>, attribute=None, validate=<OneOf(c`

class `aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.KeylistQueryFilterRequest`

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Request schema for keylist query filtering.

filter = `<fields.Dict (default=<marshmallow.missing>, attribute=None, validate=None, re`

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.clear_default_mediator` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

Clear set default mediator.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.delete_mediation_request` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

Delete a mediation request by ID.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.get_default_mediator` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

Get default mediator.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.get_keylist` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

Retrieve keylists by connection or role.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.list_mediation_requests` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

List mediation requests for either client or server role.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.mediation_request_deny` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

Deny a stored mediation request.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.mediation_request_grant` (request: *<sphinx.ext.autodoc.imported object at 0x7f760e180950>*)

Grant a stored mediation request.

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.mediation_sort_key` (mediation: *dict*)

Get the sorting key for a particular serialized mediation record.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.post_process_routes (app:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Amend swagger API.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.register (app:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Register routes.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.request_mediation (request:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Request mediation from connection.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.retrieve_mediation_request (request:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Retrieve a single mediation request.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.send_keylist_query (request:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Send keylist query to mediator.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.send_keylist_update (request:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Send keylist update to mediator.

```
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.set_default_mediator (request:
    <sphinx.ext.autodoc.importer._ImportedObject object at
    0x7f760e180950>)
```

Set default mediator.

Submodules

`aries_cloudagent.protocols.coordinate_mediation.definition` module

Version definitions for this protocol.

`aries_cloudagent.protocols.didexchange` package

Subpackages

`aries_cloudagent.protocols.didexchange.v1_0` package

Subpackages

`aries_cloudagent.protocols.didexchange.v1_0.handlers` package

Submodules

`aries_cloudagent.protocols.didexchange.v1_0.handlers.complete_handler` module

Connection complete handler under RFC 23 (DID exchange).

```
class aries_cloudagent.protocols.didexchange.v1_0.handlers.complete_handler.DIDXCompleteHan
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler class for connection complete message under RFC 23 (DID exchange).

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,    responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Handle connection complete under RFC 23 (DID exchange).
```

Parameters

- **context** – Request context
- **responder** – Responder callback

`aries_cloudagent.protocols.didexchange.v1_0.handlers.invitation_handler` module

Connect invitation handler under RFC 23 (DID exchange).

```
class aries_cloudagent.protocols.didexchange.v1_0.handlers.invitation_handler.InvitationHan
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler class for connection invitation message under RFC 23 (DID exchange).

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,    responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Handle connection invitation under RFC 23 (DID exchange).
```

Parameters

- **context** – Request context
- **responder** – Responder callback

aries_cloudagent.protocols.didexchange.v1_0.handlers.request_handler module

Connection request handler under RFC 23 (DID exchange).

class aries_cloudagent.protocols.didexchange.v1_0.handlers.request_handler.DIDXRequestHandler

Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Handler class for connection request message under RFC 23 (DID exchange).

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle connection request under RFC 23 (DID exchange).

Parameters

- **context** – Request context
- **responder** – Responder callback

aries_cloudagent.protocols.didexchange.v1_0.handlers.response_handler module

DID exchange response handler under RFC 23.

class aries_cloudagent.protocols.didexchange.v1_0.handlers.response_handler.DIDXResponseHandler

Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Handler class for DID exchange response message under RFC 23.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle DID exchange response under RFC 23.

Parameters

- **context** – Request context
- **responder** – Responder callback

aries_cloudagent.protocols.didexchange.v1_0.messages package

Submodules

aries_cloudagent.protocols.didexchange.v1_0.messages.complete module

Represents a DID exchange complete message under RFC 23.

class aries_cloudagent.protocols.didexchange.v1_0.messages.complete.DIDXComplete (**kwargs)

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a DID exchange completion.

class Meta

Bases: *object*

Metadata for DID exchange completion.

handler_class = 'aries_cloudagent.protocols.didexchange.v1_0.handlers.complete_handler'

message_type = 'didexchange/1.0/complete'

schema_class = 'DIDXCompleteSchema'

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.complete.DIDXCompleteSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    DID exchange complete schema class.

    class Meta
        Bases: object
        DID exchange complete schema metadata.

        model_class
            alias of DIDXComplete

        unknown = 'exclude'

    check_thread_deco (obj, **kwargs)
        Thread decorator, and its thid and pthid, are mandatory.
```

aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report module

Represents a connection problem report message.

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReport (*args, **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
    Base class representing a connection problem report message.

    class Meta
        Bases: object
        Connection problem report metadata.

        handler_class = 'aries_cloudagent.messaging.problem_report.handler.ProblemReportHandler'
        message_type = 'didexchange/1.0/problem_report'
        schema_class = 'ProblemReportSchema'

class aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportReasonCode
    Bases: str, enum.Enum
    Supported reason codes.

    COMPLETE_NOT_ACCEPTED = 'complete_not_accepted'
    INVITATION_NOT_ACCEPTED = 'invitation_not_accepted'
    REQUEST_NOT_ACCEPTED = 'request_not_accepted'
    REQUEST_PROCESSING_ERROR = 'request_processing_error'
```



```
RESPONSE_NOT_ACCEPTED = 'response_not_accepted'
```

```
RESPONSE_PROCESSING_ERROR = 'response_processing_error'
```

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Schema for ProblemReport base class.

```
class Meta
```

Bases: *object*

Metadata for problem report schema.

```
model_class
```

alias of *ProblemReport*

```
unknown = 'exclude'
```

```
explain = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
problem_code = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

aries_cloudagent.protocols.didexchange.v1_0.messages.request module

Represents a DID exchange request message under RFC 23.

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequest (*,
                                                                              la-
                                                                              bel:
                                                                              str
                                                                              =
                                                                              None,
                                                                              did:
                                                                              str
                                                                              =
                                                                              None,
                                                                              did_doc_attach:
                                                                              aries_cloudagent
                                                                              =
                                                                              None,
                                                                              **kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a DID exchange request under RFC 23.

```
class Meta
```

Bases: *object*

Metadata for DID exchange request under RFC 23.

```
handler_class = 'aries_cloudagent.protocols.didexchange.v1_0.handlers.request_handl
```

```
message_type = 'didexchange/1.0/request'
```

```
schema_class = 'DIDXRequestSchema'
```

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequestSchema (*args,
                                                                                       **kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Schema class for DID exchange request under RFC 23.

```
class Meta
    Bases: object

    DID exchange request schema class metadata.

    model_class
        alias of DIDXRequest

    unknown = 'exclude'

    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    did_doc_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, validat
    label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
```

aries_cloudagent.protocols.didexchange.v1_0.messages.response module

Represents a DID exchange response message under RFC 23.

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.response.DIDXResponse(*,
                                                                                   did:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   did_doc_attach:
                                                                                   aries_cloudagent
                                                                                   =
                                                                                   None,
                                                                                   **kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a DID exchange response under RFC 23.

```
class Meta
    Bases: object

    Metadata for DID exchange response under RFC 23.

    handler_class = 'aries_cloudagent.protocols.didexchange.v1_0.handlers.response_hand
    message_type = 'didexchange/1.0/response'

    schema_class = 'DIDXResponseSchema'
```

```
class aries_cloudagent.protocols.didexchange.v1_0.messages.response.DIDXResponseSchema(*args,
                                                                                       **kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Schema class for DID exchange response under RFC 23.

```
class Meta
    Bases: object

    DID exchange response schema class metadata.

    model_class
        alias of DIDXResponse

    unknown = 'exclude'

    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    did_doc_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, validat
```

Submodules

aries_cloudagent.protocols.didexchange.v1_0.manager module

Classes to manage connection establishment under RFC 23 (DID exchange).

class aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManager (session: aries_cloudagent.core.profile.IProfile, bases: aries_cloudagent.connections.base_manager.BaseConnectionManager)

Class for managing connections under RFC 23 (DID exchange).

accept_complete (complete: aries_cloudagent.protocols.didexchange.v1_0.messages.complete.DIDXComplete, receipt: aries_cloudagent.transport.inbound.receipt.MessageReceipt) → aries_cloudagent.connections.models.conn_record.ConnRecord

Accept a connection complete message under RFC 23 (DID exchange).

Process a *DIDXComplete* message by looking up the connection record and marking the exchange complete.

Parameters

- **complete** – The *DIDXComplete* to accept
- **receipt** – The message receipt

Returns The updated *ConnRecord* representing the connection

Raises *DIDXManagerError* – If the corresponding connection does not exist or is not in the response-sent state

accept_response (response: aries_cloudagent.protocols.didexchange.v1_0.messages.response.DIDXResponse, receipt: aries_cloudagent.transport.inbound.receipt.MessageReceipt) → aries_cloudagent.connections.models.conn_record.ConnRecord

Accept a connection response under RFC 23 (DID exchange).

Process a *DIDXResponse* message by looking up the connection request and setting up the pairwise connection.

Parameters

- **response** – The *DIDXResponse* to accept
- **receipt** – The message receipt

Returns The updated *ConnRecord* representing the connection

Raises

- *DIDXManagerError* – If there is no DID associated with the connection response
- *DIDXManagerError* – If the corresponding connection is not in the request-sent state

create_request (conn_rec: aries_cloudagent.connections.models.conn_record.ConnRecord, my_label: str = None, my_endpoint: str = None, mediation_id: str = None) → aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequest

Create a new connection request for a previously-received invitation.

Parameters

- **conn_rec** – The *ConnRecord* representing the invitation to accept
- **my_label** – My label for request
- **my_endpoint** – My endpoint

- **mediation_id** – The record id for mediation that contains routing_keys and service endpoint

Returns A new *DIDXRequest* message to send to the other agent

create_request_implicit (*their_public_did*: str, *my_label*: str = None, *my_endpoint*: str = None, *mediation_id*: str = None) → aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequest
Create a request against a public DID only (no explicit invitation).

Parameters

- **their_public_did** – public DID to which to request a connection
- **my_label** – my label for request
- **my_endpoint** – my endpoint
- **mediation_id** – record id for mediation with routing_keys, service endpoint

Returns DID exchange request

create_response (*conn_rec*: aries_cloudagent.connections.models.conn_record.ConnRecord, *my_endpoint*: str = None, *mediation_id*: str = None) → aries_cloudagent.protocols.didexchange.v1_0.messages.response.DIDXResponse
Create a connection response for a received connection request.

Parameters

- **conn_rec** – The *ConnRecord* with a pending connection request
- **my_endpoint** – Current agent endpoint
- **mediation_id** – The record id for mediation that contains routing_keys and service endpoint

Returns New *DIDXResponse* message

receive_invitation (*invitation*: aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationMessage, *their_public_did*: str = None, *auto_accept*: bool = None, *alias*: str = None, *mediation_id*: str = None) → aries_cloudagent.connections.models.conn_record.ConnRecord
Create a new connection record to track a received invitation.

Parameters

- **invitation** – invitation to store
- **their_public_did** – their public DID
- **auto_accept** – set to auto-accept invitation (None to use config)
- **alias** – optional alias to set on record
- **mediation_id** – record id for mediation with routing_keys, service endpoint

Returns The new *ConnRecord* instance

receive_request (*request*: aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDXRequest, *recipient_did*: str, *recipient_verkey*: str = None, *my_endpoint*: str = None, *alias*: str = None, *auto_accept_implicit*: bool = None, *mediation_id*: str = None) → aries_cloudagent.connections.models.conn_record.ConnRecord
Receive and store a connection request.

Parameters

- **request** – The *DIDXRequest* to accept

- **recipient_did** – The (unqualified) recipient DID
- **recipient_verkey** – The recipient verkey: None for public recipient DID
- **my_endpoint** – My endpoint
- **alias** – Alias for the connection
- **auto_accept** – Auto-accept request against implicit invitation
- **mediation_id** – The record id for mediation that contains routing_keys and service endpoint

Returns The new or updated *ConnRecord* instance

session

Accessor for the current profile session.

Returns The profile session for this did exchange manager

verify_diddoc (*wallet:* *aries_cloudagent.wallet.base.BaseWallet*, *attached:* *aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator*) → *aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc*
Verify DIDDoc attachment and return signed data.

exception *aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManagerError* (*args, error_code: str = None, **kwargs)
Bases: *aries_cloudagent.core.error.BaseError*
Connection error.

aries_cloudagent.protocols.didexchange.v1_0.message_types module

Message type identifiers for Connections.

aries_cloudagent.protocols.didexchange.v1_0.routes module

Connection handling admin routes.

class *aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXAcceptInvitationQueryStringSchema*
Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*
Parameters and validators for accept invitation request query string.
my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.OneOf(['https://', 'http://'])>)>
my_label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
class *aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXAcceptRequestQueryStringSchema*
Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*
Parameters and validators for accept-request request query string.
mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

```

    my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
class aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXConnIdMatchInfoSchema (*args,
                                                                    **kwargs)

```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Path parameters and validators for request taking connection id.

```

    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXConnIdRefIdMatchInfoSchema (*args,
                                                                    **kwargs)

```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Path parameters and validators for request taking connection and ref ids.

```

    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    ref_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXCreateRequestImplicitQueryStr

```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for create-request-implicit request query string.

```

    mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
    my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
    my_label = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
    their_public_did = <fields.String(default=<marshmallow.missing>, attribute=None, valid
class aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXReceiveRequestImplicitQueryStr

```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for receive-request-implicit request query string.

```

    alias = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
    auto_accept = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=
    mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
    my_endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
aries_cloudagent.protocols.didexchange.v1_0.routes.didx_accept_invitation (request:

```

```

    <sphinx.ext.autodoc.impor
    ob-
    ject
    at
    0x7f760da42990>)

```

Request handler for accepting a stored connection invitation.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```

aries_cloudagent.protocols.didexchange.v1_0.routes.didx_accept_request (request:
    <sphinx.ext.autodoc.importe
    ob-
    ject
    at
    0x7f760da42990>)

```

Request handler for accepting a stored connection request.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```
aries_cloudagent.protocols.didexchange.v1_0.routes.didx_create_request_implicit(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760da42990>)
```

Request handler for creating a request to an implicit invitation.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```
aries_cloudagent.protocols.didexchange.v1_0.routes.didx_receive_request_implicit(request:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760da42990>)
```

Request handler for receiving a request against public DID's implicit invitation.

Parameters **request** – aiohttp request object

Returns The resulting connection record details

```
aries_cloudagent.protocols.didexchange.v1_0.routes.post_process_routes(app:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760da42990>)
```

Amend swagger API.

```
aries_cloudagent.protocols.didexchange.v1_0.routes.register(app:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760da42990>)
```

Register routes.

Submodules

aries_cloudagent.protocols.didexchange.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.discovery package

Subpackages

aries_cloudagent.protocols.discovery.v1_0 package

Subpackages

aries_cloudagent.protocols.discovery.v1_0.handlers package

Submodules

aries_cloudagent.protocols.discovery.v1_0.handlers.disclose_handler module

Handler for incoming disclose messages.

```
class aries_cloudagent.protocols.discovery.v1_0.handlers.disclose_handler.DiscloseHandler  
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for incoming disclose messages.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:  
        aries_cloudagent.messaging.responder.BaseResponder)  
    Message handler implementation.
```

aries_cloudagent.protocols.discovery.v1_0.handlers.query_handler module

Handler for incoming query messages.

```
class aries_cloudagent.protocols.discovery.v1_0.handlers.query_handler.QueryHandler  
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for incoming query messages.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:  
        aries_cloudagent.messaging.responder.BaseResponder)  
    Message handler implementation.
```

aries_cloudagent.protocols.discovery.v1_0.messages package

Submodules

aries_cloudagent.protocols.discovery.v1_0.messages.disclose module

Represents a feature discovery disclosure message.

```
class aries_cloudagent.protocols.discovery.v1_0.messages.disclose.Disclose (*,  
                                                                              pro-  
                                                                              to-  
                                                                              cols:  
                                                                              Se-  
                                                                              quence[Mapping[str,  
                                                                              Map-  
                                                                              ping[KT,  
                                                                              VT_co]]]  
                                                                              =  
                                                                              None,  
                                                                              **kwargs)  
  
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
```

Represents a feature discovery disclosure, the response to a query message.


```
class Meta
    Bases: object

    Disclose metadata.

    handler_class = 'aries_cloudagent.protocols.discovery.v1_0.handlers.disclose_handler'
    message_type = 'discover-features/1.0/disclose'
    schema_class = 'DiscloseSchema'

class aries_cloudagent.protocols.discovery.v1_0.messages.disclose.DiscloseSchema(*args,
                                                                                   **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema

    Disclose message schema used in serialization/deserialization.

    class Meta
        Bases: object

        DiscloseSchema metadata.

        model_class
            alias of Disclose

        unknown = 'exclude'

    protocols = <fields.List (default=<marshmallow.missing>, attribute=None, validate=None,
```

```
class aries_cloudagent.protocols.discovery.v1_0.messages.disclose.ProtocolDescriptorSchema
```

Bases: `marshmallow.schema.Schema`

Schema for an entry in the protocols list.

`opts` = `<marshmallow.schema.SchemaOpts object>`

`aries_cloudagent.protocols.discovery.v1_0.messages.query` module

Represents a feature discovery query message.

```

class aries_cloudagent.protocols.discovery.v1_0.messages.query.Query(*,
                                                                    query:
                                                                    str =
                                                                    None,
                                                                    comment:
                                                                    str =
                                                                    None,
                                                                    **kwargs)

Bases: aries_cloudagent.messaging.agent_message.AgentMessage

Represents a feature discovery query.

Used for inspecting what message types are supported by the agent.

class Meta
    Bases: object
    Query metadata.
    handler_class = 'aries_cloudagent.protocols.discovery.v1_0.handlers.query_handler.Q
    message_type = 'discover-features/1.0/query'
    schema_class = 'QuerySchema'

class aries_cloudagent.protocols.discovery.v1_0.messages.query.QuerySchema(*args,
                                                                              **kwargs)

Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema

Query message schema used in serialization/deserialization.

class Meta
    Bases: object
    QuerySchema metadata.
    model_class
        alias of Query
    unknown = 'exclude'
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    query = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```

Submodules

aries_cloudagent.protocols.discovery.v1_0.message_types module

Message type identifiers for Feature Discovery.

aries_cloudagent.protocols.discovery.v1_0.routes module

Feature discovery admin routes.

```

class aries_cloudagent.protocols.discovery.v1_0.routes.QueryFeaturesQueryStringSchema(*args,
                                                                                       **kwargs)

Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema

Query string parameters for feature query.

```

```

    query = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
class aries_cloudagent.protocols.discovery.v1_0.routes.QueryResultSchema (*args,
                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Result schema for the protocol list.

    results = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, r
aries_cloudagent.protocols.discovery.v1_0.routes.post_process_routes (app:
                                                                    <sphinx.ext.autodoc.importer._I
                                                                    object
                                                                    at
                                                                    0x7f760e745510>)

    Amend swagger API.
aries_cloudagent.protocols.discovery.v1_0.routes.query_features (request:
                                                                    <sphinx.ext.autodoc.importer._MockOb
                                                                    object      at
                                                                    0x7f760e745510>)

    Request handler for inspecting supported protocols.

    Parameters request – aiohttp request object

    Returns The diclosed protocols response
aries_cloudagent.protocols.discovery.v1_0.routes.register (app:
                                                                    <sphinx.ext.autodoc.importer._MockObject
                                                                    object      at
                                                                    0x7f760e745510>)

    Register routes.

```

Submodules

aries_cloudagent.protocols.discovery.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.introduction package

Subpackages

aries_cloudagent.protocols.introduction.v0_1 package

Subpackages

aries_cloudagent.protocols.introduction.v0_1.handlers package

Submodules

aries_cloudagent.protocols.introduction.v0_1.handlers.forward_invitation_handler module

Handler for incoming forward invitation messages.

class `aries_cloudagent.protocols.introduction.v0_1.handlers.forward_invitation_handler.ForwardInvitationHandler`
 Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`
 Handler for incoming forward invitation messages.
handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
 Message handler implementation.

`aries_cloudagent.protocols.introduction.v0_1.handlers.invitation_handler` module

Handler for incoming invitation messages.

class `aries_cloudagent.protocols.introduction.v0_1.handlers.invitation_handler.InvitationHandler`
 Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`
 Handler for incoming invitation messages.
handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
 Message handler implementation.

`aries_cloudagent.protocols.introduction.v0_1.handlers.invitation_request_handler` module

Handler for incoming invitation request messages.

class `aries_cloudagent.protocols.introduction.v0_1.handlers.invitation_request_handler.InvitationRequestHandler`
 Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`
 Handler for incoming invitation request messages.
handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
 Message handler implementation.

`aries_cloudagent.protocols.introduction.v0_1.messages` package

Submodules

`aries_cloudagent.protocols.introduction.v0_1.messages.forward_invitation` module

Represents a forwarded invitation from another agent.

```
class aries_cloudagent.protocols.introduction.v0_1.messages.forward_invitation.ForwardInvitation
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing an invitation to be forwarded.

```
class Meta
```

Bases: *object*

Metadata for a forwarded invitation.

```
    handler_class = 'aries_cloudagent.protocols.introduction.v0_1.handlers.forward_invitation'
```

```
    message_type = 'introduction-service/0.1/forward-invitation'
```

```
    schema_class = 'ForwardInvitationSchema'
```

```
class aries_cloudagent.protocols.introduction.v0_1.messages.forward_invitation.ForwardInvitationSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

ForwardInvitation request schema class.

```
class Meta
```

Bases: *object*

ForwardInvitation request schema metadata.

```
    model_class
```

alias of *ForwardInvitation*

```
    invitation = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
    message = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
```

aries_cloudagent.protocols.introduction.v0_1.messages.invitation module

Represents an invitation returned to the introduction service.

```
class aries_cloudagent.protocols.introduction.v0_1.messages.invitation.Invitation(*,
in-
vi-
ta-
tion:
aries_cloudagent.protocols.introduction.v0_1.messages.invitation.Invitation =
None,
mes-
sage:
str
=
None,
**kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing an invitation returned to the introduction service.

```
class Meta
```

Bases: *object*

Metadata for an invitation.

```
handler_class = 'aries_cloudagent.protocols.introduction.v0_1.handlers.invitation_handler'
```

```
message_type = 'introduction-service/0.1/invitation'
```

```
schema_class = 'InvitationSchema'
```

```
class aries_cloudagent.protocols.introduction.v0_1.messages.invitation.InvitationSchema(*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Invitation request schema class.

```
class Meta
```

Bases: *object*

Invitation request schema metadata.

```
model_class
```

alias of *Invitation*

```
unknown = 'exclude'
```

```
invitation = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
message = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
```

aries_cloudagent.protocols.introduction.v0_1.messages.invitation_request module

Represents an request for an invitation from the introduction service.

```
class aries_cloudagent.protocols.introduction.v0_1.messages.invitation_request.InvitationRequest
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing an invitation request.

```
class Meta
```

Bases: *object*

Metadata for an invitation request.

```
    handler_class = 'aries_cloudagent.protocols.introduction.v0_1.handlers.invitation_request_handler'
```

```
    message_type = 'introduction-service/0.1/invitation-request'
```

```
    schema_class = 'InvitationRequestSchema'
```

```
class aries_cloudagent.protocols.introduction.v0_1.messages.invitation_request.InvitationRequestSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Invitation request schema class.

```
class Meta
```

Bases: *object*

Invitation request schema metadata.

```
    model_class
```

alias of *InvitationRequest*

```
    unknown = 'exclude'
```

```
    message = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_present=True)>
```

```
    responder = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_present=True)>
```

Submodules

`aries_cloudagent.protocols.introduction.v0_1.base_service` module

Introduction service base classes.

```
class aries_cloudagent.protocols.introduction.v0_1.base_service.BaseIntroductionService (concrete class)
```

Bases: *abc.ABC*

Service handler for allowing connections to exchange invitations.

return_invitation (*target_connection_id: str, invitation: aries_cloudagent.protocols.introduction.v0_1.messages.invitation, outbound_handler*)

Handle the forwarding of an invitation to the responder.

Parameters

- **target_connection_id** – The ID of the connection sending the Invitation
- **invitation** – The received Invitation message
- **outbound_handler** – The outbound handler coroutine for sending a message

classmethod service_handler ()

Quick accessor for conductor to use.

start_introduction (*init_connection_id: str, target_connection_id: str, outbound_handler, message: str = None*)

Start the introduction process between two connections.

Parameters

- **init_connection_id** – The connection initiating the request
- **target_connection_id** – The connection which is asked for an invitation
- **outbound_handler** – The outbound handler coroutine for sending a message
- **message** – The message to use when requesting the invitation

exception `aries_cloudagent.protocols.introduction.v0_1.base_service.IntroductionError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Generic introduction service error.

aries_cloudagent.protocols.introduction.v0_1.demo_service module

Introduction service demo classes.

class `aries_cloudagent.protocols.introduction.v0_1.demo_service.DemoIntroductionService` (concrete class, aries_cloudagent.protocols.introduction.v0_1.base_service.BaseIntroductionService)

Bases: `aries_cloudagent.protocols.introduction.v0_1.base_service.BaseIntroductionService`

Service handler for allowing connections to exchange invitations.

RECORD_TYPE = 'introduction_record'

return_invitation (*target_connection_id: str, invitation: aries_cloudagent.protocols.introduction.v0_1.messages.invitation, outbound_handler*)

Handle the forwarding of an invitation to the responder.

Parameters

- **target_connection_id** – The ID of the connection sending the Invitation
- **invitation** – The received Invitation message
- **outbound_handler** – The outbound handler coroutine for sending a message

start_introduction (*init_connection_id: str, target_connection_id: str, message: str, outbound_handler*)

Start the introduction process between two connections.

Parameters

- **init_connection_id** – The connection initiating the request
- **target_connection_id** – The connection which is asked for an invitation
- **outbound_handler** – The outbound handler coroutine for sending a message
- **message** – The message to use when requesting the invitation

aries_cloudagent.protocols.introduction.v0_1.message_types module

Message type identifiers for Introductions.

aries_cloudagent.protocols.introduction.v0_1.routes module

Introduction service admin routes.

class aries_cloudagent.protocols.introduction.v0_1.routes.ConnIdMatchInfoSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Path parameters and validators for request taking connection id.

conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

class aries_cloudagent.protocols.introduction.v0_1.routes.IntroModuleResponseSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Response schema for Introduction Module.

class aries_cloudagent.protocols.introduction.v0_1.routes.IntroStartQueryStringSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Query string parameters for request to start introduction.

message = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

target_connection_id = <fields.String(default=<marshmallow.missing>, attribute=None, v

aries_cloudagent.protocols.introduction.v0_1.routes.introduction_start (request: *<sphinx.ext.autodoc.importer.Object at 0x7f760e293e90>*)

Request handler for starting an introduction.

Parameters **request** – aiohttp request object

aries_cloudagent.protocols.introduction.v0_1.routes.post_process_routes (app: *<sphinx.ext.autodoc.importer.Object at 0x7f760e293e90>*)

Amend swagger API.

```
aries_cloudagent.protocols.introduction.v0_1.routes.register(app:
                                                         <sphinx.ext.autodoc.importer._MockObject
                                                         object          at
                                                         0x7f760e293e90>)
```

Register routes.

Submodules

`aries_cloudagent.protocols.introduction.definition` module

Version definitions for this protocol.

`aries_cloudagent.protocols.issue_credential` package

Subpackages

`aries_cloudagent.protocols.issue_credential.v1_0` package

Subpackages

`aries_cloudagent.protocols.issue_credential.v1_0.handlers` package

Submodules

`aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_ack_handler` module

Credential ack message handler.

```
class aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_ack_handler.Cred
Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for credential acks.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
        aries_cloudagent.messaging.responder.BaseResponder)
```

Message handler logic for credential acks.

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_issue_handler` module

Credential issue message handler.

```
class aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_issue_handler.C
Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for credential offers.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Message handler logic for credential offers.

Parameters

- **context** – request context
- **responder** – responder callback

aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_offer_handler module

Credential offer message handler.

class `aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_offer_handler.C`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for credential offers.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Message handler logic for credential offers.

Parameters

- **context** – request context
- **responder** – responder callback

aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_proposal_handler module

Credential proposal message handler.

class `aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_proposal_handler.C`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for credential proposals.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Message handler logic for credential proposals.

Parameters

- **context** – proposal context
- **responder** – responder callback

aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_request_handler module

Credential request message handler.

class `aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_request_handler.C`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for credential requests.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Message handler logic for credential requests.

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v1_0.messages` package

Subpackages

`aries_cloudagent.protocols.issue_credential.v1_0.messages.inner` package

Submodules

`aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview` module

A credential preview inner object.

class `aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview.C`

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Class representing a preview of an attribute.

class **Meta**

Bases: `object`

Attribute preview metadata.

schema_class = 'CredAttrSpecSchema'

b64_decoded_value() → str

Value, base64-decoded if applicable.

static **list_plain**(*plain: dict*)

Return a list of *CredAttrSpec* without MIME types from names/values.

Parameters **plain** – dict mapping names to values

Returns List of *CredAttrSpecs* with no MIME types

class `aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview.C`

Bases: `aries_cloudagent.messaging.models.base.BaseModelSchema`

Attribute preview schema.

class **Meta**

Bases: `object`

Attribute preview schema metadata.

```
    model_class
        alias of CredAttrSpec

    unknown = 'exclude'

    mime_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
    name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
    value = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
class aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview.C
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a credential preview inner object.

```
class Meta
    Bases: object

    Credential preview metadata.

    message_type = 'issue-credential/1.0/credential-preview'
    schema_class = 'CredentialPreviewSchema'

    attr_dict (decode: bool = False)
        Return name:value pair per attribute.

        Parameters
        decode – whether first to decode attributes with MIME type

    mime_types ()
        Return per-attribute mapping from name to MIME type.

        Return empty dict if no attribute has MIME type.
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview.C
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Credential preview schema.

```
class Meta
    Bases: object

    Credential preview schema metadata.

    model_class
        alias of CredentialPreview

    unknown = 'exclude'

    attributes = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=No
```

Submodules

`aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_ack` module

A credential ack message.

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_ack.CredentialAck
    Bases: aries_cloudagent.messaging.ack.message.Ack
```

Class representing a credential ack message.

```
class Meta
```

Bases: *object*

Credential ack metadata.

```
    handler_class = 'aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_ack'
```

```
    message_type = 'issue-credential/1.0/ack'
```

```
    schema_class = 'CredentialAckSchema'
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_ack.CredentialAckSchema
```

Bases: *aries_cloudagent.messaging.ack.message.AckSchema*

Credential ack schema.

```
class Meta
```

Bases: *object*

Schema metadata.

```
    model_class
```

alias of *CredentialAck*

```
    unknown = 'exclude'
```

`aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_issue` module

A credential content message.

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_issue.CredentialIssue
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a credential.

class Meta

Bases: *object*

Credential metadata.

handler_class = 'aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential'

message_type = 'issue-credential/1.0/issue-credential'

schema_class = 'CredentialIssueSchema'

indy_credential (*index: int = 0*)

Retrieve and decode indy credential from attachment.

Parameters **index** – ordinal in attachment list to decode and return (typically, list has length 1)

classmethod wrap_indy_credential (*indy_cred: dict*) → *aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator*
Convert an indy credential offer to an attachment decorator.

class *aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_issue.CredentialIssue*

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential schema.

class Meta

Bases: *object*

Credential schema metadata.

model_class

alias of *CredentialIssue*

unknown = 'exclude'

comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

credentials_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, val

aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer module

A credential offer content message.


```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer.CredentialOffer:
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a credential offer.

```
class Meta
```

Bases: *object*

CredentialOffer metadata.

```
    handler_class = 'aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_offer_handler'
```

```
    message_type = 'issue-credential/1.0/offer-credential'
```

```
    schema_class = 'CredentialOfferSchema'
```

```
    indy_offer (index: int = 0) → dict
```

Retrieve and decode indy offer from attachment.

Parameters *index* – ordinal in attachment list to decode and return (typically, list has length 1)

```
    classmethod wrap_indy_offer (indy_offer: dict) → aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator
```

Convert an indy credential offer to an attachment decorator.

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer.CredentialOfferSchema:
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential offer schema.

```
class Meta
```

Bases: *object*

Credential offer schema metadata.

```
    model_class
```

alias of *CredentialOffer*

```
    unknown = 'exclude'
```

```
comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, val
offers_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate
```

`aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal` module

A credential proposal content message.

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal.Credent
```

Bases: `aries_cloudagent.messaging.agent_message.AgentMessage`

Class representing a credential proposal.

```
class Meta
    Bases: object
    CredentialProposal metadata.
    handler_class = 'aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_proposal'
    message_type = 'issue-credential/1.0/propose-credential'
    schema_class = 'CredentialProposalSchema'

class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal.CredentialProposal
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Credential proposal schema.

class Meta
    Bases: object
    Credential proposal schema metadata.
    model_class
        alias of CredentialProposal
    unknown = 'exclude'
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    credential_proposal = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    schema_issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
```

`aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request` module

A credential request content message.

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request.CredentialRequest
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a credential request.

```
class Meta
```

Bases: *object*

CredentialRequest metadata.

```
    handler_class = 'aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_request_handler'
```

```
    message_type = 'issue-credential/1.0/request-credential'
```

```
    schema_class = 'CredentialRequestSchema'
```

```
indy_cred_req(index: int = 0)
```

Retrieve and decode indy credential request from attachment.

Parameters *index* – ordinal in attachment list to decode and return (typically, list has length 1)

```
classmethod wrap_indy_cred_req(indy_cred_req: dict) → aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator
    Convert an indy credential request to an attachment decorator.
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request.CredentialRequestSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential request schema.

```
class Meta
```

Bases: *object*

Credential request schema metadata.

```
    model_class
```

alias of *CredentialRequest*

```
    unknown = 'exclude'
```

```
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_present=True)>
```

```
    requests_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, if_present=True)>
```

aries_cloudagent.protocols.issue_credential.v1_0.models package

Submodules

aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange module

Aries#0036 v1.0 credential exchange information with non-secrets storage.

```
class aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10Creden
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseExchangeRecord*

Represents an Aries#0036 credential exchange.

INITIATOR_EXTERNAL = 'external'

INITIATOR_SELF = 'self'

class Meta

Bases: *object*

CredentialExchange metadata.

schema_class = 'V10CredentialExchangeSchema'

RECORD_ID_NAME = 'credential_exchange_id'

RECORD_TYPE = 'credential_exchange_v10'

ROLE HOLDER = 'holder'

ROLE_ISSUER = 'issuer'

STATE_ACKED = 'credential_acked'

STATE_CREDENTIAL_RECEIVED = 'credential_received'

STATE_ISSUED = 'credential_issued'

STATE_OFFER_RECEIVED = 'offer_received'

STATE_OFFER_SENT = 'offer_sent'

STATE_PROPOSAL_RECEIVED = 'proposal_received'

STATE_PROPOSAL_SENT = 'proposal_sent'

STATE_REQUEST_RECEIVED = 'request_received'

STATE_REQUEST_SENT = 'request_sent'

TAG_NAMES = {'thread_id'}

WEBHOOK_TOPIC = 'issue_credential'

credential_exchange_id

Accessor for the ID associated with this exchange.

record_value

Accessor for the JSON record value generated for this credential exchange.

classmethod retrieve_by_connection_and_thread (*session:*

aries_cloudagent.core.profile.ProfileSession,

connection_id: *str,*

thread_id: *str*) →

aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange

Retrieve a credential exchange record by connection and thread ID.

class *aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange*

Bases: *aries_cloudagent.messaging.models.base_record.BaseExchangeSchema*

Schema to allow serialization/deserialization of credential exchange records.

class Meta

Bases: *object*

V10CredentialExchangeSchema metadata.

```
model_class
    alias of V10CredentialExchange

auto_issue = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
auto_offer = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
auto_remove = <fields.Boolean(default=True, attribute=None, validate=None, required=False)
connection_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
credential = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
credential_definition_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
credential_exchange_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
credential_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
credential_offer = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
credential_offer_dict = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
credential_proposal_dict = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
credential_request = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
credential_request_metadata = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
error_msg = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
initiator = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
parent_thread_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
raw_credential = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
revoc_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
revocation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
thread_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

Submodules

aries_cloudagent.protocols.issue_credential.v1_0.manager module

Classes to manage credentials.

class aries_cloudagent.protocols.issue_credential.v1_0.manager.**CredentialManager** (*profile:* *aries_cloudagent*)

Bases: *object*

Class for managing credentials.

create_offer (*cred_ex_record:* *aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange*, *comment:* *str = None*) → *Tuple*[*aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange*, *aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer.CredentialOffer*]
Create a credential offer, update credential exchange record.

Parameters

- **cred_ex_record** – Credential exchange to create offer for
- **comment** – optional human-readable comment to set in offer message

Returns A tuple (credential exchange record, credential offer message)

```
create_proposal (connection_id: str, *, auto_offer: bool = None, auto_remove:
                  bool = None, comment: str = None, credential_preview:
                  aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview.CredentialPreview
                  = None, schema_id: str = None, schema_issuer_did: str = None,
                  schema_name: str = None, schema_version: str = None, cred_def_id:
                  str = None, issuer_did: str = None, trace: bool = False) →
                  aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange
```

Create a credential proposal.

Parameters

- **connection_id** – Connection to create proposal for
- **auto_offer** – Should this proposal request automatically be handled to offer a credential
- **auto_remove** – Should the record be automatically removed on completion
- **comment** – Optional human-readable comment to include in proposal
- **credential_preview** – The credential preview to use to create the credential proposal
- **schema_id** – Schema id for credential proposal
- **schema_issuer_did** – Schema issuer DID for credential proposal
- **schema_name** – Schema name for credential proposal
- **schema_version** – Schema version for credential proposal
- **cred_def_id** – Credential definition id for credential proposal
- **issuer_did** – Issuer DID for credential proposal

Returns Resulting credential exchange record including credential proposal

```
create_request (cred_ex_record: aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange,
                 holder_did: str) → Tuple[aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange,
                 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request.CredentialRequest]
```

Create a credential request.

Parameters

- **cred_ex_record** – Credential exchange record for which to create request
- **holder_did** – holder DID

Returns A tuple (credential exchange record, credential request message)

```
issue_credential (cred_ex_record: aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange,
                  *, comment: str = None, retries: int = 5) → Tuple[aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange,
                  aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_issue.CredentialIssue]
```

Issue a credential.

Parameters

- **cred_ex_record** – The credential exchange record for which to issue a credential

- **comment** – optional human-readable comment pertaining to credential issue

Returns (Updated credential exchange record, credential message)

Return type Tuple

prepare_send (*connection_id: str, credential_proposal: aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal.CredentialProposal, auto_remove: bool = None*) → Tuple[aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange, aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer.CredentialOffer]

Set up a new credential exchange for an automated send.

Parameters

- **connection_id** – Connection to create offer for
- **credential_proposal** – The credential proposal with preview
- **auto_remove** – Flag to automatically remove the record on completion

Returns A tuple of the new credential exchange record and credential offer message

profile

Accessor for the current profile instance.

Returns The profile instance for this credential manager

receive_credential (*message: aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_issue.CredentialIssue, connection_id: str*) → aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange

Receive a credential from an issuer.

Hold in storage potentially to be processed by controller before storing.

Returns Credential exchange record, retrieved and updated

receive_credential_ack (*message: aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_ack.CredentialAck, connection_id: str*) → aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange

Receive credential ack from holder.

Returns credential exchange record, retrieved and updated

receive_offer (*message: aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer.CredentialOffer, connection_id: str*) → aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange

Receive a credential offer.

Returns The credential exchange record, updated

receive_proposal (*message: aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal.CredentialProposal, connection_id: str*) → aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange

Receive a credential proposal.

Returns The resulting credential exchange record, created

receive_request (*message: aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request.CredentialRequest, connection_id: str*)

Receive a credential request.

Parameters **credential_request_message** – Credential request to receive

Returns credential exchange record, retrieved and updated

store_credential (*cred_ex_record: aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange, credential_id: str = None*) → Tuple[aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange, aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_ack.CredentialAck]

Store a credential in holder wallet; send ack to issuer.

Parameters

- **cred_ex_record** – credential exchange record with credential to store and ack

- **credential_id** – optional credential identifier to override default on storage

Returns (Updated credential exchange record, credential ack message)

Return type Tuple

exception `aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManagerError` (`*`

Bases: `aries_cloudagent.core.error.BaseError`

Credential error.

aries_cloudagent.protocols.issue_credential.v1_0.message_types module

Message and inner object type identifiers for Connections.

aries_cloudagent.protocols.issue_credential.v1_0.routes module

Credential exchange admin routes.

class `aries_cloudagent.protocols.issue_credential.v1_0.routes.CredExIdMatchInfoSchema` (`*`*args*,
***kwargs*)

Bases: `aries_cloudagent.messaging.models.base.OpenAPISchema`

Path parameters and validators for request taking credential exchange id.

cred_ex_id = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=<a`

class `aries_cloudagent.protocols.issue_credential.v1_0.routes.CredIdMatchInfoSchema` (`*`*args*,
***kwargs*)

Bases: `aries_cloudagent.messaging.models.base.OpenAPISchema`

Path parameters and validators for request taking credential id.

credential_id = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=`

class `aries_cloudagent.protocols.issue_credential.v1_0.routes.IssueCredentialModuleResponse`

Bases: `aries_cloudagent.messaging.models.base.OpenAPISchema`

Response schema for Issue Credential Module.

class `aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialCreateSchema` (`*`*args*,
***kwargs*)

Bases: `aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema`

Base class for request schema for sending credential proposal admin message.

auto_remove = `<fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=`

comment = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=None,`

cred_def_id = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=<`

credential_proposal = `<fields.Nested(default=<marshmallow.missing>, attribute=None, va`

issuer_did = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=<a`

```
    schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
    schema_issuer_did = <fields.String(default=<marshmallow.missing>, attribute=None, vali
    schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=N
    schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validat
    trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None,

class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeListQuer

    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Parameters and validators for credential exchange list query.

    connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=N
    role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(c
    state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(
    thread_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None,

class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeListRes

    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Result schema for Aries#0036 v1.0 credential exchange query.

    results = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, r

class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialIssueRequestSch

    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Request schema for sending credential issue admin message.

    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialOfferRequestSch

    Bases: aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema
    Request schema for sending credential offer admin message.

    auto_issue = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=N
    auto_remove = <fields.Boolean(default=True, attribute=None, validate=None, required=Fa
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=N
    cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
    credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, val
    trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None,

class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProblemReportRe

    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Request schema for sending problem report.

    explain_ltxt = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProposalRequestSchema
```

Bases: `aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProposalRequestSchemaBase`

Request schema for sending credential proposal on mandatory proposal preview.

```
credential_proposal = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProposalRequestSchema
```

Bases: `aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProposalRequestSchemaBase`

Request schema for sending credential proposal on optional proposal preview.

```
credential_proposal = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProposalRequestSchema
```

Bases: `aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema`

Base class for request schema for sending credential proposal admin message.

```
auto_remove = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
schema_issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialStoreRequestSchema
```

Bases: `aries_cloudagent.messaging.models.base.OpenAPISchema`

Request schema for sending a credential store admin message.

```
credential_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_create(request: <sphinx.ext.autodoc.directive.AutodocDirective object at 0x7f760dc00000>)
```

Request handler for creating a credential from attr values.

The internal credential record will be created without the credential being sent to any connection. This can be used in conjunction with the *oob* protocols to bind messages to an out of band message.

Parameters `request` – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_create_free_offer
```

Request handler for creating free credential offer.

Unlike with *send-offer*, this credential exchange is not tied to a specific connection. It must be dispatched out-of-band by the controller.

Parameters `request` – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_issue (request:
    <sphinx.ext.autodoc.mock.Mock object at 0x7f760dcc69d0>)
```

Request handler for sending credential.

Parameters `request` – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_list (request:
    <sphinx.ext.autodoc.mock.Mock object at 0x7f760dcc69d0>)
```

Request handler for searching credential exchange records.

Parameters `request` – aiohttp request object

Returns The connection list response

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_problem_report
```

Request handler for sending problem report.

Parameters `request` – aiohttp request object

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_remove (request:
    <sphinx.ext.autodoc.mock.Mock object at 0x7f760dcc69d0>)
```

Request handler for removing a credential exchange record.

Parameters `request` – aiohttp request object

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_retrieve(request:
    <sphinx.ext.autodoc.Document object at 0x7f760
```

Request handler for fetching single credential exchange record.

Parameters **request** – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_send(request:
    <sphinx.ext.autodoc.Document object at 0x7f760dcc69
```

Request handler for sending credential from issuer to holder from attr values.

If both issuer and holder are configured for automatic responses, the operation ultimately results in credential issue; otherwise, the result waits on the first response not automated; the credential exchange record retains state regardless.

Parameters **request** – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_send_bound_offer
```

Request handler for sending bound credential offer.

A holder initiates this sequence with a credential proposal; this message responds with an offer bound to the proposal.

Parameters **request** – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_send_free_offer
```

Request handler for sending free credential offer.

An issuer initiates a such a credential offer, free from any holder-initiated corresponding credential proposal with preview.

Parameters **request** – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_send_proposal` (request: aiohttp.Request) - Request handler for sending credential proposal.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_send_request` (request: aiohttp.Request) - Request handler for sending credential request.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v1_0.routes.credential_exchange_store` (request: aiohttp.Request) - Request handler for storing credential.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v1_0.routes.post_process_routes` (app: aiohttp.web.Application) - Amend swagger API.

`aries_cloudagent.protocols.issue_credential.v1_0.routes.register` (app: aiohttp.web.Application) - Register routes.

aries_cloudagent.protocols.issue_credential.v2_0 package

Subpackages

aries_cloudagent.protocols.issue_credential.v2_0.handlers package

Submodules

`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_ack_handler` module

Credential ack message handler.

```
class aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_ack_handler.V20CredAckHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for credential acks.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:
         aries_cloudagent.messaging.responder.BaseResponder)
    Message handler logic for credential acks.
```

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_issue_handler` module

Credential issue message handler.

```
class aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_issue_handler.V20CredIssueHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for credential offers.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:
         aries_cloudagent.messaging.responder.BaseResponder)
    Message handler logic for credential offers.
```

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_offer_handler` module

Credential offer message handler.

```
class aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_offer_handler.V20CredOfferHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for credential offers.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:
         aries_cloudagent.messaging.responder.BaseResponder)
    Message handler logic for credential offers.
```

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_proposal_handler` module

Credential proposal message handler.

class `aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_proposal_handler.V20C`

Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for credential proposals.

handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
Message handler logic for credential proposals.

Parameters

- **context** – proposal context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_request_handler` module

Credential request message handler.

class `aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_request_handler.V20Cr`

Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Message handler class for credential requests.

handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
Message handler logic for credential requests.

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.issue_credential.v2_0.messages` package

Subpackages

`aries_cloudagent.protocols.issue_credential.v2_0.messages.inner` package

Submodules

`aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.cred_preview` module

Credential preview inner object.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.cred_preview.V20CredPreview
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Attribute preview.

```
class Meta
```

Bases: *object*

Attribute preview metadata.

```
    schema_class = 'V20CredAttrSpecSchema'
```

```
    b64_decoded_value() → str
```

Value, base64-decoded if applicable.

```
    static list_plain(plain: dict) → Sequence[aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.cred_preview.V20CredAttrSpec]
    Return a list of V20CredAttrSpec (copies), absent any MIME types.
```

Parameters *plain* – dict mapping names to values

Returns List of *V20CredAttrSpec* (copies), absent any MIME types

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.cred_preview.V20CredPreviewSchema
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Attribute preview schema.

```
class Meta
```

Bases: *object*

Attribute preview schema metadata.

```
    model_class
```

alias of *V20CredAttrSpec*

```
    unknown = 'exclude'
```

```
    mime_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
```

```
    name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
```

```
    value = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
```

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.cred_preview.V20CredPreview
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Credential preview.

```
class Meta
```

Bases: *object*

Credential preview metadata.

```
message_type = 'issue-credential/2.0/credential-preview'
```

```
schema_class = 'V20CredPreviewSchema'
```

```
attr_dict (decode: bool = False)
```

Return name:value pair per attribute.

Parameters **decode** – whether first to decode attributes with MIME type

```
mime_types ()
```

Return per-attribute mapping from name to MIME type.

Return empty dict if no attribute has MIME type.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.cred_preview.V20CredPreviewSchema
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Credential preview schema.

```
class Meta
```

Bases: *object*

Credential preview schema metadata.

```
model_class
```

alias of *V20CredPreview*

```
unknown = 'exclude'
```

```
attributes = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
```

Submodules

aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ack module

Credential ack message.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ack.V20CredAck (**kwargs)
```

Bases: *aries_cloudagent.messaging.ack.message.Ack*

Credential ack.

```
class Meta
```

Bases: `object`

Credential ack metadata.

```
handler_class = 'aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_ack'
```

```
message_type = 'issue-credential/2.0/ack'
```

```
schema_class = 'V20CredAckSchema'
```

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ack.V20CredAckSchema (
```

Bases: `aries_cloudagent.messaging.ack.message.AckSchema`

Credential ack schema.

```
class Meta
```

Bases: `object`

Schema metadata.

```
model_class
```

alias of `V20CredAck`

```
unknown = 'exclude'
```

aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format module

Credential format inner object.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.FormatSpec (aries
```

aka,

de-

tail)

Bases: `tuple`

```
aka
```

Alias for field number 1

```
aries
```

Alias for field number 0

```
detail
```

Alias for field number 2

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.V20CredFormat (
```

Bases: `aries_cloudagent.messaging.models.base.BaseModel`

Credential format.

```
class Format
```

Bases: `enum.Enum`

Proposal credential format.

```
DIF = FormatSpec(aries='dif/credential-manifest@v1.0', aka={'jsonld', 'dif', 'w3c'})
```

```
INDY = FormatSpec(aries='hlindy-zkp-v1.0', aka={'hlindy', 'hyperledgerindy', 'indy'})
```

```
aka
```

Accessor for alternative identifier list.

```
aries
```

Accessor for aries identifier.

```
detail
```

Accessor for credential exchange detail class.

```
get = <bound method V20CredFormat.Format.get of <enum 'Format'>>
```

```
get_attachment_data(formats: Sequence[V20CredFormat], attachments: Sequence[aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator])  
Find attachment of current format, base64-decode and return its data.
```

```
validate_filter(data: Mapping[KT, VT_co])
```

Raise `ValidationError` for wrong filtration criteria.

```
class Meta
```

Bases: `object`

Credential format metadata.

```
schema_class = 'V20CredFormatSchema'
```

```
format
```

Return format.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.V20CredFormatSchema
```

Bases: `aries_cloudagent.messaging.models.base.BaseModelSchema`

Credential format schema.

```
class Meta
```

Bases: `object`

Credential format schema metadata.

```
model_class
```

alias of `V20CredFormat`

```
unknown = 'exclude'
```

```
attach_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
format_ = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf>>
```

aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue module

Credential issue message.

```

class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue.V20CredIssue(_id: str, *, re-
pla
men
str
=
Non
com
men
str
=
Non
for-
mat
Se-
que
=
Non
cre-
den
tial
Se-
que
=
Non
**k

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Credential issue message.

```
class Meta
```

Bases: *object*

V20CredIssue metadata.

```
handler_class = 'aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_iss
```

```
message_type = 'issue-credential/2.0/issue-credential'
```

```
schema_class = 'V20CredIssueSchema'
```

```
cred (fmt: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.V20CredFormat.Format
= None) → dict
Return attached credential.
```

Parameters *fmt* – format of attachment in list to decode and return

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue.V20CredIssueSche
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential issue schema.

```
class Meta
```

Bases: *object*

Credential issue schema metadata.

```

model_class
    alias of V20CredIssue

unknown = 'exclude'

comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
credentials_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, val
formats = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,
replacement_id = <fields.String(default=<marshmallow.missing>, attribute=None, validat

```

aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer module

Credential offer message.

```

class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.V20CredOffer(_id
    str
    =
    Non
    *,
    re-
    pla
    men
    str
    =
    Non
    com
    men
    str
    =
    Non
    cre-
    den
    tial
    arie
    =
    Non
    for-
    mat
    Se-
    que
    =
    Non
    of-
    fers
    Se-
    que
    =
    Non
    **k

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Credential offer.


```

class Meta
    Bases: object
    V20CredOffer metadata.

    handler_class = 'aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_off
    message_type = 'issue-credential/2.0/offer-credential'
    schema_class = 'V20CredOfferSchema'

offer (fmt: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.V20CredFormat.Format
      = None) → dict
    Return attached offer.

```

Parameters **fmt** – format of attachment in list to decode and return

```

class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.V20CredOfferSche

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential offer schema.

```

class Meta
    Bases: object
    Credential offer schema metadata.

    model_class
        alias of V20CredOffer

    unknown = 'exclude'

    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, val
    formats = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,
    offers_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate
    replacement_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate

```

aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal module

Credential proposal message.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal.V20CredPropo
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Credential proposal.

```
class Meta
```

Bases: *object*

V20CredProposal metadata.

```
    handler_class = 'aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_pro
```

```
    message_type = 'issue-credential/2.0/propose-credential'
```

```
    schema_class = 'V20CredProposalSchema'
```

```
    filter (fmt: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.V20CredFormat.Format
            = None) → dict
    Return attached filter.
```

Parameters *fmt* – format of attachment in list to decode and return

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal.V20CredPropo
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential proposal schema.

```
class Meta
```

Bases: *object*

Credential proposal schema metadata.

```
model_class
    alias of V20CredProposal

unknown = 'exclude'

comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, val
filters_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, validat
formats = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,
validate_fields(data, **kwargs)
    Validate filter per format.
```

`aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_request` module

Credential request message.

```
class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_request.V20CredRequest
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Credential request.

```
class Meta
```

Bases: *object*

V20CredRequest metadata.

```
handler_class = 'aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_req
```

```
message_type = 'issue-credential/2.0/request-credential'
```

```
schema_class = 'V20CredRequestSchema'
```

cred_request (*fmt: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format.V20CredFormat.Format = None*) → dict
 Return attached credential request.

Parameters **fmt** – format of attachment in list to decode and return

class aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_request.V20CredRequest

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Credential request schema.

class Meta

Bases: *object*

Credential request schema metadata.

model_class

alias of *V20CredRequest*

unknown = 'exclude'

comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

formats = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,

requests_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, valida

aries_cloudagent.protocols.issue_credential.v2_0.models package

Package-wide code and data.

Subpackages

aries_cloudagent.protocols.issue_credential.v2_0.models.detail package

Submodules

aries_cloudagent.protocols.issue_credential.v2_0.models.detail.dif module

DIF-specific credential exchange information with non-secrets storage.

class aries_cloudagent.protocols.issue_credential.v2_0.models.detail.dif.V20CredExRecordDIF

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Credential exchange DIF detail record.

```
class Meta
    Bases: object
    V20CredExRecordDIF metadata.

    schema_class = 'V20CredExRecordDIFSchema'
    RECORD_ID_NAME = 'cred_ex_dif_id'
    RECORD_TYPE = 'dif_cred_ex_v20'
    TAG_NAMES = {'cred_ex_id'}
    WEBHOOK_TOPIC = 'issue_credential_v2_0_dif'

    cred_ex_dif_id
        Accessor for the ID associated with this exchange.

    record_value
        Accessor for the JSON record value generated for this credential exchange.

    classmethod retrieve_by_cred_ex_id (session: aries_cloudagent.core.profile.ProfileSession,
                                       cred_ex_id: str) →
        aries_cloudagent.protocols.issue_credential.v2_0.models.detail.dif.V20CredExRecordDIF
        Retrieve a credential exchange DIF detail record by its cred ex id.
```

```
class aries_cloudagent.protocols.issue_credential.v2_0.models.detail.dif.V20CredExRecordDIFSchema
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

Credential exchange DIF detail record detail schema.

```
class Meta
    Bases: object
    Credential exchange DIF detail record schema metadata.

    model_class
        alias of V20CredExRecordDIF

    unknown = 'exclude'

    cred_ex_dif_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=No
    cred_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=No
    item = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, requ
```

aries_cloudagent.protocols.issue_credential.v2_0.models.detail.indy module

Indy-specific credential exchange information with non-secrets storage.

```
class aries_cloudagent.protocols.issue_credential.v2_0.models.detail.indy.V20CredExRecordIndy
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Credential exchange indy detail record.

```
class Meta
```

Bases: *object*

V20CredExRecordIndy metadata.

```
    schema_class = 'V20CredExRecordIndySchema'
```

```
RECORD_ID_NAME = 'cred_ex_indy_id'
```

```
RECORD_TYPE = 'indy_cred_ex_v20'
```

```
TAG_NAMES = {'cred_ex_id'}
```

```
WEBHOOK_TOPIC = 'issue_credential_v2_0_indy'
```

```
cred_ex_indy_id
```

Accessor for the ID associated with this exchange.

```
record_value
```

Accessor for the JSON record value generated for this credential exchange.

```
classmethod retrieve_by_cred_ex_id(session: aries_cloudagent.core.profile.ProfileSession,  
                                cred_ex_id: str) →
```

aries_cloudagent.protocols.issue_credential.v2_0.models.detail.indy.V20CredExRecordIndy
Retrieve a credential exchange indy detail record by its cred ex id.

```
class aries_cloudagent.protocols.issue_credential.v2_0.models.detail.indy.V20CredExRecordIndy
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

Credential exchange indy detail record detail schema.

```
class Meta
    Bases: object

    Credential exchange indy detail record schema metadata.

    model_class
        alias of V20CredExRecordIndy

    unknown = 'exclude'

    cred_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=No
    cred_ex_indy_id = <fields.String(default=<marshmallow.missing>, attribute=None, valida
    cred_request_metadata = <fields.Dict(default=<marshmallow.missing>, attribute=None, va
    cred_rev_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
    rev_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
```

Submodules

[aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record module](#)

Aries#0036 v1.0 credential exchange information with non-secrets storage.

```
class aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExReco
```


Bases: *aries_cloudagent.messaging.models.base_record.BaseExchangeRecord*

Represents an Aries#0036 credential exchange.

INITIATOR_EXTERNAL = 'external'

INITIATOR_SELF = 'self'

class Meta

Bases: *object*

CredentialExchange metadata.

schema_class = 'V20CredExRecordSchema'

RECORD_ID_NAME = 'cred_ex_id'

RECORD_TYPE = 'cred_ex_v20'

ROLE HOLDER = 'holder'

ROLE_ISSUER = 'issuer'

STATE_CREDENTIAL_RECEIVED = 'credential-received'

STATE_DONE = 'done'

STATE_ISSUED = 'credential-issued'

STATE_OFFER_RECEIVED = 'offer-received'

STATE_OFFER_SENT = 'offer-sent'

STATE_PROPOSAL_RECEIVED = 'proposal-received'

STATE_PROPOSAL_SENT = 'proposal-sent'

STATE_REQUEST_RECEIVED = 'request-received'

STATE_REQUEST_SENT = 'request-sent'

TAG_NAMES = {'thread_id'}

WEBHOOK_TOPIC = 'issue_credential_v2_0'

connection_id

Synonym for conn_id.

cred_ex_id

Accessor for the ID associated with this exchange.

cred_preview

Credential preview from credential proposal.

record_value

Accessor for the JSON record value generated for this credential exchange.

classmethod retrieve_by_conn_and_thread (*session: aries_cloudagent.core.profile.ProfileSession,*
conn_id: str, thread_id: str) →

aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record

Retrieve a credential exchange record by connection and thread ID.

class *aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord*

Bases: *aries_cloudagent.messaging.models.base_record.BaseExchangeSchema*

Schema to allow serialization/deserialization of credential exchange records.

```
class Meta
    Bases: object
    V20CredExSchema metadata.

    model_class
        alias of V20CredExRecord

    auto_issue = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
    auto_offer = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
    auto_remove = <fields.Boolean(default=True, attribute=None, validate=None, required=False)
    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_id_stored = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_issue = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_offer = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_preview = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_proposal = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_request = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
    cred_request_metadata = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
    error_msg = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    initiator = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    parent_thread_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(choices=
    state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(choices=
    thread_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
```

Submodules

aries_cloudagent.protocols.issue_credential.v2_0.manager module

V2.0 issue-credential protocol manager.

```
class aries_cloudagent.protocols.issue_credential.v2_0.manager.V20CredManager (profile:
                                                                    aries_cloudagent.c
```

Bases: object

Class for managing credentials.

```
create_offer (cred_ex_record: aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord,
               replacement_id: str = None, comment: str = None) → Tuple[
    aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord,
    aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.V20CredOffer]
Create credential offer, update credential exchange record.
```

Parameters

- **cred_ex_record** – credential exchange record for which to create offer

- **replacement_id** – identifier to help coordinate credential replacement
- **comment** – optional human-readable comment to set in offer message

Returns A tuple (credential exchange record, credential offer message)

create_proposal (*conn_id: str, *, auto_remove: bool = None, comment: str = None, cred_preview: aries_cloudagent.protocols.issue_credential.v2_0.messages.inner_cred_preview.V20CredPreview, fmt2filter: Mapping[aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_format.V20CredFormat, Mapping[str, str]], trace: bool = False*) → aries_cloudagent.protocols.issue_credential.v2_0.models_cred_ex_record.V20CredExRecord

Create a credential proposal.

Parameters

- **conn_id** – connection for which to create proposal
- **auto_remove** – whether to remove record automatically on completion
- **comment** – optional human-readable comment to include in proposal
- **cred_preview** – credential preview to use to create credential proposal
- **fmt2filter** – mapping between format and filter
- **trace** – whether to trace the operation

Returns Resulting credential exchange record including credential proposal

create_request (*cred_ex_record: aries_cloudagent.protocols.issue_credential.v2_0.models_cred_ex_record.V20CredExRecord, holder_id: str, comment: str = None*) → Tuple[aries_cloudagent.protocols.issue_credential.v2_0.models_cred_ex_record.V20CredExRecord, aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_request.V20CredRequest]

Create a credential request.

Parameters

- **cred_ex_record** – credential exchange record for which to create request
- **holder_id** – holder DID
- **comment** – optional human-readable comment to set in request message

Returns A tuple (credential exchange record, credential request message)

delete_cred_ex_record (*cred_ex_id: str*) → None

Delete credential exchange record and associated detail records.

get_detail_record (*cred_ex_id: str, fmt: aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_format.V20CredFormat*) → Union[aries_cloudagent.protocols.issue_credential.v2_0.models_detail_indy.V20CredExRecordIndy, aries_cloudagent.protocols.issue_credential.v2_0.models_detail_dif.V20CredExRecordDIF]

Retrieve credential exchange detail record by format.

issue_credential (*cred_ex_record: aries_cloudagent.protocols.issue_credential.v2_0.models_cred_ex_record.V20CredExRecord, *, comment: str = None, retries: int = 5*) → Tuple[aries_cloudagent.protocols.issue_credential.v2_0.models_cred_ex_record.V20CredExRecord, aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_issue.V20CredIssue]

Issue a credential.

Parameters

- **cred_ex_record** – credential exchange record for which to issue credential
- **comment** – optional human-readable comment pertaining to credential issue
- **retries** – maximum number of retries on failure

Returns (Updated credential exchange record, credential issue message)

Return type Tuple

prepare_send (*conn_id: str, cred_proposal: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal.V20CredProposal, auto_remove: bool = None*) → Tuple[aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord, aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.V20CredOffer]

Set up a new credential exchange record for an automated send.

Parameters

- **conn_id** – connection for which to create offer
- **cred_proposal** – credential proposal with preview
- **auto_remove** – flag to remove the record automatically on completion

Returns A tuple of the new credential exchange record and credential offer message

profile

Accessor for the current profile instance.

Returns The profile instance for this credential manager

receive_credential (*cred_issue_message: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue.V20CredIssue, conn_id: str*) → aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord

Receive a credential issue message from an issuer.

Hold cred in storage potentially to be processed by controller before storing.

Returns Credential exchange record, retrieved and updated

receive_credential_ack (*cred_ack_message: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ack.V20CredAck, conn_id: str*) → aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord

Receive credential ack from holder.

Parameters

- **cred_ack_message** – credential ack message to receive
- **conn_id** – connection identifier

Returns credential exchange record, retrieved and updated

receive_offer (*cred_offer_message: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.V20CredOffer, conn_id: str*) → aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord

Receive a credential offer.

Parameters

- **cred_offer_message** – credential offer message
- **conn_id** – connection identifier

Returns The credential exchange record, updated

receive_proposal (*cred_proposal_message: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal.V20CredProposal, conn_id: str*) → aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord

Receive a credential proposal.

Returns The resulting credential exchange record, created

receive_request (*cred_request_message: aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_request.V20CredRequest, conn_id: str*) → aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord

Receive a credential request.

Parameters

- **cred_request_message** – credential request to receive
- **conn_id** – connection identifier

Returns credential exchange record, retrieved and updated

store_credential (*cred_ex_record: aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord*, *cred_id: str = None*) → Tuple[aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord, aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ack.V20CredAck]
Store a credential in holder wallet; send ack to issuer.

Parameters

- **cred_ex_record** – credential exchange record with credential to store and ack
- **cred_id** – optional credential identifier to override default on storage

Returns (Updated credential exchange record, credential ack message)

Return type Tuple

exception aries_cloudagent.protocols.issue_credential.v2_0.manager.V20CredManagerError (*args, **kwargs)
er-
ror_
str
=
None
**kw

Bases: *aries_cloudagent.core.error.BaseError*

Credential manager error under issue-credential protocol v2.0.

aries_cloudagent.protocols.issue_credential.v2_0.message_types module

Message and inner object type identifiers for Connections.

aries_cloudagent.protocols.issue_credential.v2_0.routes module

Credential exchange admin routes.

class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredCreateSchema (*args, **kwargs)

Bases: *aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredSchemaCore*

Request schema for creating a credential from attr values.

credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.OneOf(['credential_preview', 'credential_ex_id'])>)>

class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredExIdMatchInfoSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base.OpenAPISchema*

Path parameters and validators for request taking credential exchange id.

cred_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.OneOf(['credential_preview', 'credential_ex_id'])>)>

class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredExRecordDetailSchema (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base.OpenAPISchema*

Credential exchange record and any per-format details.

```
cred_ex_record = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
dif = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
indy = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredExRecordListQueryStringSchema
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Parameters and validators for credential exchange record list query.
    connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
    role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(choices=('client', 'server'), error='Invalid role.'))>
    state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(choices=('active', 'expired', 'revoked'), error='Invalid state.'))>
    thread_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredExRecordListResultSchema
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Result schema for credential exchange record list query.
    results = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredFilter(*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Credential filtration criteria.
    dif = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
    indy = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
    validate_fields(data, **kwargs)
        Validate schema fields.
        Data must have indy, dif, or both.
        Parameters data – The data to validate
        Raises ValidationError – if data has neither indy nor dif
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredFilterDIF(*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    DIF credential filtration criteria.
    some_dif_criterion = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredFilterIndy(*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Indy credential filtration criteria.
    cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
    issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
    schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
    schema_issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=False)>
```

```

    schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
    schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredIdMatchInfoSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Path parameters and validators for request taking credential id.
    credential_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredIssueProblemReportRequestSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Request schema for sending problem report.
    explain_txt = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredIssueRequestSchema (*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.base.OpenAPISchema
    Request schema for sending credential issue admin message.
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredOfferRequestSchema (*args, **kwargs)
    Bases: aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredSchemaCore
    Request schema for sending credential offer admin message.
    auto_issue = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
    connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None)
    credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredProposalRequestPreviewSchema (*args, **kwargs)
    Bases: aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredProposalRequestSchemaBase
    Request schema for sending credential proposal on mandatory proposal preview.
    credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredProposalRequestPreviewSchema (*args, **kwargs)
    Bases: aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredProposalRequestSchemaBase
    Request schema for sending credential proposal on optional proposal preview.
    credential_preview = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredProposalRequestSchemaBase (*args, **kwargs)
    Bases: aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredSchemaCore
    Base class for request schema for sending credential proposal admin message.
    connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None)

```

```
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredStoreRequestSchema (*args, **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.OpenAPISchema*

Request schema for sending a credential store admin message.

```
credential_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredSchemaCore (*args, **kwargs)
```

Bases: *aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema*

Filter, auto-remove, comment, trace.

```
auto_remove = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
filter_ = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)>
```

```
class aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredentialModuleResponse
```

Bases: *aries_cloudagent.messaging.models.base.OpenAPISchema*

Response schema for v2.0 Issue Credential Module.

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_create (request: <sphinx.ext.autodoc.mock.Mock object at 0x7f760dc51b50>)
```

Request handler for creating a credential from attr values.

The internal credential record will be created without the credential being sent to any connection. This can be used in conjunction with the *oob* protocols to bind messages to an out of band message.

Parameters *request* – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_create_free_offer
```

Request handler for creating free credential offer.

Unlike with *send-offer*, this credential exchange is not tied to a specific connection. It must be dispatched out-of-band by the controller.

Parameters *request* – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_issue (request: <sphinx.ext.autodoc.mock.Mock object at 0x7f760dc51b50>)
```

Request handler for sending credential.

Parameters **request** – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_list(request:
    <sphinx.ext.autodoc.mock.Mock object at
    0x7f760dc518
```

Request handler for searching credential exchange records.

Parameters **request** – aiohttp request object

Returns The connection list response

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_problem_report
```

Request handler for sending problem report.

Parameters **request** – aiohttp request object

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_remove(request:
    <sphinx.ext.autodoc.mock.Mock object at
    0x7f760dc518
```

Request handler for removing a credential exchange record.

Parameters **request** – aiohttp request object

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_retrieve(request:
    <sphinx.ext.autodoc.mock.Mock object at
    0x7f760dc518
```

Request handler for fetching single credential exchange record.

Parameters **request** – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_send(request:
    <sphinx.ext.autodoc.mock.Mock object at
    0x7f760dc518
```

Request handler for sending credential from issuer to holder from attr values.

If both issuer and holder are configured for automatic responses, the operation ultimately results in credential issue; otherwise, the result waits on the first response not automated; the credential exchange record retains state regardless.

Parameters **request** – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_send_bound_offer`

Request handler for sending bound credential offer.

A holder initiates this sequence with a credential proposal; this message responds with an offer bound to the proposal.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_send_free_offer`

Request handler for sending free credential offer.

An issuer initiates a such a credential offer, free from any holder-initiated corresponding credential proposal with preview.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_send_proposal`

Request handler for sending credential proposal.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_send_request`

Request handler for sending credential request.

Parameters `request` – aiohttp request object

Returns The credential exchange record

`aries_cloudagent.protocols.issue_credential.v2_0.routes.credential_exchange_store`

(request: <sphinx.ext.ob-
ject
at
0x7f760dc51

Request handler for storing credential.

Parameters **request** – aiohttp request object

Returns The credential exchange record

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.post_process_routes (app:
    <sphinx.ext.autodoc.
    ob-
    ject
    at
    0x7f760dc51850>)
```

Amend swagger API.

```
aries_cloudagent.protocols.issue_credential.v2_0.routes.register (app:
    <sphinx.ext.autodoc.importer._MockC
    object    at
    0x7f760dc51850>)
```

Register routes.

Submodules

aries_cloudagent.protocols.issue_credential.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.out_of_band package

Subpackages

aries_cloudagent.protocols.out_of_band.v1_0 package

Subpackages

aries_cloudagent.protocols.out_of_band.v1_0.handlers package

Submodules

aries_cloudagent.protocols.out_of_band.v1_0.handlers.problem_report_handler module

OOB Problem Report Message Handler.

```
class aries_cloudagent.protocols.out_of_band.v1_0.handlers.problem_report_handler.OOBProblemReportMessageHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler class for OOB Problem Report Message.

Updates the ConnRecord Metadata state.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:
    aries_cloudagent.messaging.responder.BaseResponder)
    OOB Problem Report Message Handler.
```

Parameters

- **context** – Request context
- **responder** – Responder callback

`aries_cloudagent.protocols.out_of_band.v1_0.handlers.reuse_accept_handler` module

Handshake Reuse Accepted Message Handler under RFC 0434.

class `aries_cloudagent.protocols.out_of_band.v1_0.handlers.reuse_accept_handler.HandshakeRe`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Handler class for Handshake Reuse Accepted Message Handler under RFC 0434.

handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
Handle Handshake Reuse Accepted Message Handler under RFC 0434.

Parameters

- **context** – Request context
- **responder** – Responder callback

`aries_cloudagent.protocols.out_of_band.v1_0.handlers.reuse_handler` module

Handshake Reuse Message Handler under RFC 0434.

class `aries_cloudagent.protocols.out_of_band.v1_0.handlers.reuse_handler.HandshakeReuseMes`
Bases: `aries_cloudagent.messaging.base_handler.BaseHandler`

Handler class for Handshake Reuse Message Handler under RFC 0434.

handle (*context:* `aries_cloudagent.messaging.request_context.RequestContext`, *responder:* `aries_cloudagent.messaging.responder.BaseResponder`)
Handle Handshake Reuse Message Handler under RFC 0434.

Parameters

- **context** – Request context
- **responder** – Responder callback

`aries_cloudagent.protocols.out_of_band.v1_0.messages` package

Submodules

`aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation` module

An invitation content message.

class `aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.HSProto`
Bases: `enum.Enum`

Handshake protocol enum for invitation message.

RFC160 = `HSProtoSpec(rfc=160, name='connections/1.0', aka={'old', 'rfc160', 'connection`

RFC23 = `HSProtoSpec(rfc=23, name='didexchange/1.0', aka={'didexchange', 'new', 'didex'`

aka

Accessor for also-known-as.

get = <bound method HSPROTO.get of <enum 'HSPROTO'>>

rfc

Accessor for RFC.

```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.HSPROTOSpec(rfc,  
                                                                                   name,  
                                                                                   aka)
```

Bases: `tuple`

aka

Alias for field number 2

name

Alias for field number 1

rfc

Alias for field number 0

```

class aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationMessage(*,
com
men
str
=
Non
la-
bel:
str
=
Non
han
sha
Se-
que
=
Non
re-
que
Se-
que
=
Non
ser-
vice
Se-
que
str]
=
Non
ser-
vice
Se-
que
=
Non
ser-
vice
Se-
que
=
Non
ser-
vice
Se-
que
=
Non
**k

```

Bases: `aries_cloudagent.messaging.agent_message.AgentMessage`

Class representing an out of band invitation message.

```
class Meta
```

Bases: `object`

InvitationMessage metadata.

```
message_type = 'out-of-band/1.0/invitation'
```

```
schema_class = 'InvitationMessageSchema'
```

```
classmethod from_url(url: str) → aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationMessage
```

Parse a URL-encoded invitation into an *InvitationMessage* instance.

Parameters `url` – Url to decode

Returns An *InvitationMessage* object.

to_url (*base_url*: str = None) → str

Convert an invitation message to URL format for sharing.

Returns An invite url

classmethod `wrap_message` (*message*: dict) → aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator

Convert an aries message to an attachment decorator.

class aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.**InvitationMessageSchema**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

InvitationMessage schema.

class **Meta**

Bases: *object*

InvitationMessage schema metadata.

model_class

alias of *InvitationMessage*

unknown = 'exclude'

handshake_protocols = <fields.List (default=<marshmallow.missing>, attribute=None, validate=None, r

label = <fields.String (default=<marshmallow.missing>, attribute=None, validate=None, r

post_dump (*data*, ***kwargs*)

Post dump hook.

pre_load (*data*, ***kwargs*)

Pre load hook.

request_attach = <fields.Nested (default=<marshmallow.missing>, attribute=None, validate=None, r

service_blocks = <fields.Nested (default=<marshmallow.missing>, attribute=None, validate=None, r

service_dids = <fields.List (default=<marshmallow.missing>, attribute=None, validate=None, r

validate_fields (*data*, ***kwargs*)

Validate schema fields.

Parameters `data` – The data to validate

Raises *ValidationError* – If any of the fields do not validate

aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report module

Represents an OOB connection reuse problem report message.

```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReport (*,
    problem_code: str,
    explain: str,
    reason_codes: List[str],
    **kwargs)

Bases: aries_cloudagent.messaging.agent_message.AgentMessage
Base class representing an OOB connection reuse problem report message.

class Meta
    Bases: object
    OOB connection reuse problem report metadata.
    handler_class = 'aries_cloudagent.protocols.out_of_band.v1_0.handlers.problem_report'
    message_type = 'out-of-band/1.0/problem_report'
    schema_class = 'ProblemReportSchema'

class aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportReasonCode
    Bases: str, enum.Enum
    Supported reason codes.
    EXISTING_CONNECTION_DOES_NOT_EXISTS = 'existing_connection_does_not_exists'
    EXISTING_CONNECTION_NOT_ACTIVE = 'existing_connection_not_active'

class aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportSchema
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Schema for ProblemReport base class.

    class Meta
        Bases: object
        Metadata for problem report schema.
        model_class
            alias of ProblemReport
        unknown = 'exclude'

    check_thread_deco (obj, **kwargs)
        Thread decorator, and its thid and pthid, are mandatory.

    explain = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
    problem_code = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_missing=None)>
```

aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse module

Represents a Handshake Reuse message under RFC 0434.


```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse.HandshakeReuse (**kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
```

Class representing a Handshake Reuse message.

```
class Meta
```

Bases: *object*

Metadata for Handshake Reuse message.

```
    handler_class = 'aries_cloudagent.protocols.out_of_band.v1_0.handlers.reuse_handler'
```

```
    message_type = 'out-of-band/1.0/handshake-reuse'
```

```
    schema_class = 'HandshakeReuseSchema'
```

```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse.HandshakeReuseSchema (*args,
**kwargs)
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Handshake Reuse schema class.

```
class Meta
```

Bases: *object*

Handshake Reuse schema metadata.

```
    model_class
```

alias of *HandshakeReuse*

```
    unknown = 'exclude'
```

```
    check_thread_deco (obj, **kwargs)
```

Thread decorator, and its thid and pthid, are mandatory.

aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse_accept module

Represents a Handshake Reuse Accept message under RFC 0434.

```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse_accept.HandshakeReuseAccept
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
```

Class representing a Handshake Reuse Accept message.

```
class Meta
```

Bases: *object*

Metadata for Handshake Reuse Accept message.

```
    handler_class = 'aries_cloudagent.protocols.out_of_band.v1_0.handlers.reuse_accept'
```

```
    message_type = 'out-of-band/1.0/handshake-reuse-accepted'
```

```
    schema_class = 'HandshakeReuseAcceptSchema'
```

```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse_accept.HandshakeReuseAcceptSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Handshake Reuse Accept schema class.

```
class Meta
```

Bases: *object*

Handshake Reuse Accept schema metadata.

```
model_class
    alias of HandshakeReuseAccept

unknown = 'exclude'

check_thread_deco (obj, **kwargs)
    Thread decorator, and its thid and pthid, are mandatory.
```

aries_cloudagent.protocols.out_of_band.v1_0.messages.service module

Record used to represent a service block of an out of band invitation.

```
class aries_cloudagent.protocols.out_of_band.v1_0.messages.service.Service (*,
    _id:
        str
    =
        None,
    _type:
        str
    =
        None,
    did:
        str
    =
        None,
    re-
    cip-
    i-
    ent_keys:
        Se-
        quence[str]
    =
        None,
    rout-
    ing_keys:
        Se-
        quence[str]
    =
        None,
    ser-
    vice_endpoint:
        str
    =
        None)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Record used to represent a service block of an out of band invitation.

```
class Meta
    Bases: object

    Service metadata.

    schema_class = 'ServiceSchema'

class aries_cloudagent.protocols.out_of_band.v1_0.messages.service.ServiceSchema (*args,
    **kwargs)
```

Bases: `aries_cloudagent.messaging.models.base.BaseModelSchema`

Service schema.

class Meta

Bases: `object`

ServiceSchema metadata.

model_class

alias of `Service`

unknown = 'exclude'

did = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl`

post_dump (`data, **kwargs`)

Post dump hook.

recipient_keys = `<fields.List(default=<marshmallow.missing>, attribute=None, validate=`

routing_keys = `<fields.List(default=<marshmallow.missing>, attribute=None, validate=No`

service_endpoint = `<fields.String(default=<marshmallow.missing>, attribute=None, valid`

`aries_cloudagent.protocols.out_of_band.v1_0.models` package

Submodules

`aries_cloudagent.protocols.out_of_band.v1_0.models.invitation` module

Record for out of band invitations.

```

class aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.InvitationRecord(*,
in-
vi-
ta-
tion_id:
str
=
None,
state:
str
=
None,
invi_msg:
str
=
None,
in-
vi-
ta-
tion:
dict
=
None,
in-
vi-
ta-
tion_url:
str
=
None,
pub-
lic_id:
str
=
None,
trace:
bool
=
False,
**kwargs)

```

Bases: `aries_cloudagent.messaging.models.base_record.BaseExchangeRecord`

Represents an out of band invitation record.

```

class Meta
    Bases: object

    InvitationRecord metadata.

    schema_class = 'InvitationRecordSchema'

    RECORD_ID_NAME = 'invitation_id'

    RECORD_TYPE = 'oob_invitation'

    STATE_AWAIT_RESPONSE = 'await_response'

    STATE_DONE = 'done'

```

```

STATE_INITIAL = 'initial'
TAG_NAMES = {'invi_msg_id'}
WEBHOOK_TOPIC = 'oob_invitation'

invitation_id
    Accessor for the ID associated with this exchange.

record_value
    Accessor for the JSON record value generated for this invitation.
class aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.InvitationRecordSchema

Bases: aries_cloudagent.messaging.models.base_record.BaseExchangeSchema

Schema to allow serialization/deserialization of invitation records.

class Meta
    Bases: object

    InvitationRecordSchema metadata.

    model_class
        alias of InvitationRecord

invi_msg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
invitation = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
invitation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
invitation_url = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

```

Submodules

aries_cloudagent.protocols.out_of_band.v1_0.manager module

Classes to manage connections.

```

class aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager (session:
    aries_cloudagent.core,

Bases: aries_cloudagent.connections.base_manager.BaseConnectionManager

Class for managing out of band messages.

check_reuse_msg_state (conn_rec: aries_cloudagent.connections.models.conn_record.ConnRecord)
    Check reuse message state from the ConnRecord Metadata.

    Parameters conn_rec – The required ConnRecord with updated metadata

    Returns:

create_handshake_reuse_message (invi_msg: aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationRecord,
    conn_record: aries_cloudagent.connections.models.conn_record.ConnRecord)
    → None
    Create and Send a Handshake Reuse message under RFC 0434.

    Parameters
        • invi_msg – OOB Invitation Message
        • service – Service block extracted from the OOB invitation

```

Returns:

Raises

- *OutOfBandManagerError* – If there is an issue creating or
- sending the OOB invitation

create_invitation (*my_label*: str = None, *my_endpoint*: str = None, *auto_accept*: bool = None, *public*: bool = False, *hs_protos*: Sequence[aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.HSProto] = None, *multi_use*: bool = False, *alias*: str = None, *attachments*: Sequence[Mapping[KT, VT_co]] = None, *meta-data*: dict = None, *mediation_id*: str = None) → aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.InvitationRecord

Generate new connection invitation.

This interaction represents an out-of-band communication channel. In the future and in practice, these sort of invitations will be received over any number of channels such as SMS, Email, QR Code, NFC, etc.

Parameters

- **my_label** – label for this connection
- **my_endpoint** – endpoint where other party can reach me
- **auto_accept** – auto-accept a corresponding connection request (None to use config)
- **public** – set to create an invitation from the public DID
- **hs_protos** – list of handshake protocols to include
- **multi_use** – set to True to create an invitation for multiple-use connection
- **alias** – optional alias to apply to connection for later use
- **attachments** – list of dicts in form of {"id": ..., "type": ... }

Returns Invitation record

find_existing_connection (*tag_filter*: dict, *post_filter*: dict) → Optional[aries_cloudagent.connections.models.conn_record.ConnRecord]

Find existing ConnRecord.

Parameters

- **tag_filter** – The filter dictionary to apply
- **post_filter** – Additional value filters to apply matching positively, with sequence values specifying alternatives to match (hit any)

Returns ConnRecord or None

receive_invitation (*invi_msg*: aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.InvitationMessage, *use_existing_connection*: bool = True, *auto_accept*: bool = None, *alias*: str = None, *mediation_id*: str = None) → dict

Receive an out of band invitation message.

receive_problem_report (*problem_report*: aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReport, *receipt*: aries_cloudagent.transport.inbound.receipt.MessageReceipt, *conn_record*: aries_cloudagent.connections.models.conn_record.ConnRecord) → None

Receive and process a ProblemReport message from the inviter to invitee.

Process a *ProblemReport* message by updating the ConnRecord metadata state to *not_accepted*.

Parameters

- **problem_report** – The *ProblemReport* to process
- **receipt** – The message receipt

Returns:

Raises

- *OutOfBandManagerError* – if there is an error in processing the
- HandshakeReuseAccept message

receive_reuse_accepted_message (*reuse_accepted_msg*: *aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse_accepted.MessageReuseAccepted*,
receipt: *aries_cloudagent.transport.inbound.receipt.MessageReceipt*,
conn_record: *aries_cloudagent.connections.models.conn_record.ConnRecord*)
→ None

Receive and process a HandshakeReuseAccept message under RFC 0434.

Process a *HandshakeReuseAccept* message by updating the ConnRecord metadata state to *accepted*.

Parameters

- **reuse_accepted_msg** – The *HandshakeReuseAccept* to process
- **receipt** – The message receipt

Returns:

Raises

- *OutOfBandManagerError* – if there is an error in processing the
- HandshakeReuseAccept message

receive_reuse_message (*reuse_msg*: *aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse.HandshakeReuse*,
receipt: *aries_cloudagent.transport.inbound.receipt.MessageReceipt*)
→ None

Receive and process a HandshakeReuse message under RFC 0434.

Process a *HandshakeReuse* message by looking up the connection records using the MessageReceipt sender DID.

Parameters

- **reuse_msg** – The *HandshakeReuse* to process
- **receipt** – The message receipt

Returns:

Raises

- *OutOfBandManagerError* – If the existing connection is not active
- or the connection does not exists

session

Accessor for the current profile session.

Returns The profile session for this connection manager

exception *aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManagerError* (*args,
error_code:
str
=
None,
**kwargs)

Bases: *aries_cloudagent.core.error.BaseError*

Out of band error.

exception `aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManagerNotImplemented`

Bases: *aries_cloudagent.core.error.BaseError*

Out of band error for unimplemented functionality.

aries_cloudagent.protocols.out_of_band.v1_0.message_types module

Message and inner object type identifiers for Out of Band messages.

aries_cloudagent.protocols.out_of_band.v1_0.routes module

Out-of-band handling admin routes.

class `aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationCreateQueryStringSchema`

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for create invitation request query string.

auto_accept = `<fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)`

multi_use = `<fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)`

class `aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationCreateRequestSchema` (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Invitation create request Schema.

class `AttachmentDefSchema` (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Attachment Schema.

alias = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)`

attachments = `<fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None)`

handshake_protocols = `<fields.List(default=<marshmallow.missing>, attribute=None, validate=None)`

mediation_id = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=None)`

metadata = `<fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)`

my_label = `<fields.String(default=<marshmallow.missing>, attribute=None, validate=None)`

use_public_did = `<fields.Boolean(default=False, attribute=None, validate=None, required=True)`

class `aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationReceiveQueryStringSchema`

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for receive invitation request query string.


```

alias = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
auto_accept = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=
mediation_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
use_existing_connection = <fields.Boolean(default=True, attribute=None, validate=None,
class aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationReceiveRequestSchema (*arg
**kw

Bases:      aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.
            InvitationMessageSchema

Invitation request schema.

service = <fields.Field(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.protocols.out_of_band.v1_0.routes.OutOfBandModuleResponseSchema (*arg
**kw

Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema

Response schema for Out of Band Module.

aries_cloudagent.protocols.out_of_band.v1_0.routes.invitation_create(request:
                                <sphinx.ext.autodoc.importer._
                                object
                                at
                                0x7f760e8b2290>)

Request handler for creating a new connection invitation.

Parameters request – aiohttp request object

Returns The out of band invitation details

aries_cloudagent.protocols.out_of_band.v1_0.routes.invitation_receive(request:
                                <sphinx.ext.autodoc.importer._
                                ob-
                                ject at
                                0x7f760e8b2290>)

Request handler for receiving a new connection invitation.

Parameters request – aiohttp request object

Returns The out of band invitation details

aries_cloudagent.protocols.out_of_band.v1_0.routes.post_process_routes(app:
                                <sphinx.ext.autodoc.importer._
                                ob-
                                ject
                                at
                                0x7f760e8b2290>)

Amend swagger API.

aries_cloudagent.protocols.out_of_band.v1_0.routes.register(app:
                                <sphinx.ext.autodoc.importer._MockObject
                                object at
                                0x7f760e8b2290>)

Register routes.

```

Submodules

`aries_cloudagent.protocols.out_of_band.definition` module

Version definitions for this protocol.

`aries_cloudagent.protocols.present_proof` package

Subpackages

`aries_cloudagent.protocols.present_proof.v1_0` package

Subpackages

`aries_cloudagent.protocols.present_proof.v1_0.handlers` package

Submodules

`aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack_handler` module

Presentation ack message handler.

```
class aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack_handler.Presen  
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for presentation acks.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:  
        aries_cloudagent.messaging.responder.BaseResponder)  
    Message handler logic for presentation acks.
```

Parameters

- **context** – request context
- **responder** – responder callback

`aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_handler` module

Presentation message handler.

```
class aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_handler.Presentat  
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for presentations.

```
handle (context: aries_cloudagent.messaging.request_context.RequestContext, responder:  
        aries_cloudagent.messaging.responder.BaseResponder)  
    Message handler logic for presentations.
```

Parameters

- **context** – request context
- **responder** – responder callback

aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_proposal_handler module

Presentation proposal message handler.

```
class aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_proposal_handler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for presentation proposals.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Message handler logic for presentation proposals.
```

Parameters

- **context** – proposal context
- **responder** – responder callback

aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_request_handler module

Presentation request message handler.

```
class aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_request_handler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Message handler class for Aries#0037 v1.0 presentation requests.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Message handler logic for Aries#0037 v1.0 presentation requests.
```

Parameters

- **context** – request context
- **responder** – responder callback

aries_cloudagent.protocols.present_proof.v1_0.messages package

Subpackages

aries_cloudagent.protocols.present_proof.v1_0.messages.inner package

Submodules

aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview module

A presentation preview inner object.

```
class aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.Pr
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing an attribute specification within a presentation preview.

```
class Meta
```

Bases: *object*

Attr spec metadata.

```
    schema_class = 'PresAttrSpecSchema'
```

```
class Posture
```

Bases: *enum.Enum*

Attribute posture: self-attested, revealed claim or unrevealed claim.

```
    REVEALED_CLAIM = 1
```

```
    SELF_ATTESTED = 0
```

```
    UNREVEALED_CLAIM = 2
```

```
    b64_decoded_value() → str
```

Value, base64-decoded if applicable.

```
    static list_plain(plain: dict, cred_def_id: str, referent: str = None)
```

Return a list of *PresAttrSpec* on input cred def id.

Parameters

- **plain** – dict mapping names to values
- **cred_def_id** – credential definition identifier to specify
- **referent** – single referent to use, omit for none

Returns List of *PresAttrSpec* on input cred def id with no MIME types

posture

self-attested, revealed claim, or unrevealed claim.

Type Attribute posture

satisfies (*pred_spec: aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.PresPredSpec*)
Whether current specified attribute satisfies input specified predicate.

class aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.**Pres**

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Attribute specification schema.

class Meta

Bases: *object*

Attribute specification schema metadata.

model_class

alias of *PresAttrSpec*

cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<

mime_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None

name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re

referent = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None

value = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r

class aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.**Pres**

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a predicate specification within a presentation preview.

class Meta

Bases: *object*

Pred spec metadata.

schema_class = 'PresPredSpecSchema'

class aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.**Pres**

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Predicate specification schema.

class Meta

Bases: *object*

Predicate specification schema metadata.

```
model_class
    alias of PresPredSpec

cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
predicate = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
threshold = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=No

class aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.Pr
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing presentation preview.

class Meta

Bases: *object*

Presentation preview metadata.

message_type = 'present-proof/1.0/presentation-preview'

schema_class = 'PresentationPreviewSchema'

has_attr_spec (*cred_def_id: str, name: str, value: str*) → bool

Return whether preview contains given attribute specification.

Parameters

- **cred_def_id** – credential definition identifier
- **name** – attribute name
- **value** – attribute value

Returns Whether preview contains matching attribute specification.

indy_proof_request (*name: str = None, version: str = None, nonce: str = None, ledger: aries_cloudagent.ledger.indy.IndySdkLedger = None, non_revoc_intervals: Mapping[str, aries_cloudagent.revocation.models.indy.NonRevocationInterval] = None*) → dict

Return indy proof request corresponding to presentation preview.

Typically the verifier turns the proof preview into a proof request.

Parameters

- **name** – for proof request
- **version** – version for proof request
- **nonce** – nonce for proof request
- **ledger** – ledger with credential definitions, to check for revocation support
- **non_revoc_intervals** – non-revocation interval to use per cred def id where applicable (default from and to the current time if applicable)

Returns Indy proof request dict.

class aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.**Pre**

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Presentation preview schema.

class Meta

Bases: *object*

Presentation preview schema metadata.

model_class

alias of *PresentationPreview*

attributes = *<fields.Nested(default=<marshmallow.missing>, attribute=None, validate=No*

predicates = *<fields.Nested(default=<marshmallow.missing>, attribute=None, validate=No*

Submodules

aries_cloudagent.protocols.present_proof.v1_0.messages.presentation module

A (proof) presentation content message.

class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation.**Presentation** (*_id:*

str
=
None
*,
com-
ment
str
=
None
pre-
sen-
ta-
tions,
Se-
quen
=
None
***kw*

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a (proof) presentation.

```
class Meta
```

```
    Bases: object
```

```
    Presentation metadata.
```

```
    handler_class = 'aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation'
```

```
    message_type = 'present-proof/1.0/presentation'
```

```
    schema_class = 'PresentationSchema'
```

```
indy_proof (index: int = 0)
```

```
    Retrieve and decode indy proof from attachment.
```

```
        Parameters index – ordinal in attachment list to decode and return (typically, list has length 1)
```

```
class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation.PresentationSchema
```

```
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
```

```
    (Proof) presentation schema.
```

```
class Meta
```

```
    Bases: object
```

```
    Presentation schema metadata.
```

```
    model_class
```

```
        alias of Presentation
```

```
    unknown = 'exclude'
```

```
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
    presentations_attach = <fields.Nested(default=<marshmallow.missing>, attribute=None, v
```

`aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack` module

Represents an explicit RFC 15 ack message, adopted into present-proof protocol.

```
class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack.PresentationAck
```

```
    Bases: aries_cloudagent.messaging.ack.message.Ack
```

```
    Base class representing an explicit ack message for present-proof protocol.
```

```
class Meta
```

```
    Bases: object
```

```
    PresentationAck metadata.
```

```
    handler_class = 'aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation'
```

```
    message_type = 'present-proof/1.0/ack'
```

```
    schema_class = 'PresentationAckSchema'
```



```
class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack.PresentationAck
```

Bases: *aries_cloudagent.messaging.ack.message.AckSchema*

Schema for PresentationAck class.

```
class Meta
```

Bases: *object*

PresentationAck schema metadata.

```
model_class
```

alias of *PresentationAck*

```
unknown = 'exclude'
```

aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal module

A presentation proposal content message.

```
class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposal
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a presentation proposal.

```
class Meta
```

Bases: *object*

PresentationProposal metadata.

```
handler_class = 'aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_proposal_handler'
```

```
message_type = 'present-proof/1.0/propose-presentation'
```

```
schema_class = 'PresentationProposalSchema'
```

```
class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposalSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Presentation proposal schema.

```
class Meta
```

Bases: *object*

Presentation proposal schema metadata.

model_class

alias of *PresentationProposal*

unknown = 'exclude'

comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

presentation_proposal = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None,

aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request module

A presentation request content message.

class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request.**PresentationRequest**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a presentation request.

class **Meta**

Bases: *object*

PresentationRequest metadata.

handler_class = 'aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_request'

message_type = 'present-proof/1.0/request-presentation'

schema_class = 'PresentationRequestSchema'

indy_proof_request (*index: int = 0*)

Retrieve and decode indy proof request from attachment.

Parameters **index** – ordinal in attachment list to decode and return (typically, list has length 1)

class aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request.**PresentationRequestSchema**

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Presentation request schema.

class **Meta**

Bases: *object*

Presentation request schema metadata.

model_class

alias of *PresentationRequest*

unknown = 'exclude'

comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

request_presentations_attach = <fields.Nested(default=<marshmallow.missing>, attribute=

aries_cloudagent.protocols.present_proof.v1_0.models package

Submodules

aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange module

Aries#0037 v1.0 presentation exchange information with non-secrets storage.

```
class aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10Present
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseExchangeRecord*

Represents an Aries#0037 v1.0 presentation exchange.

```
INITIATOR_EXTERNAL = 'external'
INITIATOR_SELF = 'self'

class Meta
    Bases: object
    V10PresentationExchange metadata.
    schema_class = 'V10PresentationExchangeSchema'
RECORD_ID_NAME = 'presentation_exchange_id'
RECORD_TYPE = 'presentation_exchange_v10'
ROLE_PROVER = 'prover'
ROLE_VERIFIER = 'verifier'
STATE_PRESENTATION_ACKED = 'presentation_acked'
STATE_PRESENTATION_RECEIVED = 'presentation_received'
STATE_PRESENTATION_SENT = 'presentation_sent'
STATE_PROPOSAL_RECEIVED = 'proposal_received'
STATE_PROPOSAL_SENT = 'proposal_sent'
STATE_REQUEST_RECEIVED = 'request_received'
STATE_REQUEST_SENT = 'request_sent'
STATE_VERIFIED = 'verified'
TAG_NAMES = {'thread_id'}
WEBHOOK_TOPIC = 'present_proof'

presentation_exchange_id
    Accessor for the ID associated with this exchange.

record_value
    Accessor for JSON record value generated for this presentation exchange.
```

```
class aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10Present
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseExchangeSchema*

Schema for de/serialization of v1.0 presentation exchange records.

```
class Meta
    Bases: object
    V10PresentationExchangeSchema metadata.
    model_class
        alias of V10PresentationExchange

auto_present = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=
connection_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
error_msg = <fields.String(default=<marshmallow.missing>, attribute=None, validate=Non
initiator = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<On
```

```

presentation = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
presentation_exchange_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)
presentation_proposal_dict = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
presentation_request = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
presentation_request_dict = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)
role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(choices=('client', 'server'), error_message='Invalid role'))>
state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_present=True)
thread_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, if_present=True)
verified = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(choices=('true', 'false'), error_message='Invalid verified status'))>

```

aries_cloudagent.protocols.present_proof.v1_0.util package

Submodules

aries_cloudagent.protocols.present_proof.v1_0.util.indy module

Utilities for dealing with indy conventions.

`aries_cloudagent.protocols.present_proof.v1_0.util.indy.indy_proof_req2non_revoc_intervals`

Return non-revocation intervals by requested item referent in proof request.

`aries_cloudagent.protocols.present_proof.v1_0.util.indy.indy_proof_req_preview2indy_request`

Build indy requested-credentials structure.

Given input proof request and presentation preview, use credentials in holder's wallet to build indy requested credentials structure for input to proof creation.

Parameters

- **indy_proof_req** – indy proof request
- **pres_preview** – preview from presentation proposal, if applicable
- **holder** – holder injected into current context

aries_cloudagent.protocols.present_proof.v1_0.util.predicate module

Utilities for dealing with predicates.

class `aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate`
 Bases: `enum.Enum`

Enum for predicate types that indy-sdk supports.

```
GE = Relation(fortran='GE', wql='$gte', math='>=', yes=<function Predicate.<lambda>>, no,
GT = Relation(fortran='GT', wql='$gt', math='>', yes=<function Predicate.<lambda>>, no,
LE = Relation(fortran='LE', wql='$lte', math='<=', yes=<function Predicate.<lambda>>, no,
LT = Relation(fortran='LT', wql='$lt', math='<', yes=<function Predicate.<lambda>>, no,
get = <function Predicate.get>
to_int = <function Predicate.to_int>
```

```
class aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Relation(fortran,
                                                                           wql,
                                                                           math,
                                                                           yes,
                                                                           no)
```

Bases: `tuple`

```
fortran
    Alias for field number 0
math
    Alias for field number 2
no
    Alias for field number 4
wql
    Alias for field number 1
yes
    Alias for field number 3
```

Submodules

aries_cloudagent.protocols.present_proof.v1_0.manager module

Classes to manage presentations.

```
class aries_cloudagent.protocols.present_proof.v1_0.manager.PresentationManager(profile:
                                                                           aries_cloudagent
```

Bases: `object`

Class for managing presentations.

```
create_bound_request(presentation_exchange_record: aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange_record,
                    name: str = None, version: str = None, nonce: str = None, comment: str
                    = None)
```

Create a presentation request bound to a proposal.

Parameters

- **presentation_exchange_record** – Presentation exchange record for which to create presentation request
- **name** – name to use in presentation request (None for default)
- **version** – version to use in presentation request (None for default)
- **nonce** – nonce to use in presentation request (None to generate)

- **comment** – Optional human-readable comment pertaining to request creation

Returns A tuple (updated presentation exchange record, presentation request message)

create_exchange_for_proposal (*connection_id: str, presentation_proposal_message: aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposal, auto_present: bool = None*)

Create a presentation exchange record for input presentation proposal.

Parameters

- **connection_id** – connection identifier
- **presentation_proposal_message** – presentation proposal to serialize to exchange record
- **auto_present** – whether to present proof upon receiving proof request (default to configuration setting)

Returns Presentation exchange record, created

create_exchange_for_request (*connection_id: str, presentation_request_message: aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request.PresentationRequest*)

Create a presentation exchange record for input presentation request.

Parameters

- **connection_id** – connection identifier
- **presentation_request_message** – presentation request to use in creating exchange record, extracting indy proof request and thread id

Returns Presentation exchange record, updated

create_presentation (*presentation_exchange_record: aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.PresentationExchangeRecord, requested_credentials: dict, comment: str = None*)

Create a presentation.

Parameters

- **presentation_exchange_record** – Record to update
- **requested_credentials** – Indy formatted requested_credentials
- **comment** – optional human-readable comment

Example *requested_credentials* format, mapping proof request referents (uuid) to wallet referents (cred id):

```
{
  "self_attested_attributes": {
    "j233ffbc-bd35-49b1-934f-51e083106f6d": "value"
  },
  "requested_attributes": {
    "6253ffbb-bd35-49b3-934f-46e083106f6c": {
      "cred_id": "5bfa40b7-062b-4ae0-a251-a86c87922c0e",
      "revealed": true
    }
  },
  "requested_predicates": {
    "bfc8a97d-60d3-4f21-b998-85eeabe5c8c0": {
      "cred_id": "5bfa40b7-062b-4ae0-a251-a86c87922c0e"
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    }
}

```

Returns A tuple (updated presentation exchange record, presentation message)

receive_presentation (*message: aries_cloudagent.protocols.present_proof.v1_0.messages.presentation.Presentation*,
connection_record: aries_cloudagent.connections.models.conn_record.ConnRecord)
Receive a presentation, from message in context on manager creation.

Returns presentation exchange record, retrieved and updated

receive_presentation_ack (*message: aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack.PresentationAck*,
connection_record: aries_cloudagent.connections.models.conn_record.ConnRecord)
Receive a presentation ack, from message in context on manager creation.

Returns presentation exchange record, retrieved and updated

receive_proposal (*message: aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposal*,
connection_record: aries_cloudagent.connections.models.conn_record.ConnRecord)
Receive a presentation proposal from message in context on manager creation.

Returns Presentation exchange record, created

receive_request (*presentation_exchange_record: aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange_record.PresentationExchangeRecord*)
Receive a presentation request.

Parameters **presentation_exchange_record** – presentation exchange record with request to receive

Returns The presentation_exchange_record, updated

send_presentation_ack (*presentation_exchange_record: aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange_record.PresentationExchangeRecord*)
Send acknowledgement of presentation receipt.

Parameters **presentation_exchange_record** – presentation exchange record with thread id

verify_presentation (*presentation_exchange_record: aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange_record.PresentationExchangeRecord*)
Verify a presentation.

Parameters **presentation_exchange_record** – presentation exchange record with presentation request and presentation to verify

Returns presentation record, updated

exception `aries_cloudagent.protocols.present_proof.v1_0.manager.PresentationManagerError` (*a
er
ro
sta
=
Ne
**:*

Bases: `aries_cloudagent.core.error.BaseError`

Presentation error.

aries_cloudagent.protocols.present_proof.v1_0.message_types module

Message and inner object type identifiers for Connections.

aries_cloudagent.protocols.present_proof.v1_0.routes module

Admin routes for presentations.

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.CredentialsFetchQueryStringSchema
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Parameters and validators for credentials fetch request query string.

```
count = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
extra_query = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
referent = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
start = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyCredInfoSchema(*args,
                                                                              **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Schema for indy cred-info.

```
attrs = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, req
referent = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyCredPrecisSchema(*args,
                                                                              **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Schema for precis that indy credential search returns (and aca-py augments).

```
cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
cred_info = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None
cred_rev = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ari
interval = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None
presentation_referents = <fields.List(default=<marshmallow.missing>, attribute=None, v
rev_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqAttrSpecSchema(*args,
                                                                              **kwargs)
```

Bases: *aries_cloudagent.messaging.models.openapi.OpenAPISchema*

Schema for attribute specification in indy proof request.

```
name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
names = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, req
non_revoked = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=N
restrictions = <fields.List(default=<marshmallow.missing>, attribute=None, validate=No
validate_fields(data, **kwargs)
```

Validate schema fields.

Data must have exactly one of name or names; if names then restrictions are mandatory.

Parameters data – The data to validate

Raises `ValidationError` – if data has both or neither of name and names

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqNonRevokedSchema (*args, **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Non-revocation times specification in indy proof request.

```
to = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqPredSpecRestrictions
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Schema for restrictions in attr or pred specifier indy proof request.

```
cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
```

```
issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
```

```
schema_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ar
```

```
schema_issuer_id = <fields.String(default=<marshmallow.missing>, attribute=None, vali
```

```
schema_name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=N
```

```
schema_version = <fields.String(default=<marshmallow.missing>, attribute=None, validat
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqPredSpecSchema (*args, **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Schema for predicate specification in indy proof request.

```
name = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, re
```

```
non_revoked = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=N
```

```
p_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries
```

```
p_value = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None
```

```
restrictions = <fields.List(default=<marshmallow.missing>, attribute=None, validate=No
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRequestSchema (*args, **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Schema for indy proof request.

```
name = <fields.String(default='Proof request', attribute=None, validate=None, required
```

```
non_revoked = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=N
```

```
nonce = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
```

```
requested_attributes = <fields.Dict(default=<marshmallow.missing>, attribute=None, val
```

```
requested_predicates = <fields.Dict(default=<marshmallow.missing>, attribute=None, val
```

```
version = <fields.String(default='1.0', attribute=None, validate=<aries_cloudagent.mes
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyRequestedCredsRequestedAttr
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Schema for requested attributes within indy requested credentials structure.

```
cred_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
revealed = <fields.Boolean(default=True, attribute=None, validate=None, required=False>
class aries_cloudagent.protocols.present_proof.v1_0.routes.IndyRequestedCredsRequestedPredicates
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Schema for requested predicates within indy requested credentials structure.
    cred_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    timestamp = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
class aries_cloudagent.protocols.present_proof.v1_0.routes.PresExIdMatchInfoSchema(*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Path parameters and validators for request taking presentation exchange id.
    pres_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
class aries_cloudagent.protocols.present_proof.v1_0.routes.PresentProofModuleResponseSchema
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Response schema for Present Proof Module.
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationCreateRequestRequestSchema
    Bases: aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema
    Request schema for creating a proof request free of any connection.
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    proof_request = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationExchangeListQuerySchema
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Parameters and validators for presentation exchange list query.
    connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    role = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
    thread_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationExchangeListResultSchema
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Result schema for an Aries RFC 37 v1.0 presentation exchange query.
    results = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationProblemReportRequestSchema
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for sending problem report.
    explain_txt = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)>
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationProposalRequestSchema
```

Bases: *aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema*

Request schema for sending a presentation proposal admin message.

```
auto_present = <fields.Boolean(default=False, attribute=None, validate=None, required=True)
comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
presentation_proposal = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationRequestSchema(*args, **kwargs)
```

Bases: *aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema*

Request schema for sending a presentation.

```
requested_attributes = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
requested_predicates = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
self_attested_attributes = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
trace = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
```

```
class aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationSendRequestRequestSchema
```

Bases: *aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationCreateRequestRequestSchema*

Request schema for sending a proof request on a connection.

```
connection_id = <fields.UUID(default=<marshmallow.missing>, attribute=None, validate=None, required=True)
```

```
aries_cloudagent.protocols.present_proof.v1_0.routes.post_process_routes(app:
    <sphinx.ext.autodoc.importer.Object at 0x7f760dd29390>)

    Amend swagger API.
```

```
aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_create_request(request)
```

Request handler for creating a free presentation request.

The presentation request will not be bound to any proposal or existing connection.

Parameters **request** – aiohttp request object

Returns The presentation exchange details

`aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_credentials_list`

Request handler for searching applicable credential records.

Parameters `request` – aiohttp request object

Returns The credential list response

`aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_list` (*request: <sphinx.ext.autodoc.Document object at 0x7f760dd29390*

Request handler for searching presentation exchange records.

Parameters `request` – aiohttp request object

Returns The presentation exchange list response

`aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_problem_report` (*request: <sphinx.ext.autodoc.Document object at 0x7f760dd29390*

Request handler for sending problem report.

Parameters `request` – aiohttp request object

`aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_remove` (*request: <sphinx.ext.autodoc.Document object at 0x7f760dd29390*

Request handler for removing a presentation exchange record.

Parameters `request` – aiohttp request object

`aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_retrieve` (*request: <sphinx.ext.autodoc.Document object at 0x7f760dd29390*

Request handler for fetching a single presentation exchange record.

Parameters `request` – aiohttp request object

Returns The presentation exchange record response

`aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_send_bound_request`

Request handler for sending a presentation request bound to a proposal.

Parameters **request** – aiohttp request object

Returns The presentation exchange details

```
aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_send_free_request
```

Request handler for sending a presentation request free from any proposal.

Parameters **request** – aiohttp request object

Returns The presentation exchange details

```
aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_send_presentation
```

Request handler for sending a presentation.

Parameters **request** – aiohttp request object

Returns The presentation exchange details

```
aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_send_proposal (re
```

Request handler for sending a presentation proposal.

Parameters **request** – aiohttp request object

Returns The presentation exchange details

```
aries_cloudagent.protocols.present_proof.v1_0.routes.presentation_exchange_verify_presentation
```

Request handler for verifying a presentation request.

Parameters **request** – aiohttp request object

Returns The presentation exchange details

```
aries_cloudagent.protocols.present_proof.v1_0.routes.register (app:
    <sphinx.ext.autodoc.importer._MockObject object at
    0x7f760dd29390>)
```

Register routes.

Submodules

`aries_cloudagent.protocols.present_proof.definition` module

Version definitions for this protocol.

`aries_cloudagent.protocols.problem_report` package

Subpackages

`aries_cloudagent.protocols.problem_report.v1_0` package

```
aries_cloudagent.protocols.problem_report.v1_0.internal_error(err:
    aries_cloudagent.core.error.BaseError,
    http_error_class,
    record:
    Union[aries_cloudagent.connections.models.base_
    out-
    bound_handler:
    Coroutine[T_co,
    T_contra, V_co])
```

Send problem report and raise corresponding HTTP error.

Submodules

`aries_cloudagent.protocols.problem_report.v1_0.handler` module

Generic problem report handler.

```
class aries_cloudagent.protocols.problem_report.v1_0.handler.ProblemReportHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Problem report handler class.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
    aries_cloudagent.messaging.responder.BaseResponder)
    Handle problem report message.
```

Parameters

- **context** – Request context
- **responder** – Responder used to reply

`aries_cloudagent.protocols.problem_report.v1_0.message` module

Represents a generic problem report message.


```
class aries_cloudagent.protocols.problem_report.v1_0.message.ProblemReport (*,
    msg_catalog:
    str
    =
    None,
    lo-
    cale:
    str
    =
    None,
    ex-
    plain_txt:
    str
    =
    None,
    ex-
    plain_html:
    Mapping[str,
    str]
    =
    None,
    prob-
    lem_items:
    Sequence[Mapping[str,
    str]]
    =
    None,
    who_retries:
    str
    =
    None,
    fix_hint_txt:
    Mapping[str,
    str]
    =
    None,
    im-
    pact:
    str
    =
    None,
    where:
    str
    =
    None,
    time_noticed:
    str
    =
    None,
    track-
    ing_uri:
    str
    =
    None,
    es-
    ca-
    la-
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Base class representing a generic problem report message.

class Meta

Bases: *object*

Problem report metadata.

handler_class = 'aries_cloudagent.protocols.problem_report.v1_0.handler.ProblemReportHandler'

message_type = 'notification/1.0/problem-report'

schema_class = 'ProblemReportSchema'

class *aries_cloudagent.protocols.problem_report.v1_0.message.ProblemReportSchema* (*args,
**kwargs)

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

Schema for ProblemReport base class.

class Meta

Bases: *object*

Problem report schema metadata.

model_class

alias of *ProblemReport*

unknown = 'exclude'

escalation_uri = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

explain_l10n = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)>

explain_ltxt = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

fix_hint_ltxt = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None)>

impact = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf>)>

locale = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

msg_catalog = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

problem_items = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None)>

time_noticed = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

tracking_uri = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

where = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<Regexp>)>

who_retries = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None)>

aries_cloudagent.protocols.problem_report.v1_0.message_types module

Message type identifiers for problem reports.

Submodules

aries_cloudagent.protocols.problem_report.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.routing package

Subpackages

aries_cloudagent.protocols.routing.v1_0 package

Subpackages

aries_cloudagent.protocols.routing.v1_0.handlers package

Submodules

aries_cloudagent.protocols.routing.v1_0.handlers.forward_handler module

Handler for incoming forward messages.

```
class aries_cloudagent.protocols.routing.v1_0.handlers.forward_handler.ForwardHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for incoming forward messages.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Message handler implementation.
```

aries_cloudagent.protocols.routing.v1_0.handlers.route_query_request_handler module

Handler for incoming route-query-request messages.

```
class aries_cloudagent.protocols.routing.v1_0.handlers.route_query_request_handler.RouteQueryRequestHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for incoming route-query-request messages.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Message handler implementation.
```

aries_cloudagent.protocols.routing.v1_0.handlers.route_query_response_handler module

Handler for incoming route-query-response messages.

```
class aries_cloudagent.protocols.routing.v1_0.handlers.route_query_response_handler.RouteQueryResponseHandler
    Bases: aries_cloudagent.messaging.base_handler.BaseHandler
```

Handler for incoming route-query-response messages.

```
handle (context:          aries_cloudagent.messaging.request_context.RequestContext,      responder:
        aries_cloudagent.messaging.responder.BaseResponder)
    Message handler implementation.
```

aries_cloudagent.protocols.routing.v1_0.handlers.route_update_request_handler module

Handler for incoming route-update-request messages.

class aries_cloudagent.protocols.routing.v1_0.handlers.route_update_request_handler.RouteUpdateRequestHandler

Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Handler for incoming route-update-request messages.

handle (context: *aries_cloudagent.messaging.request_context.RequestContext*, responder: *aries_cloudagent.messaging.responder.BaseResponder*)
Message handler implementation.

aries_cloudagent.protocols.routing.v1_0.handlers.route_update_response_handler module

Handler for incoming route-update-response messages.

class aries_cloudagent.protocols.routing.v1_0.handlers.route_update_response_handler.RouteUpdateResponseHandler

Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Handler for incoming route-update-response messages.

handle (context: *aries_cloudagent.messaging.request_context.RequestContext*, responder: *aries_cloudagent.messaging.responder.BaseResponder*)
Message handler implementation.

aries_cloudagent.protocols.routing.v1_0.messages package

Submodules

aries_cloudagent.protocols.routing.v1_0.messages.forward module

Represents a forward message.

class aries_cloudagent.protocols.routing.v1_0.messages.forward.Forward(*, to: *str*, msg: *Union[dict, str]*, **kwargs)
= None

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Represents a request to forward a message to a connected agent.

class Meta

Bases: *object*

Forward metadata.

handler_class = 'aries_cloudagent.protocols.routing.v1_0.handlers.forward_handler.ForwardHandler'

message_type = 'routing/1.0/forward'

```

    schema_class = 'ForwardSchema'

class aries_cloudagent.protocols.routing.v1_0.messages.forward.ForwardSchema(*args,
                                                                              **kwargs)
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    Forward message schema used in serialization/deserialization.

    class Meta
        Bases: object
        ForwardSchema metadata.

        model_class
            alias of Forward

        unknown = 'exclude'

    handle_str_message(data, **kwargs)
        Accept string value for msg, as produced by previous implementation.

    msg = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, requi.
    to = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, requ.

```

aries_cloudagent.protocols.routing.v1_0.messages.route_query_request module

Query existing forwarding routes.

```
class aries_cloudagent.protocols.routing.v1_0.messages.route_query_request.RouteQueryRequest
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Query existing routes from a routing agent.

```

class Meta
    Bases: object
    RouteQueryRequest metadata.

    handler_class = 'aries_cloudagent.protocols.routing.v1_0.handlers.route_query_reque
    message_type = 'routing/1.0/route-query-request'
    schema_class = 'RouteQueryRequestSchema'

class aries_cloudagent.protocols.routing.v1_0.messages.route_query_request.RouteQueryRequest
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
    RouteQueryRequest message schema used in serialization/deserialization.

```

```

class Meta
    Bases: object

    RouteQueryRequestSchema metadata.

    model_class
        alias of RouteQueryRequest

    unknown = 'exclude'

    filter = <fields.Dict (default=<marshmallow.missing>, attribute=None, validate=None, re
    paginate = <fields.Nested (default=<marshmallow.missing>, attribute=None, validate=None

```

aries_cloudagent.protocols.routing.v1_0.messages.route_query_response module

Return existing forwarding routes in response to a query.

```

class aries_cloudagent.protocols.routing.v1_0.messages.route_query_response.RouteQueryResp

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Return existing routes from a routing agent.

```

class Meta
    Bases: object

    RouteQueryResponse metadata.

    handler_class = 'aries_cloudagent.protocols.routing.v1_0.handlers.route_query_respo
    message_type = 'routing/1.0/route-query-response'
    schema_class = 'RouteQueryResponseSchema'

```

```

class aries_cloudagent.protocols.routing.v1_0.messages.route_query_response.RouteQueryResp

```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

RouteQueryResponse message schema used in serialization/deserialization.

```

class Meta
    Bases: object

    RouteQueryResponseSchema metadata.

    model_class
        alias of RouteQueryResponse

    unknown = 'exclude'

    paginated = <fields.Nested (default=<marshmallow.missing>, attribute=None, validate=Non

```

```
routes = <fields.List (default=<marshmallow.missing>, attribute=None, validate=None, re
```

`aries_cloudagent.protocols.routing.v1_0.messages.route_update_request` module

Request to update forwarding routes.

```
class aries_cloudagent.protocols.routing.v1_0.messages.route_update_request.RouteUpdateReq
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Request to existing routes with a routing agent.

```
class Meta
```

Bases: *object*

RouteUpdateRequest metadata.

```
handler_class = 'aries_cloudagent.protocols.routing.v1_0.handlers.route_update_req
```

```
message_type = 'routing/1.0/route-update-request'
```

```
schema_class = 'RouteUpdateRequestSchema'
```

```
class aries_cloudagent.protocols.routing.v1_0.messages.route_update_request.RouteUpdateReq
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

RouteUpdateRequest message schema used in serialization/deserialization.

```
class Meta
```

Bases: *object*

RouteUpdateRequestSchema metadata.

```
model_class
```

alias of *RouteUpdateRequest*

```
unknown
```

Used by autodoc_mock_imports.

```
updates
```

Used by autodoc_mock_imports.

`aries_cloudagent.protocols.routing.v1_0.messages.route_update_response` module

Response for a route update request.

```
class aries_cloudagent.protocols.routing.v1_0.messages.route_update_response.RouteUpdateResponse
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Response for a route update request.

```
class Meta
```

Bases: *object*

RouteUpdateResponse metadata.

```
    handler_class = 'aries_cloudagent.protocols.routing.v1_0.handlers.route_update_respon
```

```
    message_type = 'routing/1.0/route-update-response'
```

```
    schema_class = 'RouteUpdateResponseSchema'
```

```
class aries_cloudagent.protocols.routing.v1_0.messages.route_update_response.RouteUpdateResponseSchema
```

Bases: *aries_cloudagent.messaging.agent_message.AgentMessageSchema*

RouteUpdateResponse message schema used in serialization/deserialization.

```
class Meta
```

Bases: *object*

RouteUpdateResponseSchema metadata.

```
    model_class
```

alias of *RouteUpdateResponse*

```
    unknown = 'exclude'
```

```
    updated = <fields.List (default=<marshmallow.missing>, attribute=None, validate=None, r
```

aries_cloudagent.protocols.routing.v1_0.models package

Submodules

aries_cloudagent.protocols.routing.v1_0.models.paginate module

An object for containing the request pagination information.

```
class aries_cloudagent.protocols.routing.v1_0.models.paginate.Paginate(*,
                                                                    limit:
                                                                    int
                                                                    =
                                                                    None,
                                                                    off-
                                                                    set:
                                                                    int
                                                                    =
                                                                    None,
                                                                    **kwargs)
```


Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing the pagination details of a request.

class Meta

Bases: *object*

Paginate metadata.

schema_class = 'PaginateSchema'

class *aries_cloudagent.protocols.routing.v1_0.models.paginate.PaginateSchema* (*args,
**kwargs)

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Paginate schema.

class Meta

Bases: *object*

PaginateSchema metadata.

model_class

alias of *Paginate*

unknown = 'exclude'

limit = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None, .

offset = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None,

aries_cloudagent.protocols.routing.v1_0.models.paginated module

An object for containing the response pagination information.

class *aries_cloudagent.protocols.routing.v1_0.models.paginated.Paginated* (*,
start:
int
=
None,
end:
int
=
None,
limit:
int
=
None,
to-
tal:
int
=
None,
**kwargs)

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing the pagination details of a response.

class Meta

Bases: *object*

Paginated metadata.

schema_class = 'PaginatedSchema'

class aries_cloudagent.protocols.routing.v1_0.models.paginated.**PaginatedSchema** (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Paginated schema.

class Meta

Bases: *object*

PaginatedSchema metadata.

model_class

alias of *Paginated*

unknown = 'exclude'

end = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None, re

limit = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None,

start = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None,

total = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None,

aries_cloudagent.protocols.routing.v1_0.models.route_query_result module

An object for containing returned route information.

class aries_cloudagent.protocols.routing.v1_0.models.route_query_result.**RouteQueryResult** (*, re

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing route information returned by a route query.

class Meta

Bases: *object*

RouteQueryResult metadata.

schema_class = 'RouteQueryResultSchema'

class aries_cloudagent.protocols.routing.v1_0.models.route_query_result.**RouteQueryResultSch**

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

RouteQueryResult schema.

class Meta

Bases: *object*

RouteQueryResultSchema metadata.

```

    model_class
        alias of RouteQueryResult

    unknown = 'exclude'

    recipient_key = <fields.String(default=<marshmallow.missing>, attribute=None, validate=

```

aries_cloudagent.protocols.routing.v1_0.models.route_record module

An object for containing information on an individual route.

```

class aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord(*,
                                                                              record_id:
                                                                              str
                                                                              =
                                                                              None,
                                                                              role:
                                                                              str
                                                                              =
                                                                              None,
                                                                              con-
                                                                              nec-
                                                                              tion_id:
                                                                              str
                                                                              =
                                                                              None,
                                                                              wal-
                                                                              let_id:
                                                                              str
                                                                              =
                                                                              None,
                                                                              re-
                                                                              cip-
                                                                              i-
                                                                              ent_key:
                                                                              str
                                                                              =
                                                                              None,
                                                                              **kwargs)

```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Class representing stored route information.

```

class Meta
    Bases: object

    RouteRecord metadata.

    schema_class = 'RouteRecordSchema'

    RECORD_ID_NAME = 'record_id'

    RECORD_TYPE = 'forward_route'

    ROLE_CLIENT = 'client'

    ROLE_SERVER = 'server'

    TAG_NAMES = {'connection_id', 'recipient_key', 'role', 'wallet_id'}

```

record_id

Get record ID.

record_value

Accessor for JSON record value.

classmethod retrieve_by_connection_id (*session: aries_cloudagent.core.profile.ProfileSession,*
connection_id: str) →
aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord

Retrieve a route record by connection ID.

Parameters

- **session** (*ProfileSession*) – session
- **connection_id** (*str*) – ID to look up

Returns retrieved route record

Return type *RouteRecord*

classmethod retrieve_by_recipient_key (*session: aries_cloudagent.core.profile.ProfileSession,*
recipient_key: str) →
aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord

Retrieve a route record by recipient key.

Parameters

- **session** (*ProfileSession*) – session
- **recipient_key** (*str*) – key to look up

Returns retrieved route record

Return type *RouteRecord*

class *aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecordSchema* (**args,*
***kwargs*)

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

RouteRecord schema.

class Meta

Bases: *object*

RouteRecordSchema metadata.

model_class

alias of *RouteRecord*

unknown

Used by autodoc_mock_imports.

connection_id

Used by autodoc_mock_imports.

recipient_key

Used by autodoc_mock_imports.

record_id

Used by autodoc_mock_imports.

role

Used by autodoc_mock_imports.

validate_fields (*data*, ***kwargs*)

Validate schema fields.

Parameters *data* – The data to validate

Raises *ValidationError* – If any of the fields do not validate

wallet_id

Used by autodoc_mock_imports.

aries_cloudagent.protocols.routing.v1_0.models.route_update module

An object for containing route information to be updated.

```
class aries_cloudagent.protocols.routing.v1_0.models.route_update.RouteUpdate (*,
re-
cip-
i-
ent_key:
str
=
None,
ac-
tion:
str
=
None,
**kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a route update request.

ACTION_CREATE = 'create'

ACTION_DELETE = 'delete'

class Meta

Bases: *object*

RouteUpdate metadata.

schema_class = 'RouteUpdateSchema'

```
class aries_cloudagent.protocols.routing.v1_0.models.route_update.RouteUpdateSchema (*args,
**kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

RouteUpdate schema.

class Meta

Bases: *object*

RouteUpdateSchema metadata.

model_class

alias of *RouteUpdate*

unknown

Used by autodoc_mock_imports.

action

Used by autodoc_mock_imports.

recipient_key

Used by autodoc_mock_imports.

aries_cloudagent.protocols.routing.v1_0.models.route_updated module

An object for containing updated route information.

```
class aries_cloudagent.protocols.routing.v1_0.models.route_updated.RouteUpdated(*,
                                     re-
                                     cip-
                                     i-
                                     ent_key:
                                     str
                                     =
                                     None,
                                     ac-
                                     tion:
                                     str
                                     =
                                     None,
                                     re-
                                     sult:
                                     str
                                     =
                                     None,
                                     **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Class representing a route update response.

class Meta

Bases: *object*

RouteUpdated metadata.

schema_class = 'RouteUpdatedSchema'

RESULT_CLIENT_ERROR = 'client_error'

RESULT_NO_CHANGE = 'no_change'

RESULT_SERVER_ERROR = 'server_error'

RESULT_SUCCESS = 'success'

```
class aries_cloudagent.protocols.routing.v1_0.models.route_updated.RouteUpdatedSchema(*args,
                                                                                       **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

RouteUpdated schema.

class Meta

Bases: *object*

RouteUpdatedSchema metadata.

model_class

alias of *RouteUpdated*

unknown

Used by autodoc_mock_imports.

action

Used by autodoc_mock_imports.

recipient_key

Used by autodoc_mock_imports.

result

Used by autodoc_mock_imports.

Submodules**aries_cloudagent.protocols.routing.v1_0.manager module**

Routing manager classes for tracking and inspecting routing records.

exception `aries_cloudagent.protocols.routing.v1_0.manager.RouteNotFoundError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.protocols.routing.v1_0.manager.RoutingManagerError`

Requested route was not found.

class `aries_cloudagent.protocols.routing.v1_0.manager.RoutingManager` (profile: `aries_cloudagent.core.profile.Profile`)

Bases: `object`

Class for handling routing records.

RECORD_TYPE = 'forward_route'

create_route_record (client_connection_id: str = None, recipient_key: str = None, internal_wallet_id: str = None) → `aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord`

Create and store a new RouteRecord.

Parameters

- **client_connection_id** – The ID of the connection record
- **recipient_key** – The recipient verkey of the route
- **internal_wallet_id** – The ID of the wallet record. Used for internal routing

Returns The new routing record

delete_route_record (route: `aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord`)

Remove an existing route record.

get_recipient (recip_verkey: str) → `aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord`

Resolve the recipient for a verkey.

Parameters **recip_verkey** – The verkey (“to”) of the incoming Forward message**Returns** The `RouteRecord` associated with this verkey

get_routes (client_connection_id: str = None, tag_filter: dict = None) → Sequence[`aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord`]

Fetch all routes associated with the current connection.

Parameters

- **client_connection_id** – The ID of the connection record
- **tag_filter** – An optional dictionary of tag filters

Returns A sequence of route records found by the query

send_create_route (*router_connection_id: str, recip_key: str, outbound_handler: Coroutine[T_co, T_contra, V_co]*)

Create and send a route update request.

Returns: the current routing state (request or done)

update_routes (*client_connection_id: str, updates: Sequence[aries_cloudagent.protocols.routing.v1_0.models.route_update.RouteUpdate]* → *Sequence[aries_cloudagent.protocols.routing.v1_0.models.route_updated.RouteUpdated]*)

Update routes associated with the current connection.

Parameters

- **client_connection_id** – The ID of the connection record
- **updates** – The sequence of route updates (create/delete) to perform.

exception `aries_cloudagent.protocols.routing.v1_0.manager.RoutingManagerError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Generic routing error.

aries_cloudagent.protocols.routing.v1_0.message_types module

Message type identifiers for Routing.

Submodules

aries_cloudagent.protocols.routing.definition module

Version definitions for this protocol.

aries_cloudagent.protocols.trustping package

Subpackages

aries_cloudagent.protocols.trustping.v1_0 package

Subpackages

aries_cloudagent.protocols.trustping.v1_0.handlers package

Submodules

aries_cloudagent.protocols.trustping.v1_0.handlers.ping_handler module

Ping handler.

class aries_cloudagent.protocols.trustping.v1_0.handlers.ping_handler.**PingHandler**
Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Ping handler class.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle ping message.

Parameters

- **context** – Request context
- **responder** – Responder used to reply

aries_cloudagent.protocols.trustping.v1_0.handlers.ping_response_handler module

Ping response handler.

class aries_cloudagent.protocols.trustping.v1_0.handlers.ping_response_handler.**PingResponseHandler**
Bases: *aries_cloudagent.messaging.base_handler.BaseHandler*

Ping response handler class.

handle (*context:* *aries_cloudagent.messaging.request_context.RequestContext*, *responder:* *aries_cloudagent.messaging.responder.BaseResponder*)
Handle ping response message.

Parameters

- **context** – Request context
- **responder** – Responder used to reply

aries_cloudagent.protocols.trustping.v1_0.messages package

Submodules

aries_cloudagent.protocols.trustping.v1_0.messages.ping module

Represents a trust ping message.

class aries_cloudagent.protocols.trustping.v1_0.messages.ping.**Ping**(*, *response_requested:* *bool* = *True*, *comment:* *str* = *None*, ***kwargs*)
Bases: *aries_cloudagent.messaging.agent_message.AgentMessage*

Class representing a trustping message.

```
class Meta
```

```
    Bases: object
```

```
    Ping metadata.
```

```
    handler_class = 'aries_cloudagent.protocols.trustping.v1_0.handlers.ping_handler.Pi
```

```
    message_type = 'trust_ping/1.0/ping'
```

```
    schema_class = 'PingSchema'
```

```
class aries_cloudagent.protocols.trustping.v1_0.messages.ping.PingSchema(*args,  
                                                                           **kwargs)
```

```
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
```

```
    Schema for Ping class.
```

```
class Meta
```

```
    Bases: object
```

```
    PingSchema metadata.
```

```
    model_class
```

```
        alias of Ping
```

```
    unknown = 'exclude'
```

```
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
```

```
    response_requested = <fields.Boolean(default=True, attribute=None, validate=None, requ
```

aries_cloudagent.protocols.trustping.v1_0.messages.ping_response module

Represents an response to a trust ping message.

```
class aries_cloudagent.protocols.trustping.v1_0.messages.ping_response.PingResponse(*,  
                                                                                       com-  
                                                                                       ment:  
                                                                                       str  
                                                                                       =  
                                                                                       None,  
                                                                                       **kwargs)
```

```
    Bases: aries_cloudagent.messaging.agent_message.AgentMessage
```

```
    Class representing a ping response.
```

```
class Meta
```

```
    Bases: object
```

```
    PingResponse metadata.
```

```
    handler_class = 'aries_cloudagent.protocols.trustping.v1_0.handlers.ping_response_h
```

```
    message_type = 'trust_ping/1.0/ping_response'
```

```
    schema_class = 'PingResponseSchema'
```

```
class aries_cloudagent.protocols.trustping.v1_0.messages.ping_response.PingResponseSchema(  
    )
```

```
    Bases: aries_cloudagent.messaging.agent_message.AgentMessageSchema
```

```
    PingResponse schema.
```

```

class Meta
    Bases: object

    PingResponseSchema metadata.

    model_class
        alias of PingResponse

    unknown = 'exclude'

    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,

```

Submodules

aries_cloudagent.protocols.trustping.v1_0.message_types module

Message type identifiers for Trust Pings.

aries_cloudagent.protocols.trustping.v1_0.routes module

Trust ping admin routes.

```

class aries_cloudagent.protocols.trustping.v1_0.routes.ConnIdMatchInfoSchema (*args,
                                                                              **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Path parameters and validators for request taking connection id.
    conn_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.protocols.trustping.v1_0.routes.PingRequestResponseSchema (*args,
                                                                              **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for performing a ping.
    thread_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.protocols.trustping.v1_0.routes.PingRequestSchema (*args,
                                                                              **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for performing a ping.
    comment = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
aries_cloudagent.protocols.trustping.v1_0.routes.connections_send_ping (request:
                                                                              <sphinx.ext.autodoc.importe
                                                                              ob-
                                                                              ject
                                                                              at
                                                                              0x7f760de3ed10>)

Request handler for sending a trust ping to a connection.

Parameters request – aiohttp request object

```

```
aries_cloudagent.protocols.trustping.v1_0.routes.post_process_routes (app:
                                                                    <sphinx.ext.autodoc.importer._
                                                                    object
                                                                    at
                                                                    0x7f760de3ed10>)
```

Amend swagger API.

```
aries_cloudagent.protocols.trustping.v1_0.routes.register (app:
                                                            <sphinx.ext.autodoc.importer._MockObject
                                                            object                    at
                                                            0x7f760de3ed10>)
```

Register routes.

Submodules

aries_cloudagent.protocols.trustping.definition module

Version definitions for this protocol.

Submodules

aries_cloudagent.protocols.didcomm_prefix module

DIDComm prefix management.

```
class aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix
    Bases: enum.Enum
    Enum for DIDComm Prefix, old or new style, per Aries RFC 384.
    NEW = 'https://didcomm.org'
    OLD = 'did:sov:BzCbsNYhMrjHiqZDTUASHg;spec'
    qualify (msg_type: str = None) → str
        Qualify input message type with prefix and separator.
    qualify_all = <bound method DIDCommPrefix.qualify_all of <enum 'DIDCommPrefix'>>
    qualify_current = <function DIDCommPrefix.qualify_current>
    set = <function DIDCommPrefix.set>
    unqualify = <function DIDCommPrefix.unqualify>
aries_cloudagent.protocols.didcomm_prefix.qualify (msg_type: str, prefix: str)
    Qualify a message type with a prefix, if unqualified.
```

aries_cloudagent.revocation package

Subpackages

aries_cloudagent.revocation.models package

Submodules

aries_cloudagent.revocation.models.indy module

Indy utilities for revocation.

```
class aries_cloudagent.revocation.models.indy.NonRevocationInterval (fro: int
                                                                    = None,
                                                                    to: int
                                                                    = None,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModel*

Indy non-revocation interval.

class Meta

Bases: *object*

NonRevocationInterval metadata.

schema_class = 'NonRevocationIntervalSchema'

covers (*timestamp: int = None*) → bool

Whether input timestamp (default now) lies within non-revocation interval.

Parameters **timestamp** – time of interest

Returns whether input time satisfies non-revocation interval

timestamp () → bool

Return a timestamp that the non-revocation interval covers.

```
class aries_cloudagent.revocation.models.indy.NonRevocationIntervalSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base.BaseModelSchema*

Schema to allow serialization/deserialization of non-revocation intervals.

class Meta

Bases: *object*

NonRevocationIntervalSchema metadata.

model_class

alias of *NonRevocationInterval*

fro = <fields.Integer (default=<marshmallow.missing>, attribute=None, validate=<aries_c

to = <fields.Integer (default=<marshmallow.missing>, attribute=None, validate=<aries_cl

aries_cloudagent.revocation.models.issuer_cred_rev_record module

Issuer credential revocation information.

```
class aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord(*,
                                                                                   record_id:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   state:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   cred_ex_id:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   rev_reg_id:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   cred_rev_id:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   cred_def_id:
                                                                                   str
                                                                                   =
                                                                                   None,
                                                                                   **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Represents credential revocation information to retain post-issue.

class Meta

Bases: *object*

IssuerCredRevRecord metadata.

schema_class = 'IssuerCredRevRecordSchema'

RECORD_ID_NAME = 'record_id'

RECORD_TYPE = 'issuer_cred_rev'

STATE_ISSUED = 'issued'

STATE_REVOKED = 'revoked'

TAG_NAMES = {'cred_def_id', 'cred_ex_id', 'cred_rev_id', 'rev_reg_id', 'state'}

WEBHOOK_TOPIC = 'issuer_cred_rev'

classmethod query_by_ids (*session:* *aries_cloudagent.core.profile.ProfileSession*,
*, *cred_def_id:* *str* = *None*, *rev_reg_id:*
str = *None*, *state:* *str* = *None*) → Sequence[aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord]

Retrieve issuer cred rev records by cred def id and/or rev reg id.

Parameters

- **session** – the profile session to use
- **cred_def_id** – the cred def id by which to filter
- **rev_reg_id** – the rev reg id by which to filter


```

class aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord(*,
    record_id:
        str
    =
    None,
    state:
        str
    =
    None,
    cred_def_id:
        str
    =
    None,
    er-
    ror_msg:
        str
    =
    None,
    is-
    suer_did:
        str
    =
    None,
    max_cred_n-
    int
    =
    None,
    re-
    voc_def_type:
        str
    =
    None,
    re-
    voc_reg_id:
        str
    =
    None,
    re-
    voc_reg_def:
        dict
    =
    None,
    re-
    voc_reg_entri-
    dict
    =
    None,
    tag:
        str
    =
    None,
    tails_hash:
        str
    =
    None,
    tails_local_L-
    str
    =
    None,
    tails_public_

```


Bases: *aries_cloudagent.messaging.models.base_record.BaseRecord*

Class for managing local issuing revocation registries.

LOG_STATE_FLAG = 'debug.revocation'

class Meta

Bases: *object*

IssuerRevRegRecord metadata.

schema_class = 'IssuerRevRegRecordSchema'

RECORD_ID_NAME = 'record_id'

RECORD_TYPE = 'issuer_rev_reg'

REVOC_DEF_TYPE_CL = 'CL_ACCUM'

STATE_ACTIVE = 'active'

STATE_FULL = 'full'

STATE_GENERATED = 'generated'

STATE_INIT = 'init'

STATE_POSTED = 'posted'

TAG_NAMES = {'cred_def_id', 'issuer_did', 'revoc_def_type', 'revoc_reg_id', 'state'}

WEBHOOK_TOPIC = 'revocation_registry'

clear_pending (*session: aries_cloudagent.core.profile.ProfileSession, cred_rev_ids: Sequence[str] = None*) → None

Clear pending revocations and save any resulting record change.

Parameters

- **session** – The profile session to use
- **cred_rev_ids** – Credential revocation identifiers to clear; default all

generate_registry (*profile: aries_cloudagent.core.profile.Profile*)

Create the revocation registry definition and tails file.

get_registry () → *aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry*

Create a *RevocationRegistry* instance from this record.

mark_pending (*session: aries_cloudagent.core.profile.ProfileSession, cred_rev_id: str*) → None

Mark a credential revocation id as revoked pending publication to ledger.

Parameters

- **session** – The profile session to use
- **cred_rev_id** – The credential revocation identifier for credential to revoke

classmethod query_by_cred_def_id (*session: aries_cloudagent.core.profile.ProfileSession, cred_def_id: str, state: str = None*) → *Sequence[aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord]*

Retrieve issuer revocation registry records by credential definition ID.

Parameters

- **session** – The profile session to use
- **cred_def_id** – The credential definition ID to filter by

- **state** – A state value to filter by

classmethod query_by_pending (*session: aries_cloudagent.core.profile.ProfileSession*) → Sequence[aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord]
Retrieve issuer revocation records with revocations pending.

Parameters session – The profile session to use

record_id
Accessor for the record ID.

record_value
Accessor for JSON value properties of this revocation registry record.

classmethod retrieve_by_revoc_reg_id (*session: aries_cloudagent.core.profile.ProfileSession, revoc_reg_id: str*) → aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord
Retrieve a revocation registry record by revocation registry ID.

Parameters

- **session** – The profile session to use
- **revoc_reg_id** – The revocation registry ID

send_def (*profile: aries_cloudagent.core.profile.Profile*)
Send the revocation registry definition to the ledger.

send_entry (*profile: aries_cloudagent.core.profile.Profile*)
Send a registry entry to the ledger.

set_state (*session: aries_cloudagent.core.profile.ProfileSession, state: str = None*)
Change the registry state (default full).

set_tails_file_public_uri (*profile: aries_cloudagent.core.profile.Profile, tails_file_uri: str*)
Update tails file's publicly accessible URI.

stage_pending_registry (*profile: aries_cloudagent.core.profile.Profile, max_attempts: int = 5*)
Prepare registry definition for future use.

class aries_cloudagent.revocation.models.issuer_rev_reg_record.**IssuerRevRegRecordSchema** (*args, **kwargs)

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

Schema to allow serialization/deserialization of issuer rev reg records.

class Meta
Bases: *object*
IssuerRevRegRecordSchema metadata.

model_class
alias of *IssuerRevRegRecord*

cred_def_id = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Length (minimum=1, maximum=100)>)>
error_msg = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Length (minimum=1, maximum=100)>)>
issuer_id = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Length (minimum=1, maximum=100)>)>
max_cred_num = <fields.Integer (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Range (minimum=1, maximum=100)>)>
pending_pub = <fields.List (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Length (minimum=1, maximum=100)>)>
record_id = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Length (minimum=1, maximum=100)>)>
revoc_def_type = <fields.String (default=<marshmallow.missing>, attribute=None, validate=<marshmallow.validate.Length (minimum=1, maximum=100)>)>

```
revoc_reg_def = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=N
revoc_reg_entry = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=
revoc_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, r
tag = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None, req
tails_hash = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
tails_local_path = <fields.String(default=<marshmallow.missing>, attribute=None, valid
tails_public_uri = <fields.String(default=<marshmallow.missing>, attribute=None, valid
```

aries_cloudagent.revocation.models.revocation_registry module

Classes for managing a revocation registry.

```

class aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry (registry_id:
                                                                    str
                                                                    =
                                                                    None,
                                                                    *,
                                                                    cred_def_id:
                                                                    str
                                                                    =
                                                                    None,
                                                                    is-
                                                                    suer_did:
                                                                    str
                                                                    =
                                                                    None,
                                                                    max_creds:
                                                                    int
                                                                    =
                                                                    None,
                                                                    reg_def_type:
                                                                    str
                                                                    =
                                                                    None,
                                                                    tag:
                                                                    str
                                                                    =
                                                                    None,
                                                                    tails_local_path:
                                                                    str
                                                                    =
                                                                    None,
                                                                    tails_public_url:
                                                                    str
                                                                    =
                                                                    None,
                                                                    tails_hash:
                                                                    str
                                                                    =
                                                                    None,
                                                                    reg_def:
                                                                    dict
                                                                    =
                                                                    None)

```

Bases: `object`

Manage a revocation registry and tails file.

MAX_SIZE = 32768

MIN_SIZE = 4

cred_def_id

Accessor for the credential definition ID.

classmethod from_definition (revoc_reg_def: dict, public_def: bool) →
 aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry
 Initialize a revocation registry instance from a definition.

get_or_fetch_local_tails_path()
Get the local tails path, retrieving from the remote if necessary.

get_receiving_tails_local_path()
Make the local path to the tails file we download from remote URI.

has_local_tails_file() → bool
Test if the tails file exists locally.

issuer_did
Accessor for the issuer DID.

max_creds
Accessor for the maximum number of issued credentials.

reg_def
Accessor for the revocation registry definition.

reg_def_type
Accessor for the revocation registry type.

registry_id
Accessor for the revocation registry ID.

retrieve_tails()
Fetch the tails file from the public URI.

tag
Accessor for the tag part of the revoc. reg. ID.

tails_hash
Accessor for the tails file hash.

tails_local_path
Accessor for the tails file local path.

tails_public_uri
Accessor for the tails file public URI.

Submodules

aries_cloudagent.revocation.error module

Revocation error classes.

exception `aries_cloudagent.revocation.error.RevocationError(*args, error_code: str = None, **kwargs)`

Bases: `aries_cloudagent.core.error.BaseError`

Base exception for revocation-related errors.

exception `aries_cloudagent.revocation.error.RevocationNotSupportedError(*args, error_code: str = None, **kwargs)`

Bases: `aries_cloudagent.revocation.error.RevocationError`

Attempted to perform revocation-related operation where inapplicable.

```
exception aries_cloudagent.revocation.error.RevocationRegistryBadSizeError (*args,  
                                                                           er-  
                                                                           ror_code:  
                                                                           str  
                                                                           =  
                                                                           None,  
                                                                           **kwargs)
```

Bases: `aries_cloudagent.revocation.error.RevocationError`

Attempted to create registry with maximum credentials too large or too small.

aries_cloudagent.revocation.indy module

Indy revocation registry management.

```
class aries_cloudagent.revocation.indy.IndyRevocation (profile:  
                                                         aries_cloudagent.core.profile.Profile)  
  
    Bases: object  
  
    Class for managing Indy credential revocation.  
  
    REV_REG_CACHE = {}  
  
    get_active_issuer_rev_reg_record (cred_def_id: str) →  
                                     aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord  
    Return current active registry for issuing a given credential definition.  
  
        Parameters cred_def_id – ID of the base credential definition  
  
    get_issuer_rev_reg_record (revoc_reg_id: str) → aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRe  
    Return a revocation registry record by identifier.  
  
        Parameters revoc_reg_id – ID of the revocation registry  
  
    get_ledger_registry (revoc_reg_id: str) → aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry  
    Get a revocation registry from the ledger, fetching as necessary.  
  
    init_issuer_registry (cred_def_id: str, max_cred_num: int = None, re-  
                          voc_def_type: str = None, tag: str = None) →  
                          aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord  
    Create a new revocation registry record for a credential definition.  
  
    list_issuer_registries () → Sequence[aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord]  
    List the issuer's current revocation registries.
```

aries_cloudagent.revocation.manager module

Classes to manage credential revocation.

```
class aries_cloudagent.revocation.manager.RevocationManager (profile:  
                                                             aries_cloudagent.core.profile.Profile)  
  
    Bases: object  
  
    Class for managing revocation operations.  
  
    clear_pending_revocations (purge: Mapping[str, Sequence[str]] = None) → Mapping[str, Se-  
                                     quence[str]]  
    Clear pending revocation publications.
```

Parameters **purge** – Mapping from revocation registry identifiers to all credential revocation identifiers within each to clear. Specify null/empty map for all revocation registries. Specify empty sequence per revocation registry identifier for all pending within the revocation registry; e.g.,

```
{ } - clear all pending revocations from all revocation registries
{ "R17v42T4pk...:4:R17v42T4pk...:3:CL:19:tag:CL_ACCUM:0": [],
  "R17v42T4pk...:4:R17v42T4pk...:3:CL:19:tag:CL_ACCUM:1": ["1", "2"]
} - clear
```

- all pending revocations from all revocation registry tagged 0
- pending ["1", "2"] from revocation registry tagged 1
- no pending revocations from any other revocation registries.

Returns mapping from revocation registry id to its remaining cred rev ids still marked pending, omitting revocation registries with no remaining pending publications.

publish_pending_revocations (*rrid2crid: Mapping[str, Sequence[str]] = None*) → Mapping[str, Sequence[str]]

Publish pending revocations to the ledger.

Parameters **rrid2crid** – Mapping from revocation registry identifiers to all credential revocation identifiers within each to publish. Specify null/empty map for all revocation registries. Specify empty sequence per revocation registry identifier for all pending within the revocation registry; e.g.,

```
{ } - publish all pending revocations from all revocation registries
{ "R17v42T4pk...:4:R17v42T4pk...:3:CL:19:tag:CL_ACCUM:0": [],
  "R17v42T4pk...:4:R17v42T4pk...:3:CL:19:tag:CL_ACCUM:1": ["1", "2"]
} - publish:
```

- all pending revocations from all revocation registry tagged 0
- pending ["1", "2"] from revocation registry tagged 1
- no pending revocations from any other revocation registries.

Returns: mapping from each revocation registry id to its cred rev ids published.

revoke_credential (*rev_reg_id: str, cred_rev_id: str, publish: bool = False*)

Revoke a credential.

Optionally, publish the corresponding revocation registry delta to the ledger.

Parameters

- **rev_reg_id** – revocation registry id
- **cred_rev_id** – credential revocation id
- **publish** – whether to publish the resulting revocation registry delta, along with any revocations pending against it

revoke_credential_by_cred_ex_id (*cred_ex_id: str, publish: bool = False*)

Revoke a credential by its credential exchange identifier at issue.

Optionally, publish the corresponding revocation registry delta to the ledger.

Parameters

- **cred_ex_id** – credential exchange identifier

- **publish** – whether to publish the resulting revocation registry delta, along with any revocations pending against it

```
exception aries_cloudagent.revocation.manager.RevocationManagerError(*args,  
                                                                    er-  
                                                                    ror_code:  
                                                                    str =  
                                                                    None,  
                                                                    **kwargs)
```

Bases: `aries_cloudagent.core.error.BaseError`

Revocation manager error.

aries_cloudagent.revocation.routes module

Revocation registry admin routes.

```
class aries_cloudagent.revocation.routes.ClearPendingRevocationsRequestSchema(*args,  
                                                                              **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Request schema for clear pending revocations API call.

```
purge = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None, req
```

```
class aries_cloudagent.revocation.routes.CredDefIdMatchInfoSchema(*args,  
                                                                  **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Path parameters and validators for request taking cred def id.

```
cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
```

```
class aries_cloudagent.revocation.routes.CredExIdMatchInfoSchema(*args,  
                                                                **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Path parameters and validators for request taking credential exchange id.

```
cred_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
```

```
class aries_cloudagent.revocation.routes.CredRevRecordQueryStringSchema(*args,  
                                                                        **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Parameters and validators for credential revocation record request.

```
cred_ex_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
```

```
cred_rev_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<
```

```
rev_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<a
```

```
validate_fields(data, **kwargs)
```

Validate schema fields - must have (rr-id and cr-id) xor cx-id.

```
class aries_cloudagent.revocation.routes.CredRevRecordResultSchema(*args,  
                                                                    **kwargs)
```

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Result schema for credential revocation record request.

```
result
```

Schema to allow de/serialization of credential revocation records.


```

class aries_cloudagent.revocation.routes.PublishRevocationsSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request and result schema for revocation publication API call.
    rrid2crid = <fields.Dict(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.revocation.routes.RevRegCreateRequestSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for revocation registry creation request.
    credential_definition_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    max_cred_num = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.revocation.routes.RevRegIdMatchInfoSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Path parameters and validators for request taking rev reg id.
    rev_reg_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.revocation.routes.RevRegIssuedResultSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Result schema for revocation registry credentials issued request.
    result = <fields.Integer(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.revocation.routes.RevRegResultSchema(*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Result schema for revocation registry creation request.
    result
        Schema to allow serialization/deserialization of issuer rev reg records.
class aries_cloudagent.revocation.routes.RevRegUpdateTailsFileUriSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Request schema for updating tails file URI.
    tails_public_uri = <fields.Url(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.revocation.routes.RevRegsCreatedQueryStringSchema(*args,
                                                                **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Query string parameters and validators for rev regs created request.
    cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
    state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=None,
class aries_cloudagent.revocation.routes.RevRegsCreatedSchema(*args, **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
    Result schema for request for revocation registries created.
    rev_reg_ids = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None,

```

```
class aries_cloudagent.revocation.routes.RevocationModuleResponseSchema (*args,
                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
```

Response schema for Revocation Module.

```
class aries_cloudagent.revocation.routes.RevokeRequestSchema (*args, **kwargs)
    Bases: aries_cloudagent.revocation.routes.CredRevRecordQueryStringSchema
```

Parameters and validators for revocation request.

```
publish = <fields.Boolean(default=<marshmallow.missing>, attribute=None, validate=None)
```

```
class aries_cloudagent.revocation.routes.SetRevRegStateQueryStringSchema (*args,
                                                                    **kwargs)
    Bases: aries_cloudagent.messaging.models.openapi.OpenAPISchema
```

Query string parameters and validators for request to set rev reg state.

```
state = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<OneOf(
```

```
aries_cloudagent.revocation.routes.clear_pending_revocations (request:
                                                                <sphinx.ext.autodoc.importer._MockObject
                                                                object at
                                                                0x7f760d85a810>)
```

Request handler for clearing pending revocations.

Parameters **request** – aiohttp request object

Returns Credential revocation ids still pending revocation by revocation registry id.

```
aries_cloudagent.revocation.routes.create_rev_reg (request:
                                                    <sphinx.ext.autodoc.importer._MockObject
                                                    object at 0x7f760d85a810>)
```

Request handler to create a new revocation registry.

Parameters **request** – aiohttp request object

Returns The issuer revocation registry record

```
aries_cloudagent.revocation.routes.get_active_rev_reg (request:
                                                        <sphinx.ext.autodoc.importer._MockObject
                                                        object at 0x7f760d85a810>)
```

Request handler to get current active revocation registry by cred def id.

Parameters **request** – aiohttp request object

Returns The revocation registry identifier

```
aries_cloudagent.revocation.routes.get_cred_rev_record (request:
                                                         <sphinx.ext.autodoc.importer._MockObject
                                                         object at
                                                         0x7f760d85a810>)
```

Request handler to get credential revocation record.

Parameters **request** – aiohttp request object

Returns The issuer credential revocation record

```
aries_cloudagent.revocation.routes.get_rev_reg (request:
                                                  <sphinx.ext.autodoc.importer._MockObject
                                                  object at 0x7f760d85a810>)
```

Request handler to get a revocation registry by rev reg id.

Parameters **request** – aiohttp request object

Returns The revocation registry

```
aries_cloudagent.revocation.routes.get_rev_reg_issued(request:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Request handler to get number of credentials issued against revocation registry.

Parameters **request** – aiohttp request object

Returns Number of credentials issued against revocation registry

```
aries_cloudagent.revocation.routes.get_tails_file(request:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>) →
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>
```

Request handler to download tails file for revocation registry.

Parameters **request** – aiohttp request object

Returns The tails file in FileResponse

```
aries_cloudagent.revocation.routes.post_process_routes(app:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Amend swagger API.

```
aries_cloudagent.revocation.routes.publish_revocations(request:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Request handler for publishing pending revocations to the ledger.

Parameters **request** – aiohttp request object

Returns Credential revocation ids published as revoked by revocation registry id.

```
aries_cloudagent.revocation.routes.register(app: <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Register routes.

```
aries_cloudagent.revocation.routes.rev_regs_created(request:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Request handler to get revocation registries that current agent created.

Parameters **request** – aiohttp request object

Returns List of identifiers of matching revocation registries.

```
aries_cloudagent.revocation.routes.revoke(request: <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Request handler for storing a credential request.

Parameters **request** – aiohttp request object

Returns The credential request details.

```
aries_cloudagent.revocation.routes.send_rev_reg_def(request:
    <sphinx.ext.autodoc.importer._MockObject
    object at 0x7f760d85a810>)
```

Request handler to send revocation registry definition by reg id to ledger.

Parameters **request** – aiohttp request object

Returns The issuer revocation registry record

`aries_cloudagent.revocation.routes.send_rev_reg_entry` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760d85a810>)

Request handler to send rev reg entry by registry id to ledger.

Parameters `request` – aiohttp request object

Returns The revocation registry record

`aries_cloudagent.revocation.routes.set_rev_reg_state` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760d85a810>)

Request handler to set a revocation registry state manually.

Parameters `request` – aiohttp request object

Returns The revocation registry record, updated

`aries_cloudagent.revocation.routes.update_rev_reg` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760d85a810>)

Request handler to update a rev reg's public tails URI by registry id.

Parameters `request` – aiohttp request object

Returns The revocation registry record

`aries_cloudagent.revocation.routes.upload_tails_file` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760d85a810>)

Request handler to upload local tails file for revocation registry.

Parameters `request` – aiohttp request object

aries_cloudagent.storage package

Submodules

aries_cloudagent.storage.base module

Abstract base classes for non-secrets storage.

class `aries_cloudagent.storage.base.BaseStorage`

Bases: `abc.ABC`

Abstract stored records interface.

add_record (*record: aries_cloudagent.storage.record.StorageRecord*)

Add a new record to the store.

Parameters `record` – *StorageRecord* to be stored

delete_all_records (*type_filter: str, tag_query: Mapping[KT, VT_co] = None*)

Remove all records matching a particular type filter and tag query.

delete_record (*record: aries_cloudagent.storage.record.StorageRecord*)

Delete an existing record.

Parameters `record` – *StorageRecord* to delete

find_all_records (*type_filter: str, tag_query: Mapping[KT, VT_co] = None, options: Mapping[KT, VT_co] = None*)

Retrieve all records matching a particular type filter and tag query.

find_record (*type_filter: str, tag_query: Mapping[KT, VT_co] = None, options: Mapping[KT, VT_co] = None*) → `aries_cloudagent.storage.record.StorageRecord`

Find a record using a unique tag filter.

Parameters

- **type_filter** – Filter string
- **tag_query** – Tags to query
- **options** – Dictionary of backend-specific options

get_record (*record_type: str, record_id: str, options: Mapping[KT, VT_co] = None*) → `aries_cloudagent.storage.record.StorageRecord`

Fetch a record from the store by type and ID.

Parameters

- **record_type** – The record type
- **record_id** – The record id
- **options** – A dictionary of backend-specific options

Returns A *StorageRecord* instance

update_record (*record: aries_cloudagent.storage.record.StorageRecord, value: str, tags: Mapping[KT, VT_co]*)

Update an existing stored record's value and tags.

Parameters

- **record** – *StorageRecord* to update
- **value** – The new value
- **tags** – The new tags

class `aries_cloudagent.storage.base.BaseStorageSearch`

Bases: `abc.ABC`

Abstract stored records search interface.

search_records (*type_filter: str, tag_query: Mapping[KT, VT_co] = None, page_size: int = None, options: Mapping[KT, VT_co] = None*) → `aries_cloudagent.storage.base.BaseStorageSearchSession`

Create a new record query.

Parameters

- **type_filter** – Filter string
- **tag_query** – Tags to query
- **page_size** – Page size
- **options** – Dictionary of backend-specific options

Returns An instance of *BaseStorageSearchSession*

class `aries_cloudagent.storage.base.BaseStorageSearchSession`

Bases: `abc.ABC`

Abstract stored records search session interface.

close()

Dispose of the search query.

fetch (*max_count: int = None*) → Sequence[aries_cloudagent.storage.record.StorageRecord]

Fetch the next list of results from the store.

Parameters **max_count** – Max number of records to return. If not provided, defaults to the backend’s preferred page size

Returns A list of *StorageRecord* instances

class aries_cloudagent.storage.base.**IterSearch** (*search: aries_cloudagent.storage.base.BaseStorageSearchSession, page_size: int = None*)

Bases: *object*

A generic record search async iterator.

aries_cloudagent.storage.base.**validate_record** (*record: aries_cloudagent.storage.record.StorageRecord, *, delete=False*)

Ensure that a record is ready to be saved/updated/deleted.

aries_cloudagent.storage.error module

Storage-related exceptions.

exception aries_cloudagent.storage.error.**StorageDuplicateError** (**args, error_code: str = None, **kwargs*)

Bases: *aries_cloudagent.storage.error.StorageError*

Duplicate record found in storage.

exception aries_cloudagent.storage.error.**StorageError** (**args, error_code: str = None, **kwargs*)

Bases: *aries_cloudagent.core.error.BaseError*

Base class for Storage errors.

exception aries_cloudagent.storage.error.**StorageNotFoundError** (**args, error_code: str = None, **kwargs*)

Bases: *aries_cloudagent.storage.error.StorageError*

Record not found in storage.

exception aries_cloudagent.storage.error.**StorageSearchError** (**args, error_code: str = None, **kwargs*)

Bases: *aries_cloudagent.storage.error.StorageError*

General exception during record search.

aries_cloudagent.storage.in_memory module

Basic in-memory storage implementation (non-wallet).

class aries_cloudagent.storage.in_memory.**InMemoryStorage** (*profile: aries_cloudagent.core.in_memory.InMemoryProfile*)

Bases: *aries_cloudagent.storage.base.BaseStorage, aries_cloudagent.storage.base.BaseStorageSearch*

Basic in-memory storage class.

add_record (*record*: *aries_cloudagent.storage.record.StorageRecord*)

Add a new record to the store.

Parameters **record** – *StorageRecord* to be stored

Raises

- *StorageError* – If no record is provided
- *StorageError* – If the record has no ID

delete_all_records (*type_filter*: *str*, *tag_query*: *Mapping[KT, VT_co] = None*)

Remove all records matching a particular type filter and tag query.

delete_record (*record*: *aries_cloudagent.storage.record.StorageRecord*)

Delete a record.

Parameters **record** – *StorageRecord* to delete

Raises *StorageNotFoundError* – If record not found

find_all_records (*type_filter*: *str*, *tag_query*: *Mapping[KT, VT_co] = None*, *options*: *Mapping[KT, VT_co] = None*)

Retrieve all records matching a particular type filter and tag query.

get_record (*record_type*: *str*, *record_id*: *str*, *options*: *Mapping[KT, VT_co] = None*) → *aries_cloudagent.storage.record.StorageRecord*

Fetch a record from the store by type and ID.

Parameters

- **record_type** – The record type
- **record_id** – The record id
- **options** – A dictionary of backend-specific options

Returns A *StorageRecord* instance

Raises *StorageNotFoundError* – If the record is not found

search_records (*type_filter*: *str*, *tag_query*: *Mapping[KT, VT_co] = None*, *page_size*: *int = None*, *options*: *Mapping[KT, VT_co] = None*) → *aries_cloudagent.storage.in_memory.InMemoryStorageSearch*

Search stored records.

Parameters

- **type_filter** – Filter string
- **tag_query** – Tags to query
- **page_size** – Page size
- **options** – Dictionary of backend-specific options

Returns An instance of *InMemoryStorageSearch*

update_record (*record*: *aries_cloudagent.storage.record.StorageRecord*, *value*: *str*, *tags*: *Mapping[KT, VT_co]*)

Update an existing stored record's value.

Parameters

- **record** – *StorageRecord* to update
- **value** – The new value

- **tags** – The new tags

Raises `StorageNotFoundError` – If record not found

```
class aries_cloudagent.storage.in_memory.InMemoryStorageSearch (profile:
                                                                    aries_cloudagent.core.in_memory.InMem
                                                                    type_filter: str,
                                                                    tag_query:
                                                                    Mapping[KT,
                                                                    VT_co],
                                                                    page_size: int =
                                                                    None, options:
                                                                    Mapping[KT,
                                                                    VT_co]          =
                                                                    None)
```

Bases: `aries_cloudagent.storage.base.BaseStorageSearchSession`

Represent an active stored records search.

close ()

Dispose of the search query.

fetch (*max_count: int = None*) → Sequence[`aries_cloudagent.storage.record.StorageRecord`]

Fetch the next list of results from the store.

Parameters **max_count** – Max number of records to return. If not provided, defaults to the backend's preferred page size

Returns A list of `StorageRecord` instances

Raises `StorageSearchError` – If the search query has not been opened

```
aries_cloudagent.storage.in_memory.tag_query_match (tags: dict, tag_query: dict) →
                                                    bool
```

Match simple tag filters (string values).

```
aries_cloudagent.storage.in_memory.tag_value_match (value: str, match: dict) → bool
```

Match a single tag against a tag subquery.

TODO: What type coercion is needed? (support int or float values?)

aries_cloudagent.storage.indy module

Indy implementation of BaseStorage interface.

```
class aries_cloudagent.storage.indy.IndySdkStorage (wallet:
                                                    aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet)
Bases: aries_cloudagent.storage.base.BaseStorage, aries_cloudagent.storage.
base.BaseStorageSearch
```

Indy Non-Secrets interface.

add_record (*record: aries_cloudagent.storage.record.StorageRecord*)

Add a new record to the store.

Parameters **record** – `StorageRecord` to be stored

delete_all_records (*type_filter: str, tag_query: Mapping[KT, VT_co] = None*)

Remove all records matching a particular type filter and tag query.

delete_record (*record: aries_cloudagent.storage.record.StorageRecord*)

Delete a record.

Parameters **record** – *StorageRecord* to delete

Raises

- *StorageNotFoundError* – If record not found
- *StorageError* – If a libindy error occurs

find_all_records (*type_filter*: *str*, *tag_query*: *Mapping[KT, VT_co]* = *None*, *options*: *Mapping[KT, VT_co]* = *None*) →

Retrieve all records matching a particular type filter and tag query.

get_record (*record_type*: *str*, *record_id*: *str*, *options*: *Mapping[KT, VT_co]* = *None*) → *aries_cloudagent.storage.record.StorageRecord*

Fetch a record from the store by type and ID.

Parameters

- **record_type** – The record type
- **record_id** – The record id
- **options** – A dictionary of backend-specific options

Returns A *StorageRecord* instance

Raises

- *StorageError* – If the record is not provided
- *StorageError* – If the record ID not provided
- *StorageNotFoundError* – If the record is not found
- *StorageError* – If record not found

search_records (*type_filter*: *str*, *tag_query*: *Mapping[KT, VT_co]* = *None*, *page_size*: *int* = *None*, *options*: *Mapping[KT, VT_co]* = *None*) → *aries_cloudagent.storage.indy.IndySdkStorageSearch*

Search stored records.

Parameters

- **type_filter** – Filter string
- **tag_query** – Tags to query
- **page_size** – Page size
- **options** – Dictionary of backend-specific options

Returns An instance of *IndySdkStorageSearch*

update_record (*record*: *aries_cloudagent.storage.record.StorageRecord*, *value*: *str*, *tags*: *Mapping[KT, VT_co]*)

Update an existing stored record's value and tags.

Parameters

- **record** – *StorageRecord* to update
- **value** – The new value
- **tags** – The new tags

Raises

- *StorageNotFoundError* – If record not found
- *StorageError* – If a libindy error occurs

wallet

Accessor for IndyOpenWallet instance.

```
class aries_cloudagent.storage.indy.IndySdkStorageSearch (store:
    aries_cloudagent.storage.indy.IndySdkStorage,
    type_filter: str,
    tag_query: Mapping[KT,
    VT_co], page_size:
    int = None, options:
    Mapping[KT, VT_co] =
    None)
```

Bases: *aries_cloudagent.storage.base.BaseStorageSearchSession*

Represent an active stored records search.

close()

Dispose of the search query.

fetch (*max_count: int = None*) → Sequence[aries_cloudagent.storage.record.StorageRecord]

Fetch the next list of results from the store.

Parameters **max_count** – Max number of records to return. If not provided, defaults to the backend's preferred page size

Returns A list of *StorageRecord* instances

Raises *StorageSearchError* – If the search query has not been opened

aries_cloudagent.storage.record module

Record instance stored and searchable by BaseStorage implementation.

```
class aries_cloudagent.storage.record.StorageRecord
    Bases: aries_cloudagent.storage.record.StorageRecord

    Storage record class.
```

aries_cloudagent.tails package

Submodules

aries_cloudagent.tails.base module

Tails server interface base class.

```
class aries_cloudagent.tails.base.BaseTailsServer
    Bases: abc.ABC

    Base class for tails server interface.

    upload_tails_file (context: aries_cloudagent.config.injection_context.InjectionContext,
        rev_reg_id: str, tails_file_path: str, interval: float = 1.0, backoff: float
        = 0.25, max_attempts: int = 5) → Tuple[bool, str]

    Upload tails file to tails server.
```

Parameters

- **rev_reg_id** – The revocation registry identifier
- **tails_file** – The path to the tails file to upload

- **interval** – initial interval between attempts
- **backoff** – exponential backoff in retry interval
- **max_attempts** – maximum number of attempts to make

aries_cloudagent.tails.error module

Tails server related errors.

exception `aries_cloudagent.tails.error.TailsServerNotConfiguredError` (**args*, *error_code: str* = *None*, ***kwargs*)

Bases: `aries_cloudagent.core.error.BaseError`

Error indicating the tails server plugin hasn't been configured.

aries_cloudagent.tails.indy_tails_server module

Indy tails server interface class.

class `aries_cloudagent.tails.indy_tails_server.IndyTailsServer`

Bases: `aries_cloudagent.tails.base.BaseTailsServer`

Indy tails server interface.

upload_tails_file (*context*, *rev_reg_id: str*, *tails_file_path: str*, *interval: float* = *1.0*, *backoff: float* = *0.25*, *max_attempts: int* = *5*) → `Tuple[bool, str]`

Upload tails file to tails server.

Parameters

- **context** – context with configuration settings
- **rev_reg_id** – revocation registry identifier
- **tails_file_path** – path to the tails file to upload
- **interval** – initial interval between attempts
- **backoff** – exponential backoff in retry interval
- **max_attempts** – maximum number of attempts to make

aries_cloudagent.transport package

Subpackages

aries_cloudagent.transport.inbound package

Submodules

aries_cloudagent.transport.inbound.base module

Base inbound transport class.

```
class aries_cloudagent.transport.inbound.base.BaseInboundTransport (scheme:  
                                                                    str, create_session:  
                                                                    Callable,  
                                                                    *,  
                                                                    max_message_size:  
                                                                    int = 0,  
                                                                    wire_format:  
                                                                    aries_cloudagent.transport.wire_format.BaseWireFormat = None)
```

Bases: `abc.ABC`

Base inbound transport class.

```
create_session (*, accept_undelivered: bool = False, can_respond: bool = False, client_info: dict  
                 = None, wire_format: aries_cloudagent.transport.wire_format.BaseWireFormat =  
                 None) → Awaitable[aries_cloudagent.transport.inbound.session.InboundSession]  
Create a new inbound session.
```

Parameters

- **accept_undelivered** – Flag for accepting undelivered messages
- **can_respond** – Flag indicating that the transport can send responses
- **client_info** – Request-specific client information
- **wire_format** – Optionally override the session wire format

max_message_size

Accessor for this transport's max message size.

scheme

Accessor for this transport's scheme.

start () → None

Start listening for on this transport.

stop () → None

Stop listening for on this transport.

```
class aries_cloudagent.transport.inbound.base.InboundTransportConfiguration (module,  
                                                                              host,  
                                                                              port)
```

Bases: `tuple`

host

Alias for field number 1

module

Alias for field number 0

port

Alias for field number 2

exception `aries_cloudagent.transport.inbound.base.InboundTransportError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.transport.error.TransportError`

Generic inbound transport error.

exception `aries_cloudagent.transport.inbound.base.InboundTransportRegistrationError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.transport.inbound.base.InboundTransportError`

Error in loading an inbound transport.

exception `aries_cloudagent.transport.inbound.base.InboundTransportSetupError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.transport.inbound.base.InboundTransportError`

Setup error for an inbound transport.

`aries_cloudagent.transport.inbound.delivery_queue` module

The Delivery Queue.

The delivery queue holds and manages messages that have not yet been delivered to their intended destination.

class `aries_cloudagent.transport.inbound.delivery_queue.DeliveryQueue`
Bases: `object`

DeliveryQueue class.

Manages undelivered messages.

add_message (`msg: aries_cloudagent.transport.outbound.message.OutboundMessage`)
Add an OutboundMessage to delivery queue.

The message is added once per recipient key

Parameters `msg` – The OutboundMessage to add

expire_messages (`ttl=None`)
Expire messages that are past the time limit.

Parameters `ttl` – Optional. Allows override of configured ttl

get_one_message_for_key (`key: str`)
Remove and return a matching message.

Parameters `key` – The key to use for lookup

has_message_for_key (*key: str*)
Check for queued messages by key.

Parameters **key** – The key to use for lookup

inspect_all_messages_for_key (*key: str*)
Return all messages for key.

Parameters **key** – The key to use for lookup

message_count_for_key (*key: str*)
Count of queued messages by key.

Parameters **key** – The key to use for lookup

remove_message_for_key (*key: str, msg: aries_cloudagent.transport.outbound.message.OutboundMessage*)
Remove specified message from queue for key.

Parameters

- **key** – The key to use for lookup
- **msg** – The message to remove from the queue

class `aries_cloudagent.transport.inbound.delivery_queue.QueuedMessage` (*msg: aries_cloudagent.transport.outbound.message.OutboundMessage*)

Bases: `object`

Wrapper Class for queued messages.

Allows tracking Metadata.

older_than (*compare_timestamp: float*) → bool
Age Comparison.

Allows you to test age as compared to the provided timestamp.

Parameters **compare_timestamp** – The timestamp to compare

aries_cloudagent.transport.inbound.http module

Http Transport classes and functions.

class `aries_cloudagent.transport.inbound.http.HttpTransport` (*host: str, port: int, create_session: bool, **kwargs*)

Bases: `aries_cloudagent.transport.inbound.base.BaseInboundTransport`

Http Transport class.

inbound_message_handler (*request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e1ca910>*)
Message handler for inbound messages.

Parameters **request** – aiohttp request object

Returns The web response

invite_message_handler (*request: <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e1ca910>*)
Message handler for invites.

Parameters **request** – aiohttp request object

Returns The web response

make_application() → <sphinx.ext.autodoc.importer._MockObject object at 0x7f760e1ca910>
Construct the aiohttp application.

start() → None
Start this transport.

Raises InboundTransportSetupError – If there was an error starting the webserver

stop() → None
Stop this transport.

aries_cloudagent.transport.inbound.manager module

Inbound transport manager.

```
class aries_cloudagent.transport.inbound.manager.InboundTransportManager (profile:
    aries_cloudagent.core.pro
    re-
    ceive_inbound:
    Corou-
    tine[T_co,
    T_contra,
    V_co],
    re-
    turn_inbound:
    Callable
    =
    None)
```

Bases: `object`

Inbound transport manager class.

closed_session (session: *aries_cloudagent.transport.inbound.session.InboundSession*)
Clean up a closed session.

Returns an undelivered message to the caller if possible.

create_session (transport_type: *str*, *, accept_undelivered: *bool* = *False*,
can_respond: *bool* = *False*, client_info: *dict* = *None*, wire_format:
aries_cloudagent.transport.wire_format.BaseWireFormat = *None*)
Create a new inbound session.

Parameters

- **transport_type** – The inbound transport identifier
- **accept_undelivered** – Flag for accepting undelivered messages
- **can_respond** – Flag indicating that the transport can send responses
- **client_info** – An optional dict describing the client
- **wire_format** – Override the wire format for this session

dispatch_complete (message: *aries_cloudagent.transport.inbound.message.InboundMessage*,
completed: *aries_cloudagent.utils.task_queue.CompletedTask*)
Handle completion of message dispatch.

get_transport_instance (transport_id: *str*) → *aries_cloudagent.transport.inbound.base.BaseInboundTransport*
Get an instance of a running transport by ID.

process_undelivered (*session: aries_cloudagent.transport.inbound.session.InboundSession*)
Interact with undelivered queue to find applicable messages.

Parameters **session** – The inbound session

register (*config: aries_cloudagent.transport.inbound.base.InboundTransportConfiguration*) → str
Register transport module.

Parameters **config** – The inbound transport configuration

register_transport (*transport: aries_cloudagent.transport.inbound.base.BaseInboundTransport*,
transport_id: str) → str
Register a new inbound transport class.

Parameters

- **transport** – Transport instance to register
- **transport_id** – The transport ID to register

return_to_session (*outbound: aries_cloudagent.transport.outbound.message.OutboundMessage*)
→ bool
Return an outbound message via an open session, if possible.

return_undelivered (*outbound: aries_cloudagent.transport.outbound.message.OutboundMessage*)
→ bool
Add an undelivered message to the undelivered queue.

At this point the message could not be associated with an inbound session and could not be delivered via an outbound transport.

setup ()
Perform setup operations.

start ()
Start all registered transports.

start_transport (*transport_id: str*)
Start a registered inbound transport.

Parameters **transport_id** – ID for the inbound transport to start

stop (*wait: bool = True*)
Stop all registered transports.

aries_cloudagent.transport.inbound.message module

Classes representing inbound messages.

```
class aries_cloudagent.transport.inbound.message.InboundMessage (payload:  
    Union[str,  
    bytes], receipt:  
    aries_cloudagent.transport.inbound.receipt.InboundReceipt,  
    *, connec-  
    tion_id: str  
    = None, ses-  
    sion_id: str =  
    None, trans-  
    port_type: str  
    = None)
```

Bases: `object`

Container class linking a message payload with its receipt details.

aries_cloudagent.transport.inbound.receipt module

Classes for representing message receipt details.

```
class aries_cloudagent.transport.inbound.receipt.MessageReceipt (*,      connec-
                                                                    tion_id:  str
                                                                    = None, di-
                                                                    rect_response_mode:
                                                                    str = None,
                                                                    in_time: date-
                                                                    time.datetime
                                                                    = None,
                                                                    raw_message:
                                                                    str = None,
                                                                    recipi-
                                                                    ent_verkey: str
                                                                    = None, recipi-
                                                                    ent_id:  str
                                                                    = None, recipi-
                                                                    ent_id_public:
                                                                    bool = None,
                                                                    sender_id:
                                                                    str = None,
                                                                    sender_verkey:
                                                                    str = None,
                                                                    thread_id: str
                                                                    = None)
```

Bases: `object`

Properties of an agent message's delivery.

REPLY_MODE_ALL = 'all'

REPLY_MODE_NONE = 'none'

REPLY_MODE_THREAD = 'thread'

connection_id

Accessor for the pairwise connection identifier.

Returns This context's connection identifier

direct_response_mode

Accessor for the requested direct response mode.

Returns This context's requested direct response mode

direct_response_requested

Accessor for the the state of the direct response mode.

Returns This context's requested direct response mode

in_time

Accessor for the datetime the message was received.

Returns This context's received time

raw_message

Accessor for the raw message text.

Returns The raw message text

recipient_did

Accessor for the recipient DID which corresponds with the verkey.

Returns The recipient DID

recipient_did_public

Check if the recipient did is public.

Indicates whether the message is associated with a public (ledger) recipient DID.

Returns True if the recipient's DID is public, else false

recipient_verkey

Accessor for the recipient verkey key used to pack the incoming request.

Returns The recipient verkey

sender_did

Accessor for the sender DID which corresponds with the verkey.

Returns The sender did

sender_verkey

Accessor for the sender public key used to pack the incoming request.

Returns This context's sender's verkey

thread_id

Accessor for the identifier of the message thread.

Returns The delivery thread ID

aries_cloudagent.transport.inbound.session module

Inbound connection handling classes.

```
class aries_cloudagent.transport.inbound.session.AcceptResult (accepted: bool,  
retry: bool =  
False)
```

Bases: `object`

Represent the result of `accept_response`.

```

class aries_cloudagent.transport.inbound.session.InboundSession(*,
    profile:
        aries_cloudagent.core.profile.Profile,
    in-
        bound_handler:
            Callable,
    session_id: str,
    wire_format:
        aries_cloudagent.transport.wire_format.WireFormat,
    accept_undelivered:
        bool = False,
    can_respond:
        bool = False,
    client_info:
        dict = None,
    close_handler:
        Callable =
        None,
    reply_mode: str
        = None,
    reply_thread_ids:
        Sequence[str]
        = None,
    reply_verkeys:
        Sequence[str]
        = None,
    transport_type: str
        = None)

```

Bases: `object`

Track an open transport connection for direct routing of outbound messages.

accept_response (*message: aries_cloudagent.transport.outbound.message.OutboundMessage*) → *aries_cloudagent.transport.inbound.session.AcceptResult*
 Try to queue an outbound message if it applies to this session.

Returns: a tuple of (message buffered, retry later)

add_reply_thread_ids (**thids*)
 Add a thread ID to the set of potential reply targets.

add_reply_verkeys (**verkeys*)
 Add a verkey to the set of potential reply targets.

can_respond
 Accessor for the session can-respond state.

clear_response ()
 Handle when the buffered response message has been delivered.

close ()
 Setter for the session closed state.

closed
 Accessor for the session closed state.

encode_outbound (*outbound: aries_cloudagent.transport.outbound.message.OutboundMessage*) → *aries_cloudagent.transport.outbound.message.OutboundMessage*
 Apply wire formatting to an outbound message.

handle_relay_context (*payload_enc: Union[str, bytes]*)

Update the session profile based on the recipients of an incoming message.

parse_inbound (*payload_enc: Union[str, bytes]*) → *aries_cloudagent.transport.inbound.message.InboundMessage*

Convert a message payload and to an inbound message.

process_inbound (*message: aries_cloudagent.transport.inbound.message.InboundMessage*)

Process an incoming message and update the session metadata as necessary.

Parameters *message* – The inbound message instance

receive (*payload_enc: Union[str, bytes]*) → *aries_cloudagent.transport.inbound.message.InboundMessage*

Receive a new message payload and dispatch the message.

receive_inbound (*message: aries_cloudagent.transport.inbound.message.InboundMessage*)

Deliver the inbound message to the conductor.

reply_mode

Accessor for the session reply mode.

reply_thread_ids

Accessor for the reply thread IDs.

reply_verkeys

Accessor for the reply verkeys.

response_buffered

Check if a response is currently buffered.

select_outbound (*message: aries_cloudagent.transport.outbound.message.OutboundMessage*) →
bool

Determine if an outbound message should be sent to this session.

Parameters *message* – The outbound message to be checked

set_response (*message: aries_cloudagent.transport.outbound.message.OutboundMessage*)

Set the contents of the response message buffer.

wait_response () → *Union[str, bytes]*

Wait for a response to be buffered and pack it.

aries_cloudagent.transport.inbound.ws module

Websockets Transport classes and functions.

class *aries_cloudagent.transport.inbound.ws.WsTransport* (*host: str, port: int, create_session, **kwargs*)

Bases: *aries_cloudagent.transport.inbound.base.BaseInboundTransport*

Websockets Transport class.

inbound_message_handler (*request*)

Message handler for inbound messages.

Parameters *request* – aiohttp request object

Returns The web response

make_application () → *<sphinx.ext.autodoc.importer._MockObject object at 0x7f760d764590>*

Construct the aiohttp application.

scheme

Accessor for this transport's scheme.

start() → None

Start this transport.

Raises `InboundTransportSetupError` – If there was an error starting the webserver

stop() → None

Stop this transport.

aries_cloudagent.transport.outbound package

Submodules

aries_cloudagent.transport.outbound.base module

Base outbound transport.

```
class aries_cloudagent.transport.outbound.base.BaseOutboundTransport (wire_format:
                                                                    aries_cloudagent.transport.wire
                                                                    =
                                                                    None)
```

Bases: `abc.ABC`

Base outbound transport class.

collector

Accessor for the stats collector instance.

handle_message (profile: `aries_cloudagent.core.profile.Profile`, payload: `Union[str, bytes]`, endpoint: `str`, metadata: `dict = None`)

Handle message from queue.

Parameters

- **profile** – the profile that produced the message
- **payload** – message payload in string or byte format
- **endpoint** – URI endpoint for delivery
- **metadata** – Additional metadata associated with the payload

start()

Start the transport.

stop()

Shut down the transport.

wire_format

Accessor for a custom wire format for the transport.

```
exception aries_cloudagent.transport.outbound.base.OutboundDeliveryError (*args,
                                                                    er-
                                                                    ror_code:
                                                                    str
                                                                    =
                                                                    None,
                                                                    **kwargs)
```

Bases: `aries_cloudagent.transport.outbound.base.OutboundTransportError`

Base exception when a message cannot be delivered via an outbound transport.

```
exception aries_cloudagent.transport.outbound.base.OutboundTransportError (*args,
                                                                           er-
                                                                           ror_code:
                                                                           str
                                                                           =
                                                                           None,
                                                                           **kwargs)
```

Bases: *aries_cloudagent.transport.error.TransportError*

Generic outbound transport error.

```
exception aries_cloudagent.transport.outbound.base.OutboundTransportRegistrationError (*args,
                                                                                       er-
                                                                                       ror_co
                                                                                       str
                                                                                       =
                                                                                       None,
                                                                                       **kwa
```

Bases: *aries_cloudagent.transport.outbound.base.OutboundTransportError*

Outbound transport registration error.

aries_cloudagent.transport.outbound.http module

Http outbound transport.

```
class aries_cloudagent.transport.outbound.http.HttpTransport
    Bases: aries_cloudagent.transport.outbound.base.BaseOutboundTransport
```

Http outbound transport class.

```
handle_message (profile: aries_cloudagent.core.profile.Profile, payload: Union[str, bytes], endpoint:
                  str, metadata: dict = None, api_key: str = None)
    Handle message from queue.
```

Parameters

- **profile** – the profile that produced the message
- **payload** – message payload in string or byte format
- **endpoint** – URI endpoint for delivery
- **metadata** – Additional metadata associated with the payload

```
schemes = ('http', 'https')
```

```
start ()
    Start the transport.
```

```
stop ()
    Stop the transport.
```

aries_cloudagent.transport.outbound.manager module

Outbound transport manager.

```

class aries_cloudagent.transport.outbound.manager.OutboundTransportManager (context:
    aries_cloudagent.conf
    handle_not_delivered:
    Callable
    =
    None)

```

Bases: `object`

Outbound transport manager class.

MAX_RETRY_COUNT = 4

deliver_queued_message (queued: `aries_cloudagent.transport.outbound.manager.QueuedOutboundMessage`)
→ `_asyncio.Task`
Kick off delivery of a queued message.

encode_queued_message (queued: `aries_cloudagent.transport.outbound.manager.QueuedOutboundMessage`)
→ `_asyncio.Task`
Kick off encoding of a queued message.

enqueue_message (profile: `aries_cloudagent.core.profile.Profile`, outbound: `aries_cloudagent.transport.outbound.message.OutboundMessage`)
Add an outbound message to the queue.

Parameters

- **profile** – The active profile for the request
- **outbound** – The outbound message to deliver

enqueue_webhook (topic: `str`, payload: `dict`, endpoint: `str`, max_attempts: `int` = None, metadata: `dict` = None)
Add a webhook to the queue.

Parameters

- **topic** – The webhook topic
- **payload** – The webhook payload
- **endpoint** – The webhook endpoint
- **max_attempts** – Override the maximum number of attempts
- **metadata** – Additional metadata associated with the payload

Raises `OutboundDeliveryError` – if the associated transport is not running

finished_deliver (queued: `aries_cloudagent.transport.outbound.manager.QueuedOutboundMessage`,
completed: `aries_cloudagent.utils.task_queue.CompletedTask`)
Handle completion of queued message delivery.

finished_encode (queued: `aries_cloudagent.transport.outbound.manager.QueuedOutboundMessage`,
completed: `aries_cloudagent.utils.task_queue.CompletedTask`)
Handle completion of queued message encoding.

flush ()
Wait for any queued messages to be delivered.

get_registered_transport_for_scheme (scheme: `str`) → `str`
Find the registered transport ID for a given scheme.

get_running_transport_for_endpoint (endpoint: `str`)
Find the running transport ID to use for a given endpoint.

get_running_transport_for_scheme (*scheme: str*) → str

Find the running transport ID for a given scheme.

get_transport_instance (*transport_id: str*) → aries_cloudagent.transport.outbound.base.BaseOutboundTransport

Get an instance of a running transport by ID.

perform_encode (*queued: aries_cloudagent.transport.outbound.manager.QueuedOutboundMessage*)

Perform message encoding.

process_queued () → asyncio.Task

Start the process to deliver queued messages if necessary.

Returns: the current queue processing task or None

register (*module: str*) → str

Register a new outbound transport by module path.

Parameters **module** – Module name to register

Raises

- OutboundTransportRegistrationError – If the imported class cannot be located
- OutboundTransportRegistrationError – If the imported class does not specify a schemes attribute
- OutboundTransportRegistrationError – If the scheme has already been registered

register_class (*transport_class: Type[aries_cloudagent.transport.outbound.base.BaseOutboundTransport],
transport_id: str = None*) → str

Register a new outbound transport class.

Parameters **transport_class** – Transport class to register

Raises

- OutboundTransportRegistrationError – If the imported class does not specify a schemes attribute
- OutboundTransportRegistrationError – If the scheme has already been registered

setup ()

Perform setup operations.

start ()

Start all transports and feed messages from the queue.

start_transport (*transport_id: str*)

Start a registered transport.

stop (*wait: bool = True*)

Stop all running transports.


```
class aries_cloudagent.transport.outbound.manager.QueuedOutboundMessage (profile:
    aries_cloudagent.core.profi
    mes-
    sage:
    aries_cloudagent.transport.
    tar-
    get:
    aries_cloudagent.connectio
    trans-
    port_id:
    str)
```

Bases: `object`

Class representing an outbound message pending delivery.

`STATE_DELIVER = 'deliver'`

`STATE_DONE = 'done'`

`STATE_ENCODE = 'encode'`

`STATE_NEW = 'new'`

`STATE_PENDING = 'pending'`

`STATE_RETRY = 'retry'`

aries_cloudagent.transport.outbound.message module

Outbound message representation.

```
class aries_cloudagent.transport.outbound.message.OutboundMessage(*, connection_id:
    str = None,
    enc_payload:
    Union[str,
    bytes] =
    None, endpoint: str
    = None,
    payload:
    Union[str,
    bytes], reply_session_id:
    str =
    None, reply_thread_id:
    str =
    None, reply_to_verkey:
    str =
    None, reply_from_verkey:
    str = None,
    target:
    aries_cloudagent.connections.model.Connection = None,
    target_get_list: Sequence[aries_cloudagent.connections.model.Connection] = None,
    to_session_only:
    bool =
    False)
```

Bases: `object`

Represents an outgoing message.

aries_cloudagent.transport.outbound.ws module

Websockets outbound transport.

```
class aries_cloudagent.transport.outbound.ws.WsTransport
```

Bases: `aries_cloudagent.transport.outbound.base.BaseOutboundTransport`

Websockets outbound transport class.

```
handle_message (profile: aries_cloudagent.core.profile.Profile, payload: Union[str, bytes], endpoint:
    str, metadata: dict = None)
```

Handle message from queue.

Parameters

- **profile** – the profile that produced the message
- **payload** – message payload in string or byte format
- **endpoint** – URI endpoint for delivery

- **metadata** – Additional metadata associated with the payload

schemes = ('ws', 'wss')

start()

Start the outbound transport.

stop()

Stop the outbound transport.

aries_cloudagent.transport.queue package

Submodules

aries_cloudagent.transport.queue.base module

Abstract message queue.

class aries_cloudagent.transport.queue.base.**BaseMessageQueue**

Bases: `abc.ABC`

Abstract message queue class.

dequeue (*, *timeout: int = None*)

Dequeue a message.

Returns The dequeued message, or None if a timeout occurs

Raises

- `asyncio.CancelledError` if the queue has been stopped
- `asyncio.TimeoutError` if the timeout is reached

enqueue (*message*)

Enqueue a message.

Parameters **message** – The message to add to the end of the queue

Raises `asyncio.CancelledError` if the queue has been stopped

join()

Wait for the queue to empty.

reset()

Empty the queue and reset the stop event.

stop()

Cancel active iteration of the queue.

task_done()

Indicate that the current task is complete.

aries_cloudagent.transport.queue.basic module

Basic in memory queue.

class aries_cloudagent.transport.queue.basic.**BasicMessageQueue**

Bases: `aries_cloudagent.transport.queue.base.BaseMessageQueue`

Basic in memory queue implementation class.

dequeue (*, *timeout: int = None*)

Dequeue a message.

Returns The dequeued message, or None if a timeout occurs

Raises

- `asyncio.CancelledError` if the queue has been stopped
- `asyncio.TimeoutError` if the timeout is reached

enqueue (*message*)

Enqueue a message.

Parameters **message** – The message to add to the end of the queue

Raises `asyncio.CancelledError` if the queue has been stopped

join ()

Wait for the queue to empty.

make_queue ()

Create the queue instance.

reset ()

Empty the queue and reset the stop event.

stop ()

Cancel active iteration of the queue.

task_done ()

Indicate that the current task is complete.

Submodules

`aries_cloudagent.transport.error` module

Transport-related error classes and codes.

exception `aries_cloudagent.transport.error.RecipientKeysError` (*args, *error_code: str = None*, **kwargs)

Bases: `aries_cloudagent.transport.error.WireFormatError`

Extract recipient keys error.

exception `aries_cloudagent.transport.error.TransportError` (*args, *error_code: str = None*, **kwargs)

Bases: `aries_cloudagent.core.error.BaseError`

Base class for all transport errors.

exception `aries_cloudagent.transport.error.WireFormatEncodeError` (*args, *error_code: str = None*, **kwargs)

Bases: `aries_cloudagent.transport.error.WireFormatError`

Encoding error when packing the wire format.

error_code = 'message_encode_error'

exception `aries_cloudagent.transport.error.WireFormatError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.transport.error.TransportError`

Base class for wire-format errors.

exception `aries_cloudagent.transport.error.WireFormatParseError` (*args, error_code: str = None, **kwargs)

Bases: `aries_cloudagent.transport.error.WireFormatError`

Parse error when unpacking the wire format.

error_code = 'message_parse_error'

aries_cloudagent.transport.pack_format module

Standard packed message format classes.

class `aries_cloudagent.transport.pack_format.PackWireFormat`
Bases: `aries_cloudagent.transport.wire_format.BaseWireFormat`

Standard DIDComm message parser and serializer.

encode_message (session: `aries_cloudagent.core.profile.ProfileSession`, message_json: Union[str, bytes], recipient_keys: Sequence[str], routing_keys: Sequence[str], sender_key: str) → Union[str, bytes]

Encode an outgoing message for transport.

Parameters

- **session** – The profile session for providing wallet access
- **message_json** – The message body to serialize
- **recipient_keys** – A sequence of recipient verkeys
- **routing_keys** – A sequence of routing verkeys
- **sender_key** – The verification key of the sending agent

Returns The encoded message

Raises `MessageEncodeError` – If the message could not be encoded

get_recipient_keys (message_body: Union[str, bytes]) → List[str]

Get all recipient keys from a wire message.

Parameters **message_body** – The body of the message

Returns List of recipient keys from the message body

Raises `RecipientKeysError` – If the recipient keys could not be extracted

pack (session: `aries_cloudagent.core.profile.ProfileSession`, message_json: Union[str, bytes], recipient_keys: Sequence[str], routing_keys: Sequence[str], sender_key: str)

Look up the wallet instance and perform the message pack.

parse_message (session: `aries_cloudagent.core.profile.ProfileSession`, message_body: Union[str, bytes]) → Tuple[dict, `aries_cloudagent.transport.inbound.receive.MessageReceipt`]

Deserialize an incoming message and further populate the request context.

Parameters

- **session** – The profile session for providing wallet access
- **message_body** – The body of the message

Returns A tuple of the parsed message and a message receipt instance

Raises

- `WireFormatParseError` – If the JSON parsing failed
- `WireFormatParseError` – If a wallet is required but can't be located

unpack (*session*: `aries_cloudagent.core.profile.ProfileSession`, *message_body*: `Union[str, bytes]`, *receipt*: `aries_cloudagent.transport.inbound.receipt.MessageReceipt`)
Look up the wallet instance and perform the message unpack.

`aries_cloudagent.transport.stats` module

aiohttp stats collector support.

```
class aries_cloudagent.transport.stats.StatsTracer (collector:  
                                                    aries_cloudagent.utils.stats.Collector,  
                                                    prefix: str)  
  
Bases: sphinx.ext.autodoc.importer._MockObject  
  
Attach hooks to client session events and report statistics.  
  
connection_queued_end (session, context, params)  
    Handle the end of a queued connection.  
  
connection_queued_start (session, context, params)  
    Handle the start of a queued connection.  
  
connection_ready (session, context, params)  
    Handle the end of connection acquisition.  
  
dns_resolvehost_end (session, context, params)  
    Handle the end of a DNS resolution.  
  
dns_resolvehost_start (session, context, params)  
    Handle the start of a DNS resolution.  
  
request_end (session, context, params)  
    Handle the end of request.  
  
request_start (session, context, params)  
    Handle the start of a request.  
  
socket_connect_start (session, context, params)  
    Handle the start of a socket connection.
```

`aries_cloudagent.transport.wire_format` module

Abstract wire format classes.

```
class aries_cloudagent.transport.wire_format.BaseWireFormat  
    Bases: object  
  
    Abstract messaging wire format.
```

encode_message (*session: aries_cloudagent.core.profile.ProfileSession, message_json: Union[str, bytes], recipient_keys: Sequence[str], routing_keys: Sequence[str], sender_key: str*) → Union[str, bytes]

Encode an outgoing message for transport.

Parameters

- **session** – The profile session for providing wallet access
- **message_json** – The message body to serialize
- **recipient_keys** – A sequence of recipient verkeys
- **routing_keys** – A sequence of routing verkeys
- **sender_key** – The verification key of the sending agent

Returns The encoded message

Raises MessageEncodeError – If the message could not be encoded

get_recipient_keys (*message_body: Union[str, bytes]*) → List[str]

Get all recipient keys from a wire message.

Parameters **message_body** – The body of the message

Returns List of recipient keys from the message body

Raises RecipientKeysError – If the recipient keys could not be extracted

parse_message (*session: aries_cloudagent.core.profile.ProfileSession, message_body: Union[str, bytes]*) → Tuple[dict, aries_cloudagent.transport.inbound.receipt.MessageReceipt]

Deserialize an incoming message and further populate the request context.

Parameters

- **session** – The profile session for providing wallet access
- **message_body** – The body of the message

Returns A tuple of the parsed message and a message receipt instance

Raises WireFormatParseError – If the message can't be parsed

class aries_cloudagent.transport.wire_format.JsonWireFormat
Bases: *aries_cloudagent.transport.wire_format.BaseWireFormat*

Unencrypted wire format.

encode_message (*session: aries_cloudagent.core.profile.ProfileSession, message_json: Union[str, bytes], recipient_keys: Sequence[str], routing_keys: Sequence[str], sender_key: str*) → Union[str, bytes]

Encode an outgoing message for transport.

Parameters

- **session** – The profile session for providing wallet access
- **message_json** – The message body to serialize
- **recipient_keys** – A sequence of recipient verkeys
- **routing_keys** – A sequence of routing verkeys
- **sender_key** – The verification key of the sending agent

Returns The encoded message

Raises MessageEncodeError – If the message could not be encoded

get_recipient_keys (*message_body*: *Union[str, bytes]*) → List[str]

Get all recipient keys from a wire message.

Parameters **message_body** – The body of the message

Returns List of recipient keys from the message body

Raises *RecipientKeysError* – If the recipient keys could not be extracted

parse_message (*session*: *aries_cloudagent.core.profile.ProfileSession*, *message_body*: *Union[str, bytes]*) → Tuple[dict, *aries_cloudagent.transport.inbound.receipt.MessageReceipt*]

Deserialize an incoming message and further populate the request context.

Parameters

- **session** – The profile session for providing wallet access
- **message_body** – The body of the message

Returns A tuple of the parsed message and a message receipt instance

Raises *WireFormatParseError* – If the JSON parsing failed

aries_cloudagent.utils package

Submodules

aries_cloudagent.utils.classloader module

The classloader provides utilities to dynamically load classes and modules.

class *aries_cloudagent.utils.classloader.ClassLoader*

Bases: *object*

Class used to load classes from modules dynamically.

classmethod **load_class** (*class_name*: str, *default_module*: str = None, *package*: str = None)

Resolve a complete class path (ie. typing.Dict) to the class itself.

Parameters

- **class_name** – the class name
- **default_module** – the default module to load, if not part of in the class name
- **package** – the parent package to search for the module

Returns The resolved class

Raises

- *ClassNotFoundError* – If the class could not be resolved at path
- *ModuleLoadError* – If there was an error loading the module

classmethod **load_module** (*mod_path*: str, *package*: str = None) → module

Load a module by its absolute path.

Parameters

- **mod_path** – the absolute or relative module path
- **package** – the parent package to search for the module

Returns The resolved module or *None* if the module cannot be found

Raises `ModuleLoadError` – If there was an error loading the module

classmethod `load_subclass_of` (`base_class`: `Type[CT_co]`, `mod_path`: `str`, `package`: `str` = `None`)

Resolve an implementation of a base path within a module.

Parameters

- **base_class** – the base class being implemented
- **mod_path** – the absolute module path
- **package** – the parent package to search for the module

Returns The resolved class

Raises

- `ClassNotFoundError` – If the module or class implementation could not be found
- `ModuleLoadError` – If there was an error loading the module

classmethod `scan_subpackages` (`package`: `str`) → `Sequence[str]`

Return a list of sub-packages defined under a named package.

exception `aries_cloudagent.utils.classloader.ClassNotFoundError` (`*args`, `error_code`: `str` = `None`, `**kwargs`)

Bases: `aries_cloudagent.core.error.BaseError`

Class not found error.

class `aries_cloudagent.utils.classloader.DeferLoad` (`cls_path`: `str`)

Bases: `object`

Helper to defer loading of a class definition.

resolved

Accessor for the resolved class instance.

exception `aries_cloudagent.utils.classloader.ModuleLoadError` (`*args`, `error_code`: `str` = `None`, `**kwargs`)

Bases: `aries_cloudagent.core.error.BaseError`

Module load error.

aries_cloudagent.utils.http module

HTTP utility methods.

exception `aries_cloudagent.utils.http.FetchError` (`*args`, `error_code`: `str` = `None`, `**kwargs`)

Bases: `aries_cloudagent.core.error.BaseError`

Error raised when an HTTP fetch fails.

exception `aries_cloudagent.utils.http.PutError` (`*args`, `error_code`: `str` = `None`, `**kwargs`)

Bases: `aries_cloudagent.core.error.BaseError`

Error raised when an HTTP put fails.

```
aries_cloudagent.utils.http.fetch(url: str, *, headers: dict = None, retry: bool = True,
                                  max_attempts: int = 5, interval: float = 1.0, back-
                                  off: float = 0.25, request_timeout: float = 10.0, con-
                                  nector: <sphinx.ext.autodoc.importer._MockObject
                                  object at 0x7f760f1b2b50> = None, session:
                                  <sphinx.ext.autodoc.importer._MockObject object at
                                  0x7f760f1b2210> = None, json: bool = False)
```

Fetch from an HTTP server with automatic retries and timeouts.

Parameters

- **url** – the address to fetch
- **headers** – an optional dict of headers to send
- **retry** – flag to retry the fetch
- **max_attempts** – the maximum number of attempts to make
- **interval** – the interval between retries, in seconds
- **backoff** – the backoff interval, in seconds
- **request_timeout** – the HTTP request timeout, in seconds
- **connector** – an optional existing BaseConnector
- **session** – a shared ClientSession
- **json** – flag to parse the result as JSON

```
aries_cloudagent.utils.http.fetch_stream(url: str, *, headers: dict = None, retry:
                                          bool = True, max_attempts: int = 5, in-
                                          terval: float = 1.0, backoff: float = 0.25,
                                          request_timeout: float = 10.0, connector:
                                          <sphinx.ext.autodoc.importer._MockObject
                                          object at 0x7f760f1b2b50> = None, session:
                                          <sphinx.ext.autodoc.importer._MockObject
                                          object at 0x7f760f1b2210> = None)
```

Fetch from an HTTP server with automatic retries and timeouts.

Parameters

- **url** – the address to fetch
- **headers** – an optional dict of headers to send
- **retry** – flag to retry the fetch
- **max_attempts** – the maximum number of attempts to make
- **interval** – the interval between retries, in seconds
- **backoff** – the backoff interval, in seconds
- **request_timeout** – the HTTP request timeout, in seconds
- **connector** – an optional existing BaseConnector
- **session** – a shared ClientSession
- **json** – flag to parse the result as JSON

```
aries_cloudagent.utils.http.put_file(url: str, file_data: dict, extra_data: dict, *, retry: bool
                                     = True, max_attempts: int = 5, interval: float = 1.0,
                                     backoff: float = 0.25, request_timeout: float = 10.0,
                                     connector: <sphinx.ext.autodoc.importer._MockObject
                                     object at 0x7f760f1b2b50> = None, session:
                                     <sphinx.ext.autodoc.importer._MockObject    object
                                     at 0x7f760f1b2210> = None, json: bool = False)
```

Put to HTTP server with automatic retries and timeouts.

Parameters

- **url** – the address to use
- **file_data** – dict with data key and path of file to upload
- **extra_data** – further content to include in data to put
- **headers** – an optional dict of headers to send
- **retry** – flag to retry the fetch
- **max_attempts** – the maximum number of attempts to make
- **interval** – the interval between retries, in seconds
- **backoff** – the backoff interval, in seconds
- **request_timeout** – the HTTP request timeout, in seconds
- **connector** – an optional existing BaseConnector
- **session** – a shared ClientSession
- **json** – flag to parse the result as JSON

aries_cloudagent.utils.outofband module

Utilities for creating out-of-band messages.

```
aries_cloudagent.utils.outofband.serialize_outofband(message:
                                                    aries_cloudagent.messaging.agent_message.AgentMessage,
                                                    did:
                                                    aries_cloudagent.wallet.base.DIDInfo,
                                                    endpoint: str) → str
```

Serialize the agent message as an out-of-band message.

Returns An OOB message in URL format.

aries_cloudagent.utils.repeat module

Utils for repeating tasks.

```
class aries_cloudagent.utils.repeat.RepeatAttempt(seq:
                                                    aries_cloudagent.utils.repeat.RepeatSequence,
                                                    index: int = 1)
```

Bases: `object`

Represents the current iteration in a repeat sequence.

final

Check if this is the last instance in the sequence.

next () → aries_cloudagent.utils.repeat.RepeatAttempt
Get the next attempt instance.

next_interval
Calculate the interval before the next attempt.

timeout (*interval: float = None*)
Create a context manager for timing out an attempt.

class aries_cloudagent.utils.repeat.**RepeatSequence** (*limit: int = 0, interval: float = 0.0, backoff: float = 0.0*)

Bases: `object`

Represents a repetition sequence.

next_interval (*index: int*) → float
Calculate the time before the next attempt.

start () → aries_cloudagent.utils.repeat.RepeatAttempt
Get the first attempt in the sequence.

aries_cloudagent.utils.stats module

Classes for tracking performance and timing.

class aries_cloudagent.utils.stats.**Collector** (*, *enabled: bool = True, log_path: str = None*)

Bases: `object`

Collector for a set of statistics.

enabled
Accessor for the collector's enabled property.

extract (*groups: Sequence[str] = None*) → dict
Extract statistics for a specific set of groups.

log (*name: str, duration: float, start: float = None*)
Log an entry in the statistics if the collector is enabled.

mark (**names*)
Make a custom decorator function for adding to the set of groups.

reset ()
Reset the collector's statistics.

results
Accessor for the current set of collected statistics.

timer (**groups*)
Create a new timer attached to this collector.

wrap (*obj, prop_name: Union[str, Sequence[str]], groups: Sequence[str] = None, *, ignore_missing: bool = False*)
Wrap a method on a class or class instance.

wrap_coro (*fn, groups: Sequence[str]*)
Wrap a coroutine instance to collect timing statistics on execution.

wrap_fn (*fn, groups: Sequence[str]*)
Wrap a function instance to collect timing statistics on execution.

```

class aries_cloudagent.utils.stats.Stats
    Bases: object

    A collection of statistics.

    extract (names: Sequence[str] = None) → dict
        Summarize the stats in a dictionary.

    log (name: str, duration: float)
        Log an entry in the stats.

class aries_cloudagent.utils.stats.Timer(collector: aries_cloudagent.utils.stats.Collector,
                                         groups: Sequence[str])

    Bases: object

    Timer instance for a running task.

    classmethod now()
        Fetch a standard timer value.

    start() → aries_cloudagent.utils.stats.Timer
        Start the timer.

    stop()
        Stop the timer.

```

aries_cloudagent.utils.task_queue module

Classes for managing a set of asyncio tasks.

```

class aries_cloudagent.utils.task_queue.CompletedTask(task: _asyncio.Task, exc_info:
                                                    Tuple, ident: str = None, tim-
                                                    ing: dict = None)

    Bases: object

    Represent the result of a queued task.

class aries_cloudagent.utils.task_queue.PendingTask(coro: Coroutine[T_co, T_contra,
                                                    V_co], complete_hook: Callable
                                                    = None, ident: str = None,
                                                    task_future: _asyncio.Future =
                                                    None, queued_time: float =
                                                    None)

    Bases: object

    Represent a task in the queue.

    cancel()
        Cancel the pending task.

    cancelled
        Accessor for the cancelled property.

    task
        Accessor for the task.

class aries_cloudagent.utils.task_queue.TaskQueue(max_active: int = 0, timed: bool =
                                                    False, trace_fn: Callable = None)

    Bases: object

    A class for managing a set of asyncio tasks.

```

add_active (*task: _asyncio.Task, task_complete: Callable = None, ident: str = None, timing: dict = None*) → *_asyncio.Task*

Register an active async task with an optional completion callback.

Parameters

- **task** – The asyncio task instance
- **task_complete** – An optional callback to run on completion
- **ident** – A string identifier for the task
- **timing** – An optional dictionary of timing information

add_pending (*pending: aries_cloudagent.utils.task_queue.PendingTask*)

Add a task to the pending queue.

Parameters pending – The *PendingTask* to add to the task queue

cancel ()

Cancel any pending or active tasks in the queue.

cancel_pending ()

Cancel any pending tasks in the queue.

cancelled

Accessor for the cancelled property of the queue.

complete (*timeout: float = None, cleanup: bool = True*)

Cancel any pending tasks and wait for, or cancel active tasks.

completed_task (*task: _asyncio.Task, task_complete: Callable, ident: str, timing: dict = None*)

Clean up after a task has completed and run callbacks.

current_active

Accessor for the current number of active tasks in the queue.

current_pending

Accessor for the current number of pending tasks in the queue.

current_size

Accessor for the total number of tasks in the queue.

drain () → *_asyncio.Task*

Start the process to run queued tasks.

flush ()

Wait for any active or pending tasks to be completed.

max_active

Accessor for the maximum number of active tasks in the queue.

put (*coro: Coroutine[T_co, T_contra, V_co], task_complete: Callable = None, ident: str = None*) →

aries_cloudagent.utils.task_queue.PendingTask

Add a new task to the queue, delaying execution if busy.

Parameters

- **coro** – The coroutine to run
- **task_complete** – A callback to run on completion
- **ident** – A string identifier for the task

Returns: a future resolving to the asyncio task instance once queued

ready

Accessor for the ready property of the queue.

run (*coro: Coroutine[T_co, T_contra, V_co]*, *task_complete: Callable = None*, *ident: str = None*, *timing: dict = None*) → `_asyncio.Task`
 Start executing a coroutine as an async task, bypassing the pending queue.

Parameters

- **coro** – The coroutine to run
- **task_complete** – An optional callback to run on completion
- **ident** – A string identifier for the task
- **timing** – An optional dictionary of timing information

Returns: the new asyncio task instance

wait_for (*timeout: float*)

Wait for all queued tasks to complete with a timeout.

`aries_cloudagent.utils.task_queue.coro_ident` (*coro: Coroutine[T_co, T_contra, V_co]*)
 Extract an identifier for a coroutine.

`aries_cloudagent.utils.task_queue.coro_timed` (*coro: Coroutine[T_co, T_contra, V_co]*,
timing: dict)

Capture timing for a coroutine.

`aries_cloudagent.utils.task_queue.task_exc_info` (*task: _asyncio.Task*)
 Extract exception info from an asyncio task.

aries_cloudagent.utils.tracing module

Event tracing.

class `aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchema` (**args*,
***kwargs*)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Request/result schema including agent message tracing.

This is to be used as a superclass for aca-py admin input/output messages that need to support tracing.

trace = `<fields.Boolean(default=False, attribute=None, validate=None, required=False, ...)`

`aries_cloudagent.utils.tracing.decode_inbound_message` (*message*)
 Return bundled message if appropriate.

`aries_cloudagent.utils.tracing.get_timer` () → float
 Return a timer.

`aries_cloudagent.utils.tracing.trace_event` (*context, message, handler: str = None, out-
 come: str = None, perf_counter: float = None,
 force_trace: bool = False, raise_errors: bool
 = False*) → float

Log a trace event to a configured target.

Parameters

- **context** – The application context, attributes of interest are: `context["trace.enabled"]`: True if we are logging events `context["trace.target"]`: Trace target ("log", "message" or an http endpoint) `context["trace.tag"]`: Tag to be included in trace output

- **message** – the current message, can be an AgentMessage, InboundMessage, OutboundMessage or Exchange record
- **event** – Dict that will be converted to json and posted to the target

`aries_cloudagent.utils.tracing.tracing_enabled(context, message) → bool`
Determine whether to log trace messages or not.

aries_cloudagent.wallet package

Abstract and Indy wallet handling.

Subpackages

aries_cloudagent.wallet.models package

Submodules

aries_cloudagent.wallet.models.wallet_record module

Wallet record.

```
class aries_cloudagent.wallet.models.wallet_record.WalletRecord(*, wallet_id:
                                                                str = None,
                                                                key_management_mode:
                                                                str = None,
                                                                settings: dict
                                                                = None, wallet_name:
                                                                str = None,
                                                                **kwargs)
```

Bases: `aries_cloudagent.messaging.models.base_record.BaseRecord`

Represents a wallet record.

MODE_MANAGED = 'managed'

MODE_UNMANAGED = 'unmanaged'

class Meta

Bases: `object`

WalletRecord metadata.

schema_class = 'WalletRecordSchema'

RECORD_ID_NAME = 'wallet_id'

RECORD_TYPE = 'wallet_record'

TAG_NAMES = {'wallet_name'}

is_managed

Accessor to check if the key management mode is managed.

record_value

Accessor for the JSON record value generated for this record.

requires_external_key

Accessor to check if the wallet requires an external key.

settings

Accessor for the context settings associated with this wallet.

update_settings (*settings: dict*)

Update settings.

wallet_dispatch_type

Accessor for webhook dispatch type of the wallet.

wallet_id

Accessor for the ID associated with this record.

wallet_key

Accessor for the key of the wallet.

wallet_name

Accessor for the name of the wallet.

wallet_type

Accessor for the type of the wallet.

wallet_webhook_urls

Accessor for webhook_urls of the wallet.

```
class aries_cloudagent.wallet.models.wallet_record.WalletRecordSchema (*args,
                                                                    **kwargs)
```

Bases: *aries_cloudagent.messaging.models.base_record.BaseRecordSchema*

Schema to allow serialization/deserialization of record.

class Meta

Bases: *object*

WalletRecordSchema metadata.

model_class

alias of *WalletRecord*

unknown

Used by autodoc_mock_imports.

key_management_mode

Used by autodoc_mock_imports.

settings

Used by autodoc_mock_imports.

wallet_id

Used by autodoc_mock_imports.

Submodules

aries_cloudagent.wallet.base module

Wallet base class.

```
class aries_cloudagent.wallet.base.BaseWallet
    Bases: abc.ABC
```

Abstract wallet interface.

create_local_did (*seed: str = None, did: str = None, metadata: dict = None*) → *aries_cloudagent.wallet.base.DIDInfo*

Create and store a new local DID.

Parameters

- **seed** – Optional seed to use for DID
- **did** – The DID to use
- **metadata** – Metadata to store with DID

Returns The created *DIDInfo*

create_public_did (*seed: str = None, did: str = None, metadata: dict = {}*) → *aries_cloudagent.wallet.base.DIDInfo*

Create and store a new public DID.

Implicitly flags all other dids as not public.

Parameters

- **seed** – Optional seed to use for DID
- **did** – The DID to use
- **metadata** – Metadata to store with DID

Returns The created *DIDInfo*

create_signing_key (*seed: str = None, metadata: dict = None*) → *aries_cloudagent.wallet.base.KeyInfo*

Create a new public/private signing keypair.

Parameters

- **seed** – Optional seed allowing deterministic key creation
- **metadata** – Optional metadata to store with the keypair

Returns A *KeyInfo* representing the new record

get_local_did (*did: str*) → *aries_cloudagent.wallet.base.DIDInfo*

Find info for a local DID.

Parameters **did** – The DID for which to get info

Returns A *DIDInfo* instance for the DID

get_local_did_for_verkey (*verkey: str*) → *aries_cloudagent.wallet.base.DIDInfo*

Resolve a local DID from a verkey.

Parameters **verkey** – Verkey for which to get DID info

Returns A *DIDInfo* instance for the DID

get_local_dids () → *Sequence[aries_cloudagent.wallet.base.DIDInfo]*

Get list of defined local DIDs.

Returns A list of *DIDInfo* instances

get_posted_dids () → *Sequence[aries_cloudagent.wallet.base.DIDInfo]*

Get list of defined posted DIDs, excluding public DID.

Returns A list of *DIDInfo* instances

get_public_did() → aries_cloudagent.wallet.base.DIDInfo
Retrieve the public DID.

Returns The created *DIDInfo*

get_signing_key(*verkey: str*) → aries_cloudagent.wallet.base.KeyInfo
Fetch info for a signing keypair.

Parameters **verkey** – The verification key of the keypair

Returns A *KeyInfo* representing the keypair

pack_message(*message: str, to_verkeys: Sequence[str], from_verkey: str = None*) → bytes
Pack a message for one or more recipients.

Parameters

- **message** – The message to pack
- **to_verkeys** – The verkeys to pack the message for
- **from_verkey** – The sender verkey

Returns The packed message

replace_local_did_metadata(*did: str, metadata: dict*)
Replace the metadata associated with a local DID.

Prefer *set_did_endpoint()* to set endpoint in metadata.

Parameters

- **did** – DID for which to replace metadata
- **metadata** – The new metadata

replace_signing_key_metadata(*verkey: str, metadata: dict*)
Replace the metadata associated with a signing keypair.

Parameters

- **verkey** – The verification key of the keypair
- **metadata** – The new metadata to store

rotate_did_keypair_apply(*did: str*) → None
Apply temporary keypair as main for DID that wallet owns.

Parameters **did** – signing DID

Raises

- *WalletNotFoundError* – if wallet does not own DID
- *WalletError* – if wallet has not started key rotation

rotate_did_keypair_start(*did: str, next_seed: str = None*) → str
Begin key rotation for DID that wallet owns: generate new keypair.

Parameters

- **did** – signing DID
- **next_seed** – seed for incoming ed25519 key pair (default random)

Returns The new verification key

Raises *WalletNotFoundError* – if wallet does not own DID

set_did_endpoint (*did: str, endpoint: str, ledger: aries_cloudagent.ledger.base.BaseLedger, endpoint_type: aries_cloudagent.ledger.endpoint_type.EndpointType = None*)
 Update the endpoint for a DID in the wallet, send to ledger if public or posted.

Parameters

- **did** – DID for which to set endpoint
- **endpoint** – the endpoint to set, None to clear
- **ledger** – the ledger to which to send endpoint update if DID is public or posted
- **endpoint_type** – the type of the endpoint/service. Only endpoint_type 'endpoint' affects local wallet

set_public_did (*did: str*) → aries_cloudagent.wallet.base.DIDInfo
 Assign the public DID.

Returns The created *DIDInfo*

sign_message (*message: bytes, from_verkey: str*) → bytes
 Sign a message using the private key associated with a given verkey.

Parameters

- **message** – The message to sign
- **from_verkey** – Sign using the private key related to this verkey

Returns The signature

unpack_message (*enc_message: bytes*) -> (<class 'str'>, <class 'str'>, <class 'str'>)
 Unpack a message.

Parameters **enc_message** – The encrypted message

Returns (message, from_verkey, to_verkey)

Return type A tuple

verify_message (*message: bytes, signature: bytes, from_verkey: str*) → bool
 Verify a signature against the public key of the signer.

Parameters

- **message** – The message to verify
- **signature** – The signature to verify
- **from_verkey** – Verkey to use in verification

Returns True if verified, else False

class aries_cloudagent.wallet.base.**DIDInfo** (*did, verkey, metadata*)
 Bases: `tuple`

did

Alias for field number 0

metadata

Alias for field number 2

verkey

Alias for field number 1

class aries_cloudagent.wallet.base.**KeyInfo** (*verkey, metadata*)
 Bases: `tuple`

metadata

Alias for field number 1

verkey

Alias for field number 0

aries_cloudagent.wallet.crypto module

Cryptography functions used by BasicWallet.

```
class aries_cloudagent.wallet.crypto.PackMessageSchema (*args, **kwargs)
```

Bases: sphinx.ext.autodoc.importer._MockObject

Packed message schema.

ciphertext

Used by autodoc_mock_imports.

iv

Used by autodoc_mock_imports.

protected

Used by autodoc_mock_imports.

tag

Used by autodoc_mock_imports.

```
class aries_cloudagent.wallet.crypto.PackRecipientHeaderSchema (*args,  
                                                                **kwargs)
```

Bases: sphinx.ext.autodoc.importer._MockObject

Packed recipient header schema.

iv

Used by autodoc_mock_imports.

kid

Used by autodoc_mock_imports.

sender

Used by autodoc_mock_imports.

```
class aries_cloudagent.wallet.crypto.PackRecipientSchema (*args, **kwargs)
```

Bases: sphinx.ext.autodoc.importer._MockObject

Packed recipient schema.

encrypted_key

Used by autodoc_mock_imports.

header

Used by autodoc_mock_imports.

```
class aries_cloudagent.wallet.crypto.PackRecipientsSchema (*args, **kwargs)
```

Bases: sphinx.ext.autodoc.importer._MockObject

Packed recipients schema.

alg

Used by autodoc_mock_imports.

enc

Used by autodoc_mock_imports.

recipients

Used by `autodoc_mock_imports`.

typ

Used by `autodoc_mock_imports`.

`aries_cloudagent.wallet.crypto.create_keypair` (*seed: bytes = None*) → Tuple[bytes, bytes]

Create a public and private signing keypair from a seed value.

Parameters *seed* – Seed for keypair

Returns A tuple of (public key, secret key)

`aries_cloudagent.wallet.crypto.decode_pack_message` (*enc_message: bytes, find_key: Callable*) → Tuple[str, Optional[str], str]

Decode a packed message.

Disassemble and unencrypt a packed message, returning the message content, verification key of the sender (if available), and verification key of the recipient.

Parameters

- **enc_message** – The encrypted message
- **find_key** – Function to retrieve private key

Returns A tuple of (message, sender_vk, recip_vk)

Raises

- `ValueError` – If the packed message is invalid
- `ValueError` – If the packed message recipients are invalid
- `ValueError` – If the pack algorithm is unsupported
- `ValueError` – If the sender's public key was not provided

`aries_cloudagent.wallet.crypto.decode_pack_message_outer` (*enc_message: bytes*) → Tuple[dict, dict, bool]

Decode the outer wrapper of a packed message and extract the recipients.

Parameters *enc_message* – The encrypted message

Returns: a tuple of the decoded wrapper, recipients, and authcrypt flag

`aries_cloudagent.wallet.crypto.decode_pack_message_payload` (*wrapper: dict, payload_key: bytes*) → str

Decode the payload of a packed message once the CEK is known.

Parameters

- **wrapper** – The decoded message wrapper
- **payload_key** – The decrypted payload key

`aries_cloudagent.wallet.crypto.decrypt_plaintext` (*ciphertext: bytes, recips_bin: bytes, nonce: bytes, key: bytes*) → str

Decrypt the payload of a packed message.

Parameters

- **ciphertext** –
- **recips_bin** –

- **nonce** –
- **key** –

Returns The decrypted string

`aries_cloudagent.wallet.crypto.encode_pack_message` (*message: str, to_verkeys: Sequence[bytes], from_secret: bytes = None*) → bytes

Assemble a packed message for a set of recipients, optionally including the sender.

Parameters

- **message** – The message to pack
- **to_verkeys** – The verkeys to pack the message for
- **from_secret** – The sender secret

Returns The encoded message

`aries_cloudagent.wallet.crypto.encrypt_plaintext` (*message: str, add_data: bytes, key: bytes*) → Tuple[bytes, bytes, bytes]

Encrypt the payload of a packed message.

Parameters

- **message** – Message to encrypt
- **add_data** –
- **key** – Key used for encryption

Returns A tuple of (ciphertext, nonce, tag)

`aries_cloudagent.wallet.crypto.extract_pack_recipients` (*recipients: Sequence[dict]*) → dict

Extract the pack message recipients into a dict indexed by verkey.

Parameters **recipients** – Recipients to locate

Raises `ValueError` – If the recipients block is mal-formatted

`aries_cloudagent.wallet.crypto.extract_payload_key` (*sender_cek: dict, recip_secret: bytes*) → Tuple[bytes, str]

Extract the payload key from pack recipient details.

Returns: A tuple of the CEK and sender verkey

`aries_cloudagent.wallet.crypto.prepare_pack_recipient_keys` (*to_verkeys: Sequence[bytes], from_secret: bytes = None*) → Tuple[str, bytes]

Assemble the recipients block of a packed message.

Parameters

- **to_verkeys** – Verkeys of recipients
- **from_secret** – Secret to use for signing keys

Returns A tuple of (json result, key)

`aries_cloudagent.wallet.crypto.random_seed`() → bytes

Generate a random seed value.

Returns A new random seed

`aries_cloudagent.wallet.crypto.seed_to_did(seed: str) → str`
Derive a DID from a seed value.

Parameters `seed` – The seed to derive

Returns The DID derived from the seed

`aries_cloudagent.wallet.crypto.sign_message(message: bytes, secret: bytes) → bytes`
Sign a message using a private signing key.

Parameters

- **message** – The message to sign
- **secret** – The private signing key

Returns The signature

`aries_cloudagent.wallet.crypto.sign_pk_from_sk(secret: bytes) → bytes`
Extract the verkey from a secret signing key.

`aries_cloudagent.wallet.crypto.validate_seed(seed: (<class 'str'>, <class 'bytes'>)) → bytes`
Convert a seed parameter to standard format and check length.

Parameters `seed` – The seed to validate

Returns The validated and encoded seed

`aries_cloudagent.wallet.crypto.verify_signed_message(signed: bytes, verkey: bytes) → bool`
Verify a signed message according to a public verification key.

Parameters

- **signed** – The signed message
- **verkey** – The verkey to use in verification

Returns True if verified, else False

aries_cloudagent.wallet.did_posture module

Ledger utilities.

class `aries_cloudagent.wallet.did_posture.DIDPosture`

Bases: `enum.Enum`

Enum for DID postures: public, posted but not public, or in wallet only.

POSTED = `DIDPostureSpec(moniker='posted', ordinal=1, public=False, posted=True)`

PUBLIC = `DIDPostureSpec(moniker='public', ordinal=0, public=True, posted=True)`

WALLET_ONLY = `DIDPostureSpec(moniker='wallet_only', ordinal=2, public=False, posted=False)`

get = `<function DIDPosture.get>`

metadata

DID metadata for DID posture.

moniker

Name for DID posture.

ordinal

public first, then posted and wallet-only.

Type Ordinal for presentation

```
class aries_cloudagent.wallet.did_posture.DIDPostureSpec(moniker, ordinal, public,  
                                                    posted)
```

Bases: `tuple`

moniker

Alias for field number 0

ordinal

Alias for field number 1

posted

Alias for field number 3

public

Alias for field number 2

aries_cloudagent.wallet.error module

Wallet-related exceptions.

```
exception aries_cloudagent.wallet.error.WalletDuplicateError(*args, error_code:  
                                                             str = None,  
                                                             **kwargs)
```

Bases: `aries_cloudagent.wallet.error.WalletError`

Duplicate record exception.

```
exception aries_cloudagent.wallet.error.WalletError(*args, error_code: str = None,  
                                                    **kwargs)
```

Bases: `aries_cloudagent.core.error.BaseError`

General wallet exception.

```
exception aries_cloudagent.wallet.error.WalletNotFoundError(*args, error_code:  
                                                             str = None,  
                                                             **kwargs)
```

Bases: `aries_cloudagent.wallet.error.WalletError`

Record not found exception.

```
exception aries_cloudagent.wallet.error.WalletSettingsError(*args, error_code:  
                                                             str = None,  
                                                             **kwargs)
```

Bases: `aries_cloudagent.wallet.error.WalletError`

Invalid settings exception.

aries_cloudagent.wallet.in_memory module

In-memory implementation of BaseWallet interface.

```
class aries_cloudagent.wallet.in_memory.InMemoryWallet(profile:  
                                                    aries_cloudagent.core.in_memory.InMemoryProfile)
```

Bases: `aries_cloudagent.wallet.base.BaseWallet`

In-memory wallet implementation.

create_local_did (*seed: str = None, did: str = None, metadata: dict = None*) → *aries_cloudagent.wallet.base.DIDInfo*
Create and store a new local DID.

Parameters

- **seed** – Optional seed to use for DID
- **did** – The DID to use
- **metadata** – Metadata to store with DID

Returns A *DIDInfo* instance representing the created DID

Raises *WalletDuplicateError* – If the DID already exists in the wallet

create_signing_key (*seed: str = None, metadata: dict = None*) → *aries_cloudagent.wallet.base.KeyInfo*
Create a new public/private signing keypair.

Parameters

- **seed** – Seed to use for signing key
- **metadata** – Optional metadata to store with the keypair

Returns A *KeyInfo* representing the new record

Raises *WalletDuplicateError* – If the resulting verkey already exists in the wallet

get_local_did (*did: str*) → *aries_cloudagent.wallet.base.DIDInfo*
Find info for a local DID.

Parameters **did** – The DID for which to get info

Returns A *DIDInfo* instance representing the found DID

Raises *WalletNotFoundError* – If the DID is not found

get_local_did_for_verkey (*verkey: str*) → *aries_cloudagent.wallet.base.DIDInfo*
Resolve a local DID from a verkey.

Parameters **verkey** – The verkey for which to get the local DID

Returns A *DIDInfo* instance representing the found DID

Raises *WalletNotFoundError* – If the verkey is not found

get_local_dids () → *Sequence[aries_cloudagent.wallet.base.DIDInfo]*
Get list of defined local DIDs.

Returns A list of locally stored DIDs as *DIDInfo* instances

get_signing_key (*verkey: str*) → *aries_cloudagent.wallet.base.KeyInfo*
Fetch info for a signing keypair.

Parameters **verkey** – The verification key of the keypair

Returns A *KeyInfo* representing the keypair

Raises *WalletNotFoundError* – if no keypair is associated with the verification key

pack_message (*message: str, to_verkeys: Sequence[str], from_verkey: str = None*) → *bytes*
Pack a message for one or more recipients.

Parameters

- **message** – The message to pack

- **to_verkeys** – List of verkeys for which to pack
- **from_verkey** – Sender verkey from which to pack

Returns The resulting packed message bytes

Raises `WalletError` – If the message is not provided

replace_local_did_metadata (*did: str, metadata: dict*)

Replace metadata for a local DID.

Parameters

- **did** – The DID for which to replace metadata
- **metadata** – The new metadata

Raises `WalletNotFoundError` – If the DID doesn't exist

replace_signing_key_metadata (*verkey: str, metadata: dict*)

Replace the metadata associated with a signing keypair.

Parameters

- **verkey** – The verification key of the keypair
- **metadata** – The new metadata to store

Raises `WalletNotFoundError` – if no keypair is associated with the verification key

rotate_did_keypair_apply (*did: str*) → None

Apply temporary keypair as main for DID that wallet owns.

Parameters **did** – signing DID

Raises

- `WalletNotFoundError` – if wallet does not own DID
- `WalletError` – if wallet has not started key rotation

rotate_did_keypair_start (*did: str, next_seed: str = None*) → str

Begin key rotation for DID that wallet owns: generate new keypair.

Parameters

- **did** – signing DID
- **next_seed** – incoming replacement seed (default random)

Returns The new verification key

Raises `WalletNotFoundError` – if wallet does not own DID

sign_message (*message: bytes, from_verkey: str*) → bytes

Sign a message using the private key associated with a given verkey.

Parameters

- **message** – Message bytes to sign
- **from_verkey** – The verkey to use to sign

Returns A signature

Raises

- `WalletError` – If the message is not provided
- `WalletError` – If the verkey is not provided

unpack_message (*enc_message: bytes*) -> (<class 'str'>, <class 'str'>, <class 'str'>)

Unpack a message.

Parameters **enc_message** – The packed message bytes

Returns (message, from_verkey, to_verkey)

Return type A tuple

Raises

- `WalletError` – If the message is not provided
- `WalletError` – If there is a problem unpacking the message

verify_message (*message: bytes, signature: bytes, from_verkey: str*) → bool

Verify a signature against the public key of the signer.

Parameters

- **message** – Message to verify
- **signature** – Signature to verify
- **from_verkey** – Verkey to use in verification

Returns True if verified, else False

Raises

- `WalletError` – If the verkey is not provided
- `WalletError` – If the signature is not provided
- `WalletError` – If the message is not provided

aries_cloudagent.wallet.indy module

Indy implementation of BaseWallet interface.

class `aries_cloudagent.wallet.indy.IndySdkWallet` (*opened: aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet*)

Bases: `aries_cloudagent.wallet.base.BaseWallet`

Indy identity wallet implementation.

create_local_did (*seed: str = None, did: str = None, metadata: dict = None*) → `aries_cloudagent.wallet.base.DIDInfo`

Create and store a new local DID.

Parameters

- **seed** – Optional seed to use for DID
- **did** – The DID to use
- **metadata** – Metadata to store with DID

Returns A *DIDInfo* instance representing the created DID

Raises

- `WalletDuplicateError` – If the DID already exists in the wallet
- `WalletError` – If there is a libindy error

create_signing_key (*seed: str = None, metadata: dict = None*) → *aries_cloudagent.wallet.base.KeyInfo*
 Create a new public/private signing keypair.

Parameters

- **seed** – Seed for key
- **metadata** – Optional metadata to store with the keypair

Returns A *KeyInfo* representing the new record

Raises

- *WalletDuplicateError* – If the resulting verkey already exists in the wallet
- *WalletError* – If there is a libindy error

classmethod generate_wallet_key (*seed: str = None*) → *str*
 Generate a raw Indy wallet key.

get_local_did (*did: str*) → *aries_cloudagent.wallet.base.DIDInfo*
 Find info for a local DID.

Parameters **did** – The DID for which to get info

Returns A *DIDInfo* instance representing the found DID

Raises

- *WalletNotFoundError* – If the DID is not found
- *WalletError* – If there is a libindy error

get_local_did_for_verkey (*verkey: str*) → *aries_cloudagent.wallet.base.DIDInfo*
 Resolve a local DID from a verkey.

Parameters **verkey** – The verkey for which to get the local DID

Returns A *DIDInfo* instance representing the found DID

Raises *WalletNotFoundError* – If the verkey is not found

get_local_dids () → *Sequence[aries_cloudagent.wallet.base.DIDInfo]*
 Get list of defined local DIDs.

Returns A list of locally stored DIDs as *DIDInfo* instances

get_signing_key (*verkey: str*) → *aries_cloudagent.wallet.base.KeyInfo*
 Fetch info for a signing keypair.

Parameters **verkey** – The verification key of the keypair

Returns A *KeyInfo* representing the keypair

Raises

- *WalletNotFoundError* – If no keypair is associated with the verification key
- *WalletError* – If there is a libindy error

pack_message (*message: str, to_verkeys: Sequence[str], from_verkey: str = None*) → *bytes*
 Pack a message for one or more recipients.

Parameters

- **message** – The message to pack
- **to_verkeys** – List of verkeys for which to pack

- **from_verkey** – Sender verkey from which to pack

Returns The resulting packed message bytes

Raises

- `WalletError` – If no message is provided
- `WalletError` – If a libindy error occurs

replace_local_did_metadata (*did: str, metadata: dict*)

Replace metadata for a local DID.

Parameters

- **did** – The DID for which to replace metadata
- **metadata** – The new metadata

replace_signing_key_metadata (*verkey: str, metadata: dict*)

Replace the metadata associated with a signing keypair.

Parameters

- **verkey** – The verification key of the keypair
- **metadata** – The new metadata to store

Raises `WalletNotFoundError` – if no keypair is associated with the verification key

rotate_did_keypair_apply (*did: str*) → `aries_cloudagent.wallet.base.DIDInfo`

Apply temporary keypair as main for DID that wallet owns.

Parameters **did** – signing DID

Returns `DIDInfo` with new verification key and metadata for DID

rotate_did_keypair_start (*did: str, next_seed: str = None*) → `str`

Begin key rotation for DID that wallet owns: generate new keypair.

Parameters

- **did** – signing DID
- **next_seed** – incoming replacement seed (default random)

Returns The new verification key

set_did_endpoint (*did: str, endpoint: str, ledger: aries_cloudagent.ledger.base.BaseLedger, endpoint_type: aries_cloudagent.ledger.endpoint_type.EndpointType = None*)

Update the endpoint for a DID in the wallet, send to ledger if public or posted.

Parameters

- **did** – DID for which to set endpoint
- **endpoint** – the endpoint to set, `None` to clear
- **ledger** – the ledger to which to send endpoint update if DID is public or posted
- **endpoint_type** – the type of the endpoint/service. Only `endpoint_type` ‘endpoint’ affects local wallet

sign_message (*message: bytes, from_verkey: str*) → `bytes`

Sign a message using the private key associated with a given verkey.

Parameters

- **message** – Message bytes to sign

- **from_verkey** – The verkey to use to sign

Returns A signature

Raises

- `WalletError` – If the message is not provided
- `WalletError` – If the verkey is not provided
- `WalletError` – If a libindy error occurs

unpack_message (*enc_message: bytes*) -> (<class 'str'>, <class 'str'>, <class 'str'>)

Unpack a message.

Parameters **enc_message** – The packed message bytes

Returns (message, from_verkey, to_verkey)

Return type A tuple

Raises

- `WalletError` – If the message is not provided
- `WalletError` – If a libindy error occurs

verify_message (*message: bytes, signature: bytes, from_verkey: str*) → bool

Verify a signature against the public key of the signer.

Parameters

- **message** – Message to verify
- **signature** – Signature to verify
- **from_verkey** – Verkey to use in verification

Returns True if verified, else False

Raises

- `WalletError` – If the verkey is not provided
- `WalletError` – If the signature is not provided
- `WalletError` – If the message is not provided
- `WalletError` – If a libindy error occurs

aries_cloudagent.wallet.routes module

Wallet admin routes.

class `aries_cloudagent.wallet.routes.CredDefIdMatchInfoSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Path parameters and validators for request taking credential definition id.

cred_def_id = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<

class `aries_cloudagent.wallet.routes.DIDEndpointSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Request schema to set DID endpoint; response schema to get DID endpoint.

did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl

```
    endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ari
```

class `aries_cloudagent.wallet.routes.DIDEndpointWithTypeSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Request schema to set DID endpoint of particular type.

```
    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    endpoint = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<ari
    endpoint_type = <fields.String(default=<marshmallow.missing>, attribute=None, validate=
```

class `aries_cloudagent.wallet.routes.DIDListQueryStringSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Parameters and validators for DID list request query string.

```
    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    posture = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<arie
    verkey = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
```

class `aries_cloudagent.wallet.routes.DIDListSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Result schema for connection list.

```
    results = <fields.List(default=<marshmallow.missing>, attribute=None, validate=None, r
```

class `aries_cloudagent.wallet.routes.DIDQueryStringSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Parameters and validators for set public DID request query string.

```
    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
```

class `aries_cloudagent.wallet.routes.DIDResultSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Result schema for a DID.

```
    result = <fields.Nested(default=<marshmallow.missing>, attribute=None, validate=None, :
```

class `aries_cloudagent.wallet.routes.DIDSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Result schema for a DID.

```
    did = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_cl
    posture = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<arie
    verkey = <fields.String(default=<marshmallow.missing>, attribute=None, validate=<aries_
```

class `aries_cloudagent.wallet.routes.WalletModuleResponseSchema` (*args, **kwargs)

Bases: `aries_cloudagent.messaging.models.openapi.OpenAPISchema`

Response schema for Wallet Module.

`aries_cloudagent.wallet.routes.format_did_info` (info: `aries_cloudagent.wallet.base.DIDInfo`)

Serialize a DIDInfo object.

`aries_cloudagent.wallet.routes.post_process_routes` (*app:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Amend swagger API.

`aries_cloudagent.wallet.routes.register` (*app:* *<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Register routes.

`aries_cloudagent.wallet.routes.wallet_create_did` (*request:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Request handler for creating a new local DID in the wallet.

Parameters `request` – aiohttp request object

Returns The DID info

`aries_cloudagent.wallet.routes.wallet_did_list` (*request:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Request handler for searching wallet DIDs.

Parameters `request` – aiohttp request object

Returns The DID list response

`aries_cloudagent.wallet.routes.wallet_get_did_endpoint` (*request:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Request handler for getting the current DID endpoint from the wallet.

Parameters `request` – aiohttp request object

Returns The updated DID info

`aries_cloudagent.wallet.routes.wallet_get_public_did` (*request:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Request handler for fetching the current public DID.

Parameters `request` – aiohttp request object

Returns The DID info

`aries_cloudagent.wallet.routes.wallet_rotate_did_keypair` (*request:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Request handler for rotating local DID keypair.

Parameters `request` – aiohttp request object

Returns An empty JSON response

`aries_cloudagent.wallet.routes.wallet_set_did_endpoint` (*request:*
*<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>*)

Request handler for setting an endpoint for a DID.

Parameters `request` – aiohttp request object

`aries_cloudagent.wallet.routes.wallet_set_public_did` (*request:*
<sphinx.ext.autodoc.importer._MockObject
object at 0x7f760dab8110>)

Request handler for setting the current public DID.

Parameters `request` – aiohttp request object

Returns The updated DID info

`aries_cloudagent.wallet.util` module

Wallet utility functions.

`aries_cloudagent.wallet.util.b58_to_bytes` (*val: str*) → bytes

Convert a base 58 string to bytes.

`aries_cloudagent.wallet.util.b64_to_bytes` (*val: str, urlsafe=False*) → bytes

Convert a base 64 string to bytes.

`aries_cloudagent.wallet.util.b64_to_str` (*val: str, urlsafe=False, encoding=None*) → str

Convert a base 64 string to string on input encoding (default utf-8).

`aries_cloudagent.wallet.util.bytes_to_b58` (*val: bytes*) → str

Convert a byte string to base 58.

`aries_cloudagent.wallet.util.bytes_to_b64` (*val: bytes, urlsafe=False, pad=True*) → str

Convert a byte string to base 64.

`aries_cloudagent.wallet.util.did_key_to_naked` (*did_key: str*) → str

Convert a W3C did:key to naked ed25519 verkey format.

`aries_cloudagent.wallet.util.full_verkey` (*did: str, abbr_verkey: str*) → str

Given a DID and a short verkey, return the full verkey.

`aries_cloudagent.wallet.util.naked_to_did_key` (*key: str*) → str

Convert a naked ed25519 verkey to W3C did:key format.

`aries_cloudagent.wallet.util.pad` (*val: str*) → str

Pad base64 values if need be: JWT calls to omit trailing padding.

`aries_cloudagent.wallet.util.set_urlsafe_b64` (*val: str, urlsafe: bool = True*) → str

Set URL safety in base64 encoding.

`aries_cloudagent.wallet.util.str_to_b64` (*val: str, urlsafe=False, encoding=None,*
pad=True) → str

Convert a string to base64 string on input encoding (default utf-8).

`aries_cloudagent.wallet.util.unpad` (*val: str*) → str

Remove padding from base64 values if need be.

1.1.2 Submodules

1.1.3 `aries_cloudagent.version` module

Library version information.

CHAPTER 2

Indices and tables

- `genindex`

a

aries_cloudagent, 3
aries_cloudagent.admin, 3
aries_cloudagent.admin.base_server, 3
aries_cloudagent.admin.error, 4
aries_cloudagent.admin.request_context, 4
aries_cloudagent.admin.server, 5
aries_cloudagent.cache, 8
aries_cloudagent.cache.base, 8
aries_cloudagent.cache.in_memory, 9
aries_cloudagent.commands, 10
aries_cloudagent.commands.help, 10
aries_cloudagent.commands.provision, 10
aries_cloudagent.commands.start, 11
aries_cloudagent.config, 11
aries_cloudagent.config.argparse, 11
aries_cloudagent.config.base, 14
aries_cloudagent.config.base_context, 16
aries_cloudagent.config.default_context, 16
aries_cloudagent.config.error, 16
aries_cloudagent.config.injection_context, 17
aries_cloudagent.config.injector, 18
aries_cloudagent.config.ledger, 19
aries_cloudagent.config.logging, 19
aries_cloudagent.config.provider, 20
aries_cloudagent.config.settings, 21
aries_cloudagent.config.util, 21
aries_cloudagent.config.wallet, 22
aries_cloudagent.connections, 22
aries_cloudagent.connections.base_manager, 37
aries_cloudagent.connections.models, 22
aries_cloudagent.connections.models.connection_record, 30
aries_cloudagent.connections.models.connection_target, 36
aries_cloudagent.connections.models.diddoc, 22
aries_cloudagent.connections.models.diddoc.diddoc, 25
aries_cloudagent.connections.models.diddoc.publickey, 27
aries_cloudagent.connections.models.diddoc.service, 28
aries_cloudagent.connections.models.diddoc.util, 29
aries_cloudagent.connections.util, 38
aries_cloudagent.core, 39
aries_cloudagent.core.conductor, 39
aries_cloudagent.core.dispatcher, 40
aries_cloudagent.core.error, 42
aries_cloudagent.core.in_memory, 43
aries_cloudagent.core.plugin_registry, 45
aries_cloudagent.core.profile, 45
aries_cloudagent.core.protocol_registry, 47
aries_cloudagent.holder, 48
aries_cloudagent.holder.routes, 48
aries_cloudagent.indy, 50
aries_cloudagent.indy.holder, 58
aries_cloudagent.indy.issuer, 60
aries_cloudagent.indy.sdk, 50
aries_cloudagent.indy.sdk.error, 50
aries_cloudagent.indy.sdk.holder, 51
aries_cloudagent.indy.sdk.issuer, 53
aries_cloudagent.indy.sdk.profile, 55
aries_cloudagent.indy.sdk.util, 56
aries_cloudagent.indy.sdk.verifier, 56
aries_cloudagent.indy.sdk.wallet_plugin, 57
aries_cloudagent.indy.sdk.wallet_setup, 57
aries_cloudagent.indy.util, 62
aries_cloudagent.indy.verifier, 63

```

aries_cloudagent.ledger, 63
aries_cloudagent.ledger.base, 63
aries_cloudagent.ledger.endpoint_type, 66
aries_cloudagent.ledger.error, 66
aries_cloudagent.ledger.indy, 67
aries_cloudagent.ledger.routes, 71
aries_cloudagent.ledger.util, 73
aries_cloudagent.messaging, 73
aries_cloudagent.messaging.ack, 73
aries_cloudagent.messaging.ack.message, 73
aries_cloudagent.messaging.agent_message, 106
aries_cloudagent.messaging.base_handler, 109
aries_cloudagent.messaging.credential_definition, 74
aries_cloudagent.messaging.credential_definition, 74
aries_cloudagent.messaging.credential_definition, 76
aries_cloudagent.messaging.decorators, 76
aries_cloudagent.messaging.decorators.attach_decorator, 76
aries_cloudagent.messaging.decorators.base, 83
aries_cloudagent.messaging.decorators.definition, 84
aries_cloudagent.messaging.decorators.localization_decorator, 84
aries_cloudagent.messaging.decorators.placeholder_decorator, 86
aries_cloudagent.messaging.decorators.signature_decorator, 86
aries_cloudagent.messaging.decorators.thread_decorator, 88
aries_cloudagent.messaging.decorators.timing_decorator, 89
aries_cloudagent.messaging.decorators.transaction_decorator, 91
aries_cloudagent.messaging.decorators.transport_decorator, 95
aries_cloudagent.messaging.error, 110
aries_cloudagent.messaging.jsonld, 96
aries_cloudagent.messaging.jsonld.create_verified_data, 96
aries_cloudagent.messaging.jsonld.credentials, 96
aries_cloudagent.messaging.jsonld.routes, 97
aries_cloudagent.messaging.models, 98
aries_cloudagent.messaging.models.base, 98
aries_cloudagent.messaging.models.base_record, 100
aries_cloudagent.messaging.models.openapi, 104
aries_cloudagent.messaging.request_context, 110
aries_cloudagent.messaging.responder, 112
aries_cloudagent.messaging.schemas, 104
aries_cloudagent.messaging.schemas.routes, 104
aries_cloudagent.messaging.schemas.util, 106
aries_cloudagent.messaging.util, 113
aries_cloudagent.messaging.valid, 114
aries_cloudagent.multitenant, 117
aries_cloudagent.multitenant.admin, 117
aries_cloudagent.multitenant.admin.routes, 118
aries_cloudagent.multitenant.error, 120
aries_cloudagent.multitenant.manager, 121
aries_cloudagent.protocols, 123
aries_cloudagent.protocols.actionmenu, 123
aries_cloudagent.protocols.actionmenu.definition, 134
aries_cloudagent.protocols.actionmenu.v1_0, 123
aries_cloudagent.protocols.actionmenu.v1_0.base_service, 131
aries_cloudagent.protocols.actionmenu.v1_0.controller, 131
aries_cloudagent.protocols.actionmenu.v1_0.driver, 131
aries_cloudagent.protocols.actionmenu.v1_0.handlers, 123
aries_cloudagent.protocols.actionmenu.v1_0.handlers, 123
aries_cloudagent.protocols.actionmenu.v1_0.handlers, 124
aries_cloudagent.protocols.actionmenu.v1_0.handlers, 124
aries_cloudagent.protocols.actionmenu.v1_0.message, 132
aries_cloudagent.protocols.actionmenu.v1_0.messages, 124
aries_cloudagent.protocols.actionmenu.v1_0.messages, 124
aries_cloudagent.protocols.actionmenu.v1_0.messages, 125
aries_cloudagent.protocols.actionmenu.v1_0.messages, 126

```



```

aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols_key_id_discovery_v1_0.handlers,
164 188
aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols_key_id_discovery_v1_0.handlers
164 188
aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols_key_id_discovery_v1_0.handlers
165 188
aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols_key_id_discovery_v1_0.message_t
166 191
aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols.discovery_v1_0.messages,
167 188
aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols.discovery_v1_0.messages
168 188
aries_cloudagent.protocols.coordinate_mediation_on_cloudagent_protocols.discovery_v1_0.messages
174 190
aries_cloudagent.protocols.didcomm_prefix_aries_cloudagent.protocols.discovery_v1_0.routes,
304 191
aries_cloudagent.protocols.didexchange, aries_cloudagent.protocols.introduction,
178 192
aries_cloudagent.protocols.didexchange.definition_aries_cloudagent.protocols.introduction.definition,
187 199
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1,
178 192
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.base_
178 196
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.demo_
178 197
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.handle
178 192
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.handle
179 192
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.handle
179 193
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.handle
183 193
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.messa
185 198
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.messa
179 193
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.messa
179 193
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.messa
180 194
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.messa
181 195
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.introduction.v0_1.messa
182 198
aries_cloudagent.protocols.didexchange.variables_aries_cloudagent.protocols.issue_credential,
185 199
aries_cloudagent.protocols.discovery, aries_cloudagent.protocols.issue_credential.definition
187 247
aries_cloudagent.protocols.discovery.definition_aries_cloudagent.protocols.issue_credential.v1_0,
192 199
aries_cloudagent.protocols.discovery.v1_0_aries_cloudagent.protocols.issue_credential.v1_0.ha
187 199

```


aries_cloudagent.protocols.issue_credentials	aries_v1cloudagentsprotocols.issue_credentials.v2_0.message
199	222
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagentsprotocols.issue_credentials.v2_0.message
199	224
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagentsprotocols.issue_credentials.v2_0.message
200	225
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagentsprotocols.issue_credentials.v2_0.message
200	226
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagentsprotocols.issue_credentials.v2_0.message
200	228
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagent.protocols.issue_credentials.v2_0.message
212	229
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessage_typesprotocols.issue_credentials.v2_0.message
215	231
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
201	222
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
203	222
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
203	232
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
204	235
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
206	232
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
207	232
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
201	233
aries_cloudagent.protocols.issue_credentials	aries_v1cloudmessagesprotocols.issue_credentials.v2_0.message
201	241
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagent.protocols.out_of_band,
209	247
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagent.protocols.out_of_band,
209	262
aries_cloudagent.protocols.issue_credentials	aries_v1cloudagent.protocols.out_of_band.v1_0,
215	247
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagent.protocols.out_of_band.v1_0.handler
220	247
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagentsprotocols.out_of_band.v1_0.handler
220	247
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagentsprotocols.out_of_band.v1_0.handler
221	248
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagentsprotocols.out_of_band.v1_0.handler
221	248
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagentsprotocols.out_of_band.v1_0.manager
221	257
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagentsprotocols.out_of_band.v1_0.manager
222	260
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagentsprotocols.out_of_band.v1_0.manager
222	248
aries_cloudagent.protocols.issue_credentials	aries_v2cloudagent.protocols.out_of_band.v1_0.manager
238	248
aries_cloudagent.protocols.issue_credentials	aries_v2cloudmessage_typesprotocols.out_of_band.v1_0.manager
241	251

```

aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent.protocols.present_proof.v1_0.util
252 274
aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent.protocols.present_proof.v1_0.util
253 274
aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent.protocols.problem_report,
254 284
aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent.protocols.problem_report.definition
255 286
aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent.protocols.problem_report.v1_0,
255 284
aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent.protocols.problem_report.v1_0.han
260 284
aries_cloudagent.protocols.present_proof.aries_cloudagent.protocols.problem_report.v1_0.mes
262 284
aries_cloudagent.protocols.present_proof.aries_cloudagent.protocols.problem_report.v1_0.mes
284 286
aries_cloudagent.protocols.present_proof.aries_cloudagent.protocols.routing, 287
262 aries_cloudagent.protocols.routing.definition,
aries_cloudagent.protocols.present_proof.v1_0.handlers,
262 aries_cloudagent.protocols.routing.v1_0,
aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack_handler,
262 aries_cloudagent.protocols.routing.v1_0.handlers,
aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_handler,
262 aries_cloudagent.protocols.routing.v1_0.handlers.f
aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_proposal_handler,
263 aries_cloudagent.protocols.routing.v1_0.handlers.ro
aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_request_handler,
263 aries_cloudagent.protocols.routing.v1_0.handlers.ro
aries_cloudagent.protocols.present_proof.v1_0.manager,
275 aries_cloudagent.protocols.routing.v1_0.handlers.ro
aries_cloudagent.protocols.present_proof.v1_0.messages,
277 aries_cloudagent.protocols.routing.v1_0.handlers.ro
aries_cloudagent.protocols.present_proof.v1_0.messages,
263 aries_cloudagent.protocols.routing.v1_0.manager,
aries_cloudagent.protocols.present_proof.v1_0.messages.inner,
263 aries_cloudagent.protocols.routing.v1_0.message_ty
aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview,
263 aries_cloudagent.protocols.routing.v1_0.messages,
aries_cloudagent.protocols.present_proof.v1_0.messages.presentation,
267 aries_cloudagent.protocols.routing.v1_0.messages.f
aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack,
268 aries_cloudagent.protocols.routing.v1_0.messages.ro
aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal,
269 aries_cloudagent.protocols.routing.v1_0.messages.ro
aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request,
270 aries_cloudagent.protocols.routing.v1_0.messages.ro
aries_cloudagent.protocols.present_proof.v1_0.models,
271 aries_cloudagent.protocols.routing.v1_0.messages.ro
aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange,
271 aries_cloudagent.protocols.routing.v1_0.models,
aries_cloudagent.protocols.present_proof.v1_0.messages,
278 aries_cloudagent.protocols.routing.v1_0.models.pag
aries_cloudagent.protocols.present_proof.v1_0.messages,
274 aries_cloudagent.protocols.routing.v1_0.models.pag

```

293 aries_cloudagent.tails.indy_tails_server,
aries_cloudagent.protocols.routing.v1_0.models.route_query_result,
294 aries_cloudagent.transport, 327
aries_cloudagent.protocols.routing.v1_0.models.route_query_result, 344
295 aries_cloudagent.transport.inbound, 327
aries_cloudagent.protocols.routing.v1_0.models.route_update, 328
297 aries_cloudagent.transport.inbound.base,
aries_cloudagent.protocols.routing.v1_0.models.route_update, 329
298 aries_cloudagent.transport.inbound.delivery_queue,
300 aries_cloudagent.transport.inbound.http,
aries_cloudagent.protocols.trustping.definition, 330
304 aries_cloudagent.transport.inbound.manager,
aries_cloudagent.protocols.trustping.v1_0.messages, 331
300 aries_cloudagent.transport.inbound.message,
aries_cloudagent.protocols.trustping.v1_0.messages, 332
300 aries_cloudagent.transport.inbound.receipt,
aries_cloudagent.protocols.trustping.v1_0.messages, 333
301 aries_cloudagent.transport.inbound.session,
aries_cloudagent.protocols.trustping.v1_0.messages, 334
301 aries_cloudagent.transport.inbound.ws,
aries_cloudagent.protocols.trustping.v1_0.messages, 336
303 aries_cloudagent.transport.outbound, 337
aries_cloudagent.transport.outbound.base,
aries_cloudagent.protocols.trustping.v1_0.messages, 337
301 aries_cloudagent.transport.outbound.http,
aries_cloudagent.protocols.trustping.v1_0.messages, 338
301 aries_cloudagent.transport.outbound.manager,
aries_cloudagent.protocols.trustping.v1_0.messages, 338
302 aries_cloudagent.transport.outbound.message,
aries_cloudagent.protocols.trustping.v1_0.messages, 341
303 aries_cloudagent.transport.outbound.ws,
aries_cloudagent.revocation, 304 342
aries_cloudagent.revocation.error, 313 aries_cloudagent.transport.pack_format,
aries_cloudagent.revocation.indy, 314 345
aries_cloudagent.revocation.manager, 314 aries_cloudagent.transport.queue, 343
aries_cloudagent.revocation.models, 304 aries_cloudagent.transport.queue.base,
aries_cloudagent.revocation.models.indy, 343
305 aries_cloudagent.transport.queue.basic,
aries_cloudagent.revocation.models.issuer_cred_rev_record, 343
305 aries_cloudagent.transport.stats, 346
aries_cloudagent.revocation.models.issuer_rev_reg_record, 346
307 aries_cloudagent.transport.wire_format,
aries_cloudagent.revocation.models.revocation_registry, 348
311 aries_cloudagent.utils, 348
aries_cloudagent.revocation.routes, 316 aries_cloudagent.utils.classloader, 348
aries_cloudagent.storage, 320 aries_cloudagent.utils.http, 349
aries_cloudagent.storage.base, 320 aries_cloudagent.utils.outofband, 351
aries_cloudagent.storage.error, 322 aries_cloudagent.utils.repeat, 351
aries_cloudagent.storage.in_memory, 322 aries_cloudagent.utils.stats, 352
aries_cloudagent.storage.indy, 324 aries_cloudagent.utils.task_queue, 353
aries_cloudagent.storage.record, 326 aries_cloudagent.utils.tracing, 355
aries_cloudagent.tails, 326 aries_cloudagent.version, 374
aries_cloudagent.tails.base, 326 aries_cloudagent.wallet, 356
aries_cloudagent.tails.error, 327 aries_cloudagent.wallet.base, 357
aries_cloudagent.wallet.crypto, 361

`aries_cloudagent.wallet.did_posture`, [364](#)
`aries_cloudagent.wallet.error`, [365](#)
`aries_cloudagent.wallet.in_memory`, [365](#)
`aries_cloudagent.wallet.indy`, [368](#)
`aries_cloudagent.wallet.models`, [356](#)
`aries_cloudagent.wallet.models.wallet_record`,
 [356](#)
`aries_cloudagent.wallet.routes`, [371](#)
`aries_cloudagent.wallet.util`, [374](#)

A

ABANDONED (aries_cloudagent.connections.models.conn_record.ConnRecord.State attribute), 32
 accept (aries_cloudagent.connections.models.conn_record.ConnRecordSchema attribute), 35
 ACCEPT_AUTO (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 31
 accept_complete() (aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManager method), 183
 ACCEPT_MANUAL (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 31
 accept_response() (aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager method), 146
 accept_response() (aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManager method), 183
 accept_response() (aries_cloudagent.transport.inbound.session.InboundSession method), 335
 accept_taa() (in module aries_cloudagent.config.ledger), 19
 accept_txn_author_agreement() (aries_cloudagent.ledger.base.BaseLedger method), 63
 accept_txn_author_agreement() (aries_cloudagent.ledger.indy.IndySdkLedger method), 67
 AcceptInvitationQueryStringSchema (class in aries_cloudagent.protocols.connections.v1_0.routes), 150
 AcceptRequestQueryStringSchema (class in aries_cloudagent.protocols.connections.v1_0.routes), 150
 AcceptResult (class in aries_cloudagent.transport.inbound.session), 334
 Ack (class in aries_cloudagent.messaging.ack.message), 73
 AckMeta (class in aries_cloudagent.messaging.ack.message), 73
 AckSchema (class in aries_cloudagent.messaging.ack.message), 74
 AckSchema.Meta (class in aries_cloudagent.messaging.ack.message), 74
 acquire() (aries_cloudagent.cache.base.BaseCache method), 8
 action (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages attribute), 160
 action (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages attribute), 161
 action (aries_cloudagent.protocols.routing.v1_0.models.route_update.RouteUpdate attribute), 297
 action (aries_cloudagent.protocols.routing.v1_0.models.route_updated.RouteUpdated attribute), 298
 ACTION_CREATE (aries_cloudagent.protocols.routing.v1_0.models.route_update.RouteUpdate attribute), 297
 ACTION_DELETE (aries_cloudagent.protocols.routing.v1_0.models.route_update.RouteUpdate attribute), 297
 actionmenu_close() (in module aries_cloudagent.protocols.actionmenu.v1_0.routes), 133
 actionmenu_fetch() (in module aries_cloudagent.protocols.actionmenu.v1_0.routes), 133
 actionmenu_perform() (in module aries_cloudagent.protocols.actionmenu.v1_0.routes), 133
 actionmenu_request() (in module aries_cloudagent.protocols.actionmenu.v1_0.routes), 133
 actionmenu_send() (in module aries_cloudagent.protocols.actionmenu.v1_0.routes), 133
 ActionMenuModulesResultSchema (class in aries_cloudagent.protocols.actionmenu.v1_0.routes), 133

132		add_service_pubkeys()	
active	(aries_cloudagent.core.profile.ProfileSession attribute), 47	(aries_cloudagent.connections.models.diddoc.DIDDoc method), 22	
add_active()	(aries_cloudagent.utils.task_queue.TaskQueue method), 353	add_service_pubkeys()	(aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc method), 26
add_arguments()	(aries_cloudagent.config.argparse.AdminGroup method), 11	add_trace_decorator()	(aries_cloudagent.messaging.agent_message.AgentMessage method), 107
add_arguments()	(aries_cloudagent.config.argparse.ArgumentGroup method), 11	add_trace_report()	(aries_cloudagent.messaging.agent_message.AgentMessage method), 107
add_arguments()	(aries_cloudagent.config.argparse.DenyGroup method), 12	add_webhook_target()	(aries_cloudagent.admin.base_server.BaseAdminServer method), 3
add_arguments()	(aries_cloudagent.config.argparse.GeneralGroup method), 12	add_webhook_target()	(aries_cloudagent.admin.server.AdminServer method), 6
add_arguments()	(aries_cloudagent.config.argparse.LedgerGroup method), 12	AdminAPIMessageTracingSchema	(class in aries_cloudagent.utils.tracing), 355
add_arguments()	(aries_cloudagent.config.argparse.LoggingGroup method), 12	AdminError	4
add_arguments()	(aries_cloudagent.config.argparse.MediationGroup method), 13	AdminGroup	(class in aries_cloudagent.config.argparse), 11
add_arguments()	(aries_cloudagent.config.argparse.MultitenantGroup method), 13	AdminMessageTracingSchema	(class in aries_cloudagent.protocols.coordinate_mediation.v1_0.routes), 355
add_arguments()	(aries_cloudagent.config.argparse.PrincipalGroup method), 13	AdminRequestContext	(class in aries_cloudagent.admin.request_context), 4
add_arguments()	(aries_cloudagent.config.argparse.StartupGroup method), 13	AdminSetupError	4
add_arguments()	(aries_cloudagent.config.argparse.TransportGroup method), 13	AdminStatusReadinessSchema	(class in aries_cloudagent.admin.server), 7
add_arguments()	(aries_cloudagent.config.argparse.WalletGroup method), 14	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_key()	(aries_cloudagent.multitenant.manager.MultitenantManager method), 121	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_key()	(aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager method), 171	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_key_for_did()	(aries_cloudagent.connections.base_manager.BaseConnectionManager method), 37	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_message()	(aries_cloudagent.transport.inbound.deliverer.Deliverer method), 329	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_model()	(aries_cloudagent.messaging.decorators.base_decorator.BaseDecorator method), 83	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_pending()	(aries_cloudagent.utils.task_queue.TaskQueue method), 354	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_record()	(aries_cloudagent.storage.base.BaseStorage method), 320	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_record()	(aries_cloudagent.storage.in_memory.InMemoryStorage method), 323	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_record()	(aries_cloudagent.storage.indy.IndySdkStorage method), 324	AdminStatusSchema	(class in aries_cloudagent.admin.server), 7
add_reply_thread_ids()	(aries_cloudagent.transport.inbound.session.InboundSession method), 335	AgentMessage.Meta	(class in aries_cloudagent.messaging.agent_message), 106
add_reply_verkeys()	(aries_cloudagent.transport.inbound.session.InboundSession method), 335	AgentMessageError	108
		AgentMessageSchema	(class in aries_cloudagent.messaging.agent_message), 108

AgentMessageSchema.Meta (class in [ule](#)), 3
 aries_cloudagent.messaging.agent_message), 108
 aka (aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_format.FormatSpec attribute), 225
 aka (aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_format.V20CredFormatFormat attribute), 226
 aka (aries_cloudagent.protocols.out_of_band.v1_0.messages_invitation_HSProtocol attribute), 248
 aka (aries_cloudagent.protocols.out_of_band.v1_0.messages_invitation_HSProtocolSpec attribute), 249
 alg (aries_cloudagent.wallet.crypto.PackRecipientsSchema attribute), 361
 alias (aries_cloudagent.connections.models.conn_record.ConnRecordSchema attribute), 35
 alias (aries_cloudagent.ledger.routes.RegisterLedgerNymQueryStringSchema attribute), 71
 alias (aries_cloudagent.protocols.connections.v1_0.routes_connections_list.QueryStringSchema attribute), 151
 alias (aries_cloudagent.protocols.connections.v1_0.routes_connection_static_request_schema.base attribute), 151
 alias (aries_cloudagent.protocols.connections.v1_0.routes_create_invitation_query_string_schema attribute), 152
 alias (aries_cloudagent.protocols.connections.v1_0.routes_receive_invitation_query_string_schema attribute), 152
 alias (aries_cloudagent.protocols.didexchange.v1_0.routes_didx_receive_request_implicit_query_string_schema attribute), 186
 alias (aries_cloudagent.protocols.out_of_band.v1_0.routes_invitation_create_request_schema attribute), 260
 alias (aries_cloudagent.protocols.out_of_band.v1_0.routes_invitation_receive_query_string_schema attribute), 260
 alive (aries_cloudagent.admin.server.AdminStatusLivelinessSchema attribute), 7
 aml (aries_cloudagent.ledger.routes.AMLRecordSchema attribute), 71
 aml_record (aries_cloudagent.ledger.routes.TAAInfoSchema attribute), 72
 amlContext (aries_cloudagent.ledger.routes.AMLRecordSchema attribute), 71
 AMLRecordSchema (class in [22](#)), 71
 append_trace_report () (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecoratorFor method), 91
 ArgsParseError, 16
 ArgumentGroup (class in [aries_cloudagent.config.argparse](#)), 11
 aries (aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_format.FormatSpec attribute), 225
 aries (aries_cloudagent.protocols.issue_credential.v2_0.messages_cred_format.V20CredFormatFormat attribute), 226
 aries_cloudagent (module), 3
 aries_cloudagent.admin (module), 3
 aries_cloudagent.admin.base_server (mod-

aries_cloudagent.admin.error (module), 4
 aries_cloudagent.admin.request_context (module), 4
 aries_cloudagent.admin.server (module), 5
 aries_cloudagent.cache.base (module), 8
 aries_cloudagent.cache.in_memory (module), 9
 aries_cloudagent.commands (module), 10
 aries_cloudagent.commands.help (module), 10
 aries_cloudagent.commands.provision (module), 10
 aries_cloudagent.commands.start (module), 10
 aries_cloudagent.config (module), 11
 aries_cloudagent.config.argparse (module), 11
 aries_cloudagent.config.base_context (module), 14
 aries_cloudagent.config.base_context (module), 14
 aries_cloudagent.config.default_context (module), 16
 aries_cloudagent.config.error (module), 16
 aries_cloudagent.config.error_context (module), 17
 aries_cloudagent.config.injector (module), 18
 aries_cloudagent.config.ledger (module), 19
 aries_cloudagent.config.logging (module), 19
 aries_cloudagent.config.provider (module), 20
 aries_cloudagent.config.settings (module), 21
 aries_cloudagent.config.util (module), 21
 aries_cloudagent.config.wallet (module), 22
 aries_cloudagent.connections (module), 22
 aries_cloudagent.connections.base_manager (module), 22
 aries_cloudagent.connections.models (module), 22
 aries_cloudagent.connections.models.conn_record (module), 30
 aries_cloudagent.connections.models.connection_target (module), 36
 aries_cloudagent.connections.models.diddoc (module), 22
 aries_cloudagent.connections.models.diddoc.diddoc (module), 25
 aries_cloudagent.connections.models.diddoc.publickey (module), 25

(module), 27
 aries_cloudagent.connections.models.diddoc (module), 28
 aries_cloudagent.connections.models.diddoc.util (module), 29
 aries_cloudagent.connections.util (module), 38
 aries_cloudagent.core (module), 39
 aries_cloudagent.core.conductor (module), 39
 aries_cloudagent.core.dispatcher (module), 40
 aries_cloudagent.core.error (module), 42
 aries_cloudagent.core.in_memory (module), 43
 aries_cloudagent.core.plugin_registry (module), 45
 aries_cloudagent.core.profile (module), 45
 aries_cloudagent.core.protocol_registry (module), 47
 aries_cloudagent.holder (module), 48
 aries_cloudagent.holder.routes (module), 48
 aries_cloudagent.indy (module), 50
 aries_cloudagent.indy.holder (module), 58
 aries_cloudagent.indy.issuer (module), 60
 aries_cloudagent.indy.sdk (module), 50
 aries_cloudagent.indy.sdk.error (module), 50
 aries_cloudagent.indy.sdk.holder (module), 51
 aries_cloudagent.indy.sdk.issuer (module), 53
 aries_cloudagent.indy.sdk.profile (module), 55
 aries_cloudagent.indy.sdk.util (module), 56
 aries_cloudagent.indy.sdk.verifier (module), 56
 aries_cloudagent.indy.sdk.wallet_plugin (module), 57
 aries_cloudagent.indy.sdk.wallet_setup (module), 57
 aries_cloudagent.indy.util (module), 62
 aries_cloudagent.indy.verifier (module), 63
 aries_cloudagent.ledger (module), 63
 aries_cloudagent.ledger.base (module), 63
 aries_cloudagent.ledger.endpoint_type (module), 66
 aries_cloudagent.ledger.error (module), 66
 aries_cloudagent.ledger.indy (module), 67
 aries_cloudagent.ledger.routes (module), 71
 aries_cloudagent.ledger.util (module), 73
 aries_cloudagent.messaging (module), 73
 aries_cloudagent.messaging.ack (module), 73
 aries_cloudagent.messaging.ack.message (module), 73
 aries_cloudagent.messaging.agent_message (module), 106
 aries_cloudagent.messaging.base_handler (module), 109
 aries_cloudagent.messaging.credential_definitions (module), 74
 aries_cloudagent.messaging.credential_definitions.v1 (module), 74
 aries_cloudagent.messaging.credential_definitions.v2 (module), 76
 aries_cloudagent.messaging.decorators (module), 76
 aries_cloudagent.messaging.decorators.attach_decorator (module), 76
 aries_cloudagent.messaging.decorators.base (module), 83
 aries_cloudagent.messaging.decorators.default (module), 84
 aries_cloudagent.messaging.decorators.localization (module), 84
 aries_cloudagent.messaging.decorators.please_ack_decorator (module), 86
 aries_cloudagent.messaging.decorators.signature_decorator (module), 86
 aries_cloudagent.messaging.decorators.thread_decorator (module), 88
 aries_cloudagent.messaging.decorators.timing_decorator (module), 89
 aries_cloudagent.messaging.decorators.trace_decorator (module), 91
 aries_cloudagent.messaging.decorators.transport_decorator (module), 95
 aries_cloudagent.messaging.error (module), 110
 aries_cloudagent.messaging.jsonld (module), 96
 aries_cloudagent.messaging.jsonld.create_verify_data (module), 96
 aries_cloudagent.messaging.jsonld.credential (module), 96
 aries_cloudagent.messaging.jsonld.routes (module), 97
 aries_cloudagent.messaging.models (module), 98
 aries_cloudagent.messaging.models.base (module), 98
 aries_cloudagent.messaging.models.base_record (module), 100

`aries_cloudagent.messaging.models.openapi` (module), 104
`aries_cloudagent.messaging.request_context` (module), 110
`aries_cloudagent.messaging.responder` (module), 112
`aries_cloudagent.messaging.schemas` (module), 104
`aries_cloudagent.messaging.schemas.routes` (module), 104
`aries_cloudagent.messaging.schemas.util` (module), 106
`aries_cloudagent.messaging.util` (module), 113
`aries_cloudagent.messaging.valid` (module), 114
`aries_cloudagent.multitenant` (module), 117
`aries_cloudagent.multitenant.admin` (module), 117
`aries_cloudagent.multitenant.admin.routes` (module), 118
`aries_cloudagent.multitenant.error` (module), 120
`aries_cloudagent.multitenant.manager` (module), 121
`aries_cloudagent.protocols` (module), 123
`aries_cloudagent.protocols.actionmenu` (module), 123
`aries_cloudagent.protocols.actionmenu.definition` (module), 134
`aries_cloudagent.protocols.actionmenu.v1_0` (module), 123
`aries_cloudagent.protocols.actionmenu.v1_0.base` (module), 131
`aries_cloudagent.protocols.actionmenu.v1_0.connection` (module), 131
`aries_cloudagent.protocols.actionmenu.v1_0.driver` (module), 131
`aries_cloudagent.protocols.actionmenu.v1_0.handler` (module), 123
`aries_cloudagent.protocols.actionmenu.v1_0.handler_base` (module), 123
`aries_cloudagent.protocols.actionmenu.v1_0.handler_base.request_handler` (module), 124
`aries_cloudagent.protocols.actionmenu.v1_0.handler_base.request_handler.connections` (module), 124
`aries_cloudagent.protocols.actionmenu.v1_0.messages` (module), 132
`aries_cloudagent.protocols.actionmenu.v1_0.messages.agent` (module), 124
`aries_cloudagent.protocols.actionmenu.v1_0.messages.agent.request` (module), 124
`aries_cloudagent.protocols.actionmenu.v1_0.messages.agent.request.bl` (module), 125
`aries_cloudagent.protocols.actionmenu.v1_0.messages` (module), 126
`aries_cloudagent.protocols.actionmenu.v1_0.models` (module), 127
`aries_cloudagent.protocols.actionmenu.v1_0.models.n` (module), 127
`aries_cloudagent.protocols.actionmenu.v1_0.models.n` (module), 128
`aries_cloudagent.protocols.actionmenu.v1_0.models.n` (module), 129
`aries_cloudagent.protocols.actionmenu.v1_0.routes` (module), 132
`aries_cloudagent.protocols.actionmenu.v1_0.util` (module), 134
`aries_cloudagent.protocols.basicmessage` (module), 134
`aries_cloudagent.protocols.basicmessage.definition` (module), 137
`aries_cloudagent.protocols.basicmessage.v1_0` (module), 134
`aries_cloudagent.protocols.basicmessage.v1_0.handler` (module), 135
`aries_cloudagent.protocols.basicmessage.v1_0.handler` (module), 135
`aries_cloudagent.protocols.basicmessage.v1_0.messages` (module), 136
`aries_cloudagent.protocols.basicmessage.v1_0.messages` (module), 135
`aries_cloudagent.protocols.basicmessage.v1_0.messages` (module), 135
`aries_cloudagent.protocols.basicmessage.v1_0.routes` (module), 136
`aries_cloudagent.protocols.connections` (module), 137
`aries_cloudagent.protocols.connections.definition` (module), 155
`aries_cloudagent.protocols.connections.v1_0` (module), 137
`aries_cloudagent.protocols.connections.v1_0.handler` (module), 137
`aries_cloudagent.protocols.connections.v1_0.handler` (module), 137
`aries_cloudagent.protocols.connections.v1_0.handler` (module), 138
`aries_cloudagent.protocols.connections.v1_0.handler` (module), 138
`aries_cloudagent.protocols.connections.v1_0.manager` (module), 146
`aries_cloudagent.protocols.connections.v1_0.messages` (module), 149
`aries_cloudagent.protocols.connections.v1_0.messages` (module), 138
`aries_cloudagent.protocols.connections.v1_0.messages` (module), 138

<code>aries_cloudagent.storage.base</code> (module), 320	<code>aries_cloudagent.transport.queue.basic</code> (module), 343
<code>aries_cloudagent.storage.error</code> (module), 322	<code>aries_cloudagent.transport.stats</code> (module), 346
<code>aries_cloudagent.storage.in_memory</code> (module), 322	<code>aries_cloudagent.transport.wire_format</code> (module), 346
<code>aries_cloudagent.storage.indy</code> (module), 324	<code>aries_cloudagent.utils</code> (module), 348
<code>aries_cloudagent.storage.record</code> (module), 326	<code>aries_cloudagent.utils.classloader</code> (module), 348
<code>aries_cloudagent.tails</code> (module), 326	<code>aries_cloudagent.utils.http</code> (module), 349
<code>aries_cloudagent.tails.base</code> (module), 326	<code>aries_cloudagent.utils.outofband</code> (module), 351
<code>aries_cloudagent.tails.error</code> (module), 327	<code>aries_cloudagent.utils.repeat</code> (module), 351
<code>aries_cloudagent.tails.indy_tails_server</code> (module), 327	<code>aries_cloudagent.utils.stats</code> (module), 352
<code>aries_cloudagent.transport</code> (module), 327	<code>aries_cloudagent.utils.task_queue</code> (module), 353
<code>aries_cloudagent.transport.error</code> (module), 344	<code>aries_cloudagent.utils.tracing</code> (module), 355
<code>aries_cloudagent.transport.inbound</code> (module), 327	<code>aries_cloudagent.version</code> (module), 374
<code>aries_cloudagent.transport.inbound.base</code> (module), 328	<code>aries_cloudagent.wallet</code> (module), 356
<code>aries_cloudagent.transport.inbound.deliver_queue</code> (module), 329	<code>aries_cloudagent.wallet.base</code> (module), 357
<code>aries_cloudagent.transport.inbound.http</code> (module), 330	<code>aries_cloudagent.wallet.crypto</code> (module), 361
<code>aries_cloudagent.transport.inbound.manager</code> (module), 331	<code>aries_cloudagent.wallet.did_posture</code> (module), 364
<code>aries_cloudagent.transport.inbound.message</code> (module), 332	<code>aries_cloudagent.wallet.error</code> (module), 365
<code>aries_cloudagent.transport.inbound.receiver</code> (module), 333	<code>aries_cloudagent.wallet.in_memory</code> (module), 365
<code>aries_cloudagent.transport.inbound.session</code> (module), 334	<code>aries_cloudagent.wallet.indy</code> (module), 368
<code>aries_cloudagent.transport.inbound.ws</code> (module), 336	<code>aries_cloudagent.wallet.models</code> (module), 356
<code>aries_cloudagent.transport.outbound</code> (module), 337	<code>aries_cloudagent.wallet.models.wallet_record</code> (module), 356
<code>aries_cloudagent.transport.outbound.base</code> (module), 337	<code>aries_cloudagent.wallet.routes</code> (module), 371
<code>aries_cloudagent.transport.outbound.http</code> (module), 338	<code>aries_cloudagent.wallet.util</code> (module), 374
<code>aries_cloudagent.transport.outbound.manager</code> (module), 338	<code>assign_thread_from()</code> (aries_cloudagent.messaging.agent_message.AgentMessage method), 107
<code>aries_cloudagent.transport.outbound.message</code> (module), 341	<code>assign_thread_id()</code> (aries_cloudagent.messaging.agent_message.AgentMessage method), 107
<code>aries_cloudagent.transport.outbound.ws</code> (module), 342	<code>assign_trace_decorator()</code> (aries_cloudagent.messaging.agent_message.AgentMessage method), 107
<code>aries_cloudagent.transport.pack_format</code> (module), 345	<code>assign_trace_from()</code> (aries_cloudagent.messaging.agent_message.AgentMessage method), 107
<code>aries_cloudagent.transport.queue</code> (module), 343	<code>attach_id</code> (aries_cloudagent.protocols.issue_credential.v2_0.messages attribute), 226
<code>aries_cloudagent.transport.queue.base</code> (module), 343	<code>attach_invitation()</code> (aries_cloudagent.connections.models.conn_record.ConnRecord method), 107

Index 395

attribute), 260	aries_cloudagent.messaging.jsonld.credential),
auto_accept (aries_cloudagent.protocols.out_of_band.v1_0.route	InvitationReceiveQueryStringSchema
attribute), 261	backend (aries_cloudagent.core.profile.Profile at-
auto_issue (aries_cloudagent.protocols.issue_credential.v1_0.mod	credential_exchange.V10CredentialExchangeSchema
attribute), 212	backend (aries_cloudagent.ledger.base.BaseLedger at-
auto_issue (aries_cloudagent.protocols.issue_credential.v1_0.route	credential_offer.V10CredentialOfferRequestSchema
attribute), 216	BACKEND_NAME (aries_cloudagent.core.in_memory.InMemoryProfile
auto_issue (aries_cloudagent.protocols.issue_credential.v2_0.mod	credential_record.V20CredExRecordSchema
attribute), 238	BACKEND_NAME (aries_cloudagent.core.profile.Profile
auto_issue (aries_cloudagent.protocols.issue_credential.v2_0.route	credential_offer.V20CredExOfferRequestSchema
attribute), 243	BACKEND_NAME (aries_cloudagent.indy.sdk.profile.IndySdkProfile
auto_offer (aries_cloudagent.protocols.issue_credential.v1_0.mod	credential_exchange.V10CredentialExchangeSchema
attribute), 212	BACKEND_NAME (aries_cloudagent.ledger.base.BaseLedger
auto_offer (aries_cloudagent.protocols.issue_credential.v2_0.mod	credential_record.V20CredExRecordSchema
attribute), 238	BACKEND_NAME (aries_cloudagent.ledger.indy.IndySdkLedger
auto_present (aries_cloudagent.protocols.present_proof.v1_0.mod	credential_exchange.V10PresentationExchangeSchema
attribute), 273	BadLedgerRequestError, 66
auto_present (aries_cloudagent.protocols.present_proof.v1_0.route	credential_proposal.V10PresentationProposalRequestSchema
attribute), 281	aries_cloudagent.messaging.valid), 114
auto_remove (aries_cloudagent.protocols.issue_credential.v1_0.mod	credential_exchange.V10CredentialExchangeSchema
attribute), 212	aries_cloudagent.messaging.valid), 114
auto_remove (aries_cloudagent.protocols.issue_credential.v1_0.route	credential_exchange.V10CredentialExchangeSchema
attribute), 215	aries_cloudagent.messaging.valid), 114
auto_remove (aries_cloudagent.protocols.issue_credential.v1_0.route	credential_offer.V10CredentialOfferRequestSchema
attribute), 216	aries_cloudagent.messaging.valid), 114
auto_remove (aries_cloudagent.protocols.issue_credential.v1_0.route	credential_proposal.V10CredentialProposalRequestSchema
attribute), 217	aries_cloudagent.messaging.valid), 114
auto_remove (aries_cloudagent.protocols.issue_credential.v2_0.mod	credential_record.V20CredExRecordSchema
attribute), 238	aries_cloudagent.messaging.valid), 114
auto_remove (aries_cloudagent.protocols.issue_credential.v2_0.mod	credential_record.V20CredExRecordSchema
attribute), 244	aries_cloudagent.admin.base_server), 3
available_commands () (in module	BaseCache (class in aries_cloudagent.cache.base), 8
aries_cloudagent.commands), 10	BaseConnectionManager (class in
	aries_cloudagent.connections.base_manager),
	37
B	
b58_to_bytes () (in module	BaseConnectionManagerError, 38
aries_cloudagent.wallet.util), 374	BaseDecoratorSet (class in
b64_decoded_value ()	aries_cloudagent.messaging.decorators.base),
(aries_cloudagent.protocols.issue_credential.v1_0.messages.inner.credential_preview.CredAttrSpec	
method), 201	BaseError, 42
b64_decoded_value ()	BaseExchangeRecord (class in
(aries_cloudagent.protocols.issue_credential.v2_0.messages.inner.credential_preview.V20CredExRecordSpec	aries_cloudagent.messaging.models.base_record),
method), 223	100
b64_decoded_value ()	BaseExchangeSchema (class in
(aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.PresAttrSpec	aries_cloudagent.messaging.models.base_record),
method), 264	100
b64_to_bytes () (in module	BaseExchangeSchema.Meta (class in
aries_cloudagent.wallet.util), 374	aries_cloudagent.messaging.models.base_record),
b64_to_str () (in module	100
aries_cloudagent.wallet.util), 374	BaseHandler (class in
b64decode () (in module	aries_cloudagent.messaging.base_handler),
aries_cloudagent.messaging.jsonld.credential),	109
96	BaseInboundTransport (class in
b64encode () (in module	aries_cloudagent.transport.inbound.base),

328			<i>aries_cloudagent.storage.base</i>), 321
BaseInjector	(class in <i>aries_cloudagent.config.base</i>), 14	in BaseTailsServer	(class in <i>aries_cloudagent.tails.base</i>), 326
BaseIntroductionService	(class in <i>aries_cloudagent.protocols.introduction.v0_1.base_service</i>), 196	BaseWallet	(class in <i>aries_cloudagent.wallet.base</i>), 357
BaseLedger	(class in <i>aries_cloudagent.ledger.base</i>), 63	BaseWireFormat	(class in <i>aries_cloudagent.transport.wire_format</i>), 346
BaseMenuService	(class in <i>aries_cloudagent.protocols.actionmenu.v1_0.base_service</i>), 131	BasicMessage	(class in <i>aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage</i>), 135
BaseMessageQueue	(class in <i>aries_cloudagent.transport.queue.base</i>), 343	BasicMessage.Meta	(class in <i>aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage</i>), 135
BaseModel	(class in <i>aries_cloudagent.messaging.models.base</i>), 98	BasicMessageHandler	(class in <i>aries_cloudagent.protocols.basicmessage.v1_0.handlers.basicmessage</i>), 135
BaseModel.Meta	(class in <i>aries_cloudagent.messaging.models.base</i>), 98	BasicMessageModuleResponseSchema	(class in <i>aries_cloudagent.protocols.basicmessage.v1_0.routes</i>), 136
BaseModelError	98	BasicMessageQueue	(class in <i>aries_cloudagent.transport.queue.basic</i>), 343
BaseModelSchema	(class in <i>aries_cloudagent.messaging.models.base</i>), 98	BasicMessageSchema	(class in <i>aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage</i>), 136
BaseModelSchema.Meta	(class in <i>aries_cloudagent.messaging.models.base</i>), 99	BasicMessageSchema.Meta	(class in <i>aries_cloudagent.protocols.basicmessage.v1_0.messages.basicmessage</i>), 136
BaseOutboundTransport	(class in <i>aries_cloudagent.transport.outbound.base</i>), 337	bind_instance()	(<i>aries_cloudagent.config.injector.Injector</i> method), 18
BaseProvider	(class in <i>aries_cloudagent.config.base</i>), 14	bind_provider()	(<i>aries_cloudagent.config.injector.Injector</i> method), 18
BaseRecord	(class in <i>aries_cloudagent.messaging.models.base_record</i>), 100	bind_providers()	(<i>aries_cloudagent.config.default_context.DefaultContext</i> method), 16
BaseRecord.Meta	(class in <i>aries_cloudagent.messaging.models.base_record</i>), 101	bind_providers()	(<i>aries_cloudagent.indy.sdk.profile.IndySdkProfile</i> method), 55
BaseRecordSchema	(class in <i>aries_cloudagent.messaging.models.base_record</i>), 103	build_context()	(<i>aries_cloudagent.config.base_context.ContextBuilder</i> method), 16
BaseRecordSchema.Meta	(class in <i>aries_cloudagent.messaging.models.base_record</i>), 103	build_context()	(<i>aries_cloudagent.config.default_context.DefaultContext</i> method), 16
BaseResponder	(class in <i>aries_cloudagent.messaging.responder</i>), 112	byte_count	(<i>aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator</i> attribute), 83
BaseSettings	(class in <i>aries_cloudagent.config.base</i>), 15	bytes_to_b58()	(in <i>aries_cloudagent.wallet.util</i>), 374
BaseStorage	(class in <i>aries_cloudagent.storage.base</i>), 320	bytes_to_b64()	(in <i>aries_cloudagent.wallet.util</i>), 374
BaseStorageSearch	(class in <i>aries_cloudagent.storage.base</i>), 321	ByteSize	(class in <i>aries_cloudagent.config.util</i>), 21
BaseStorageSearchSession	(class in	C	
		CachedProvider	(class in <i>aries_cloudagent.config.provider</i>), 20
		CacheError	8

CacheKeyLock (class in method), 70
aries_cloudagent.cache.base), 9
check_reuse_msg_state()
can_respond(aries_cloudagent.transport.inbound.session.InboundSession, aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager), 335
method), 257
cancel() (aries_cloudagent.utils.task_queue.PendingTask, method), 353
check_thread_deco()
cancel() (aries_cloudagent.utils.task_queue.TaskQueue, method), 180
method), 354
check_thread_deco()
cancel_pending() (aries_cloudagent.utils.task_queue.TaskQueue, method), 354
aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report, 252
method), 252
cancelled(aries_cloudagent.utils.task_queue.PendingTask, attribute), 353
check_thread_deco()
cancelled(aries_cloudagent.utils.task_queue.TaskQueue, method), 253
method), 253
check_thread_deco()
canon() (in module aries_cloudagent.messaging.util), (aries_cloudagent.protocols.out_of_band.v1_0.messages.reuse_authentication, 113
method), 254
canon.did() (in module aries_cloudagent.connections.models.diddoc.util), (aries_cloudagent.indy.sdk.verifier.IndySdkVerifier, 30
method), 56
canon_ref() (in module aries_cloudagent.connections.models.diddoc.util), (aries_cloudagent.indy.holder.IndyHolder, 30
attribute), 58
ciphertext(aries_cloudagent.wallet.crypto.PackMessageSchema, attribute), 211
catalogs(aries_cloudagent.messaging.decorators.localization_decorator, attribute), 85
LocalizationDecoratorSchema, 211
ClassLoader (class in aries_cloudagent.utils.classloader), 348
ClassNotFoundError, 349
ClassProvider (class in aries_cloudagent.config.provider), 20
Inject (class in aries_cloudagent.config.provider), 20
CLEAR (aries_cloudagent.cache.base.BaseCache, attribute), 8
clear() (aries_cloudagent.cache.in_memory.InMemoryCache, method), 9
CLEAR (aries_cloudagent.config.provider, attribute), 18
clear_binding() (aries_cloudagent.config.injector.Injector, method), 18
CLEAR (aries_cloudagent.config.provider, attribute), 13
cached_key()
CLEAR (aries_cloudagent.config.provider, attribute), 13
(aries_cloudagent.messaging.models.base_record.BaseRecord, class method), 101
clear_default_mediator()
CLEAR (aries_cloudagent.config.provider, attribute), 13
(aries_cloudagent.protocols.coordinate_mediation.v1_0.manager, method), 171
CLEAR (aries_cloudagent.config.provider, attribute), 13
clear_default_mediator() (in module aries_cloudagent.protocols.coordinate_mediation.v1_0.routes), 175
CLEAR (aries_cloudagent.config.provider, attribute), 14
clear_pending() (aries_cloudagent.revocation.models.issuer_rev_reg_record, method), 309
clear_pending_revocations()
check_dump_decorators() (aries_cloudagent.messaging.agent_message.AgentMessageSchema, method), 109
(aries_cloudagent.revocation.manager.RevocationManager, method), 314
check_existing_schema()
(aries_cloudagent.ledger.indy.IndySdkLedger, method), 67
clear_pending_revocations() (in module aries_cloudagent.revocation.routes), 318
check_pool_config()
clear_response() (aries_cloudagent.transport.inbound.session.InboundSession, method), 335
(aries_cloudagent.ledger.indy.IndySdkLedgerPool, method), 335

clear_value()	(aries_cloudagent.config.settings.Settings		comment	(aries_cloudagent.protocols.issue_credential.v2_0.routes.V20Cr	
method), 21			attribute), 243		
ClearPendingRevocationsRequestSchema			comment	(aries_cloudagent.protocols.issue_credential.v2_0.routes.V20Iss	
(class in aries_cloudagent.revocation.routes),			attribute), 244		
316					
close()	(aries_cloudagent.core.profile.Profile	comment	(aries_cloudagent.protocols.present_proof.v1_0.messages.presen		
method), 45			attribute), 268		
close()	(aries_cloudagent.indy.sdk.profile.IndySdkProfile	comment	(aries_cloudagent.protocols.present_proof.v1_0.messages.presen		
method), 55			attribute), 270		
close()	(aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet	comment	(aries_cloudagent.protocols.present_proof.v1_0.messages.presen		
method), 57			attribute), 271		
close()	(aries_cloudagent.ledger.indy.IndySdkLedgerPool	comment	(aries_cloudagent.protocols.present_proof.v1_0.routes.V10Presen		
method), 70			attribute), 280		
close()	(aries_cloudagent.storage.base.BaseStorageSearchSession	comment	(aries_cloudagent.protocols.present_proof.v1_0.routes.V10Presen		
method), 321			attribute), 281		
close()	(aries_cloudagent.storage.in_memory.InMemoryStorageSearch	comment	(aries_cloudagent.protocols.trustping.v1_0.messages.ping.PingS		
method), 324			attribute), 302		
close()	(aries_cloudagent.storage.indy.IndySdkStorageSearch	comment	(aries_cloudagent.protocols.trustping.v1_0.messages.ping_respo		
method), 326			attribute), 303		
close()	(aries_cloudagent.transport.inbound.session.InboundSession	comment	(aries_cloudagent.protocols.trustping.v1_0.routes.PingRequestS		
method), 335			attribute), 303		
closed	(aries_cloudagent.transport.inbound.session.InboundSession	comment	(aries_cloudagent.core.profile.ProfileSession		
attribute), 335			method), 47		
closed_session()	(aries_cloudagent.transport.inbound.manager.InboundTransportManager		module		
method), 331			aries_cloudagent.config.util), 21		
ClosedPoolError,	66		complete()	(aries_cloudagent.core.dispatcher.Dispatcher	
collector	(aries_cloudagent.transport.outbound.base.BaseOutboundThreadPool		attribute), 337		
attribute), 337			complete()	(aries_cloudagent.utils.task_queue.TaskQueue	
Collector	(class in aries_cloudagent.utils.stats), 352		method), 354		
comment	(aries_cloudagent.protocols.discovery.v1_0.messages.COMPLETED_ONION_SCHEMA				
attribute), 191			(aries_cloudagent.protocols.didexchange.v1_0.messages.problem		
comment	(aries_cloudagent.protocols.issue_credential.v1_0.messages.cred_issue.CredentialIssueSchema		attribute), 180		
attribute), 204			COMPLETED	(aries_cloudagent.connections.models.conn_record.ConnRec	
comment	(aries_cloudagent.protocols.issue_credential.v1_0.messages.cred_offer.CredentialOfferSchema		attribute), 202		
attribute), 205			completed_task()	(aries_cloudagent.utils.task_queue.TaskQueue	
comment	(aries_cloudagent.protocols.issue_credential.v1_0.messages.cred_proposal.CredentialProposalSchema		method), 254		
attribute), 207			CompletedTask	(class in	
comment	(aries_cloudagent.protocols.issue_credential.v1_0.messages.cred_request.CredentialRequestSchema		attribute), 253		
attribute), 208			Conductor	(class in aries_cloudagent.core.conductor),	
comment	(aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialCreateSchema				
attribute), 215			ConfigError,	15	
comment	(aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialIssueRequestSchema				
attribute), 216			Logging.LoggingConfigurator		
comment	(aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialOfferRequestSchema		class method), 19		
attribute), 216			connectionmenu.v1_0.routes.ConnIdMat		
comment	(aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialProposeRequestSchema		attribute), 132		
attribute), 217			connections.v1_0.routes.ConnIdM		
comment	(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue.CredentialIssueSchema		attribute), 150		
attribute), 228			connections.v1_0.routes.ConnIdRef		
comment	(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.CredentialOfferSchema		attribute), 150		
attribute), 229			mediation.v1_0.routes.C		
comment	(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal.CredentialProposalSchema		attribute), 174		
attribute), 231			mediation.v1_0.routes.M		
comment	(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_request.CredentialRequestSchema				

[illegible]

ConnectionMetadataSetRequestSchema (class in connections_list() (in module
 aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.protocols.connections.v1_0.routes),
 150 154

ConnectionModuleResponseSchema (class in connections_metadata() (in module
 aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.protocols.connections.v1_0.routes),
 151 154

ConnectionRequest (class in connections_metadata_set() (in module
 aries_cloudagent.protocols.connections.v1_0.messages.connection_request), aries_cloudagent.protocols.connections.v1_0.routes),
 140 154

ConnectionRequest.Meta (class in connections_receive_invitation() (in mod-
 aries_cloudagent.protocols.connections.v1_0.messages.connection_request), aries_cloudagent.protocols.connections.v1_0.routes),
 140 154

ConnectionRequestHandler (class in connections_remove() (in module
 aries_cloudagent.protocols.connections.v1_0.handlers.connection_request_handler), aries_cloudagent.protocols.connections.v1_0.routes),
 138 154

ConnectionRequestSchema (class in connections_retrieve() (in module
 aries_cloudagent.protocols.connections.v1_0.messages.connection_request), aries_cloudagent.protocols.connections.v1_0.routes),
 141 155

ConnectionRequestSchema.Meta (class in connections_send_message() (in module
 aries_cloudagent.protocols.connections.v1_0.messages.connection_request), aries_cloudagent.protocols.basicmessage.v1_0.routes),
 141 136

ConnectionResponse (class in connections_send_ping() (in module
 aries_cloudagent.protocols.connections.v1_0.messages.connection_response), aries_cloudagent.protocols.trustping.v1_0.routes),
 141 303

ConnectionResponse.Meta (class in ConnectionsListQueryStringSchema (class in
 aries_cloudagent.protocols.connections.v1_0.messages.connection_response), aries_cloudagent.protocols.connections.v1_0.routes),
 141 151

ConnectionResponseHandler (class in ConnectionStaticRequestSchema (class in
 aries_cloudagent.protocols.connections.v1_0.handlers.connection_response_handler), aries_cloudagent.protocols.connections.v1_0.routes),
 138 151

ConnectionResponseSchema (class in ConnectionStaticResultSchema (class in
 aries_cloudagent.protocols.connections.v1_0.messages.connection_response), aries_cloudagent.protocols.connections.v1_0.routes),
 141 151

ConnectionResponseSchema.Meta (class in ConnectionTarget (class in
 aries_cloudagent.protocols.connections.v1_0.messages.connection_response), aries_cloudagent.connections.models.connection_target),
 141 36

connections_accept_invitation() (in mod- ConnectionTarget.Meta (class in
 ule aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.connections.models.connection_target),
 153 36

connections_accept_request() (in module ConnectionTargetSchema (class in
 aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.connections.models.connection_target),
 153 36

connections_create_invitation() (in mod- ConnectionTargetSchema.Meta (class in
 ule aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.connections.models.connection_target),
 153 36

connections_create_static() (in module ConnIdMatchInfoSchema (class in
 aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.protocols.actionmenu.v1_0.routes),
 153 132

connections_endpoints() (in module ConnIdMatchInfoSchema (class in
 aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.protocols.basicmessage.v1_0.routes),
 153 136

connections_establish_inbound() (in mod- ConnIdMatchInfoSchema (class in
 ule aries_cloudagent.protocols.connections.v1_0.routes), aries_cloudagent.protocols.connections.v1_0.routes),
 154 150

ConnIdMatchInfoSchema	(class	in	aries_cloudagent.protocols.actionmenu.v1_0.controller),	131
			aries_cloudagent.protocols.introduction.v0_1.routes),	198
ConnIdMatchInfoSchema	(class	in	controllers (aries_cloudagent.core.protocol_registry.ProtocolRegistry	attribute), 47
			aries_cloudagent.protocols.trustping.v1_0.routes)copy ()	(aries_cloudagent.config.base.BaseInjector
			method), 14	303
ConnIdRefIdMatchInfoSchema	(class	in	copy ()	(aries_cloudagent.config.base.BaseSettings
			aries_cloudagent.protocols.connections.v1_0.routes),	method), 15
			copy ()	(aries_cloudagent.config.injection_context.InjectionContext
			method), 17	150
ConnRecord	(class	in	copy ()	(aries_cloudagent.config.injector.Injector
			aries_cloudagent.connections.models.conn_record)copy ()	method), 18
			method), 18	30
ConnRecord.Meta	(class	in	copy ()	(aries_cloudagent.config.settings.Settings
			aries_cloudagent.connections.models.conn_record),	method), 21
			copy ()	(aries_cloudagent.messaging.decorators.base.BaseDecoratorSet
			method), 83	32
ConnRecord.Role	(class	in	coro_ident ()	(in
			aries_cloudagent.utils.task_queue),	module
			aries_cloudagent.utils.task_queue),	355
ConnRecord.State	(class	in	coro_timed ()	(in
			aries_cloudagent.connections.models.conn_record),	module
			aries_cloudagent.utils.task_queue),	355
			count	(aries_cloudagent.holder.routes.CredentialsListQueryStringSchema
ConnRecordSchema	(class	in	attribute), 49	
			aries_cloudagent.connections.models.conn_record)count	(aries_cloudagent.protocols.present_proof.v1_0.routes.CredentialsList
			attribute), 278	35
ConnRecordSchema.Meta	(class	in	covers ()	(aries_cloudagent.revocation.models.indy.NonRevocationInter
			aries_cloudagent.connections.models.conn_record),	method), 305
			create ()	(aries_cloudagent.messaging.decorators.signature_decorator.S
content	(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator),	77		
			create_and_send_credential_definition ()	
content	(aries_cloudagent.protocols.basicmessage.v1_0.messages.kasienalaskuBasicMessageSchemaBaseLedger	attribute), 136		
			method), 63	
content	(aries_cloudagent.protocols.basicmessage.v1_0.routes.SendMessageSchemaCredential_definition ()	attribute), 136		
			(aries_cloudagent.ledger.indy.IndySdkLedger	
CONTEXT	(aries_cloudagent.connections.models.diddoc.DIDDoc	method), 67		
			create_and_send_schema ()	
CONTEXT	(aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc	aries_cloudagent.ledger.base.BaseLedger		
		method), 64		
context	(aries_cloudagent.core.conductor.Conductor	create_and_send_schema ()		
		(aries_cloudagent.ledger.indy.IndySdkLedger		
context	(aries_cloudagent.core.profile.Profile	attribute), 68		
		method), 68		
context	(aries_cloudagent.core.profile.ProfileSession	create_and_store_credential_definition ()		
		(aries_cloudagent.indy.issuer.IndyIssuer		
		method), 60		
context_close ()	(aries_cloudagent.ledger.indy.IndySdkLedgerPool	create_and_store_credential_definition ()		
		(aries_cloudagent.indy.sdk.issuer.IndySdkIssuer		
context_open ()	(aries_cloudagent.ledger.indy.IndySdkLedgerPool	method), 53		
		method), 70		
		create_and_store_revocation_registry ()		
ContextBuilder	(class	in	(aries_cloudagent.indy.issuer.IndyIssuer	
		aries_cloudagent.config.base_context), 16		
		method), 60		
controller	(aries_cloudagent.connections.models.diddoc.PublicKey	create_and_store_revocation_registry ()		
		(aries_cloudagent.indy.sdk.issuer.IndySdkIssuer		
controller	(aries_cloudagent.connections.models.diddoc.publickey.PublicKey	attribute), 53		
		attribute), 28		
		create_argument_parser ()	(in	module
Controller	(class	in	aries_cloudagent.config.argparse), 14	

`create_auth_token()` (*aries_cloudagent.wallet.in_memory.InMemoryWallet*
(*aries_cloudagent.multitenant.manager.MultitenantManager*
method), 121
`create_bound_request()` (*aries_cloudagent.wallet.indy.IndySdkWallet*
(*aries_cloudagent.protocols.present_proof.v1_0.manager.PresentProofManager*
method), 275
`create_credential()` (*aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager*
method), 212
(*aries_cloudagent.indy.issuer.IndyIssuer*
method), 61
`create_credential()` (*aries_cloudagent.protocols.issue_credential.v2_0.manager.V200CredentialManager*
method), 238
(*aries_cloudagent.indy.sdk.issuer.IndySdkIssuer*
method), 53
`create_credential_offer()` (*aries_cloudagent.core.dispatcher.DispatcherResponder*
method), 42
(*aries_cloudagent.indy.issuer.IndyIssuer*
method), 61
`create_credential_offer()` (*aries_cloudagent.messaging.responder.BaseResponder*
method), 112
(*aries_cloudagent.indy.sdk.issuer.IndySdkIssuer*
method), 54
`create_credential_request()` (*aries_cloudagent.ledger.indy.IndySdkLedgerPool*
method), 70
(*aries_cloudagent.indy.holder.IndyHolder*
method), 59
`create_credential_request()` (*aries_cloudagent.indy.holder.IndyHolder*
method), 59
(*aries_cloudagent.indy.sdk.holder.IndySdkHolder*
method), 51
`create_credential_request()` (*aries_cloudagent.indy.sdk.holder.IndySdkHolder*
method), 51
`create_did_document()` (*aries_cloudagent.protocols.present_proof.v1_0.manager.PresentProofManager*
method), 276
(*aries_cloudagent.connections.base_manager.BaseConnectionManager*
method), 37
`create_exchange_for_proposal()` (*aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager*
method), 213
(*aries_cloudagent.protocols.present_proof.v1_0.manager.PresentProofManager*
method), 276
`create_exchange_for_request()` (*aries_cloudagent.protocols.issue_credential.v2_0.manager.V200CredentialManager*
method), 239
(*aries_cloudagent.protocols.present_proof.v1_0.manager.PresentProofManager*
method), 276
`create_handshake_reuse_message()` (*aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager*
method), 213
(*aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager*
method), 257
`create_invitation()` (*aries_cloudagent.wallet.base.BaseWallet*
method), 358
(*aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager*
method), 146
`create_invitation()` (*aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManager*
method), 183
(*aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager*
method), 258
`create_jws()` (*aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager*
method), 213
(in module *aries_cloudagent.messaging.jsonld.credential*),
96
`create_keylist_query_response()` (*aries_cloudagent.protocols.issue_credential.v2_0.manager.V200CredentialManager*
method), 239
(*aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager*
method), 171
`create_keypair()` (*aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager*
method), 147
(in module *aries_cloudagent.wallet.crypto*), 362
`create_local_did()` (*aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManager*
method), 184
(*aries_cloudagent.wallet.base.BaseWallet*
method), 358
`create_local_did()` (*aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXManager*
method), 184
(in module

`aries_cloudagent.revocation.routes)`, 318
`create_revocation_state()`
 (`aries_cloudagent.indy.holder.IndyHolder`
 method), 59
`create_revocation_state()`
 (`aries_cloudagent.indy.sdk.holder.IndySdkHolder`
 method), 51
`create_route_record()`
 (`aries_cloudagent.protocols.routing.v1_0.manager.RoutingManager`
 method), 299
`create_schema()` (`aries_cloudagent.indy.issuer.IndyIssuer`
 method), 61
`create_schema()` (`aries_cloudagent.indy.sdk.issuer.IndySdkIssuer`
 method), 54
`create_session()` (`aries_cloudagent.transport.inbound.base.BaseInboundTransport`
 method), 328
`create_session()` (`aries_cloudagent.transport.inbound.manager.InboundTransportManager`
 method), 331
`create_signing_key()`
 (`aries_cloudagent.wallet.base.BaseWallet`
 method), 358
`create_signing_key()`
 (`aries_cloudagent.wallet.in_memory.InMemoryWallet`
 method), 366
`create_signing_key()`
 (`aries_cloudagent.wallet.indy.IndySdkWallet`
 method), 368
`create_static_connection()`
 (`aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager`
 method), 147
`create_tails_reader()` (in module
 `aries_cloudagent.indy.sdk.util`), 56
`create_tails_writer()` (in module
 `aries_cloudagent.indy.sdk.util`), 56
`create_verify_data()` (in module
 `aries_cloudagent.messaging.jsonld.create_verify_data`),
 96
`create_wallet()` (`aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig`
 method), 58
`create_wallet()` (`aries_cloudagent.multitenant.manager.MultitenantManager`
 method), 121
`created` (`aries_cloudagent.core.profile.Profile` at-
 tribute), 46
`created_at` (`aries_cloudagent.messaging.models.base_record.BaseRecordSchema`
 attribute), 103
`CreateInvitationQueryStringSchema` (class
 in `aries_cloudagent.protocols.connections.v1_0.routes`),
 152
`CreateInvitationRequestSchema` (class in
 `aries_cloudagent.protocols.connections.v1_0.routes`),
 152
`CreateWalletRequestSchema` (class in
 `aries_cloudagent.multitenant.admin.routes`),
 118
`CreateWalletResponseSchema` (class in
 `aries_cloudagent.multitenant.admin.routes`),
 118
`CreateWalletTokenRequestSchema` (class in
 `aries_cloudagent.multitenant.admin.routes`),
 118
`CreateWalletTokenResponseSchema` (class in
 `aries_cloudagent.multitenant.admin.routes`),
 118
`cred()` (`aries_cloudagent.protocols.issue_credential.v2_0.messages.cred`
 method), 227
`cred_def_id` (`aries_cloudagent.holder.routes.CredBriefSchema`
 attribute), 48
`cred_def_id` (`aries_cloudagent.messaging.credential_definitions.routes`
 attribute), 176
`cred_def_id` (`aries_cloudagent.messaging.credential_definitions.util.CredDefUtil`
 attribute), 176
`cred_def_id` (`aries_cloudagent.protocols.issue_credential.v1_0.messages`
 attribute), 207
`cred_def_id` (`aries_cloudagent.protocols.issue_credential.v1_0.routes`
 attribute), 215
`cred_def_id` (`aries_cloudagent.protocols.issue_credential.v1_0.routes`
 attribute), 216
`cred_def_id` (`aries_cloudagent.protocols.issue_credential.v1_0.routes`
 attribute), 217
`cred_def_id` (`aries_cloudagent.protocols.issue_credential.v2_0.routes`
 attribute), 242
`cred_def_id` (`aries_cloudagent.protocols.present_proof.v1_0.messages`
 attribute), 266
`cred_def_id` (`aries_cloudagent.protocols.present_proof.v1_0.messages`
 attribute), 278
`cred_def_id` (`aries_cloudagent.protocols.present_proof.v1_0.routes`
 attribute), 279
`cred_def_id` (`aries_cloudagent.revocation.models.issuer_cred_rev_recor`
 attribute), 307
`cred_def_id` (`aries_cloudagent.revocation.models.issuer_rev_reg_recor`
 attribute), 310
`cred_def_id` (`aries_cloudagent.revocation.models.revocation_registry.R`
 attribute), 312
`cred_def_id` (`aries_cloudagent.revocation.routes.CredDefIdMatchInfoS`
 attribute), 316
`cred_def_id` (`aries_cloudagent.revocation.routes.CredDefIdMatchInfoS`
 attribute), 317
`cred_def_id` (`aries_cloudagent.revocation.routes.RevRegsCreatedQuer`
 attribute), 317
`cred_def_id` (`aries_cloudagent.wallet.routes.CredDefIdMatchInfoSche`
 attribute), 371
`cred_ex_def_id` (`aries_cloudagent.protocols.issue_credential.v2_0.mo`
 attribute), 233
`cred_ex_def_id` (`aries_cloudagent.protocols.issue_credential.v2_0.mo`
 attribute), 233
`cred_ex_id` (`aries_cloudagent.protocols.issue_credential.v1_0.routes.C`
 attribute), 215
`cred_ex_id` (`aries_cloudagent.protocols.issue_credential.v2_0.models.c`

Index	405
--------------	------------

```

credential_definition_in_wallet() credential_exchange_retrieve() (in module
    (aries_cloudagent.indy.sdk.issuer.IndySdkIssuer aries_cloudagent.protocols.issue_credential.v1_0.routes),
    method), 54 218
credential_definitions_created() (in module credential_exchange_retrieve() (in module
    aries_cloudagent.messaging.credential_definitions.routes), aries_cloudagent.protocols.issue_credential.v2_0.routes),
    75 245
credential_definitions_get_credential_definition() credential_exchange_send() (in module
    (in module aries_cloudagent.messaging.credential_definitions.routes), aries_cloudagent.protocols.issue_credential.v1_0.routes),
    75 219
credential_definitions_send_credential_definition() credential_exchange_send() (in module
    (in module aries_cloudagent.messaging.credential_definitions.routes), aries_cloudagent.protocols.issue_credential.v2_0.routes),
    75 245
credential_exchange_create() (in module credential_exchange_send_bound_offer()
    aries_cloudagent.protocols.issue_credential.v1_0.routes), (in module aries_cloudagent.protocols.issue_credential.v1_0.routes),
    217 219
credential_exchange_create() (in module credential_exchange_send_bound_offer()
    aries_cloudagent.protocols.issue_credential.v2_0.routes), (in module aries_cloudagent.protocols.issue_credential.v2_0.routes),
    244 246
credential_exchange_create_free_offer() credential_exchange_send_free_offer()
    (in module aries_cloudagent.protocols.issue_credential.v1_0.routes), (in module aries_cloudagent.protocols.issue_credential.v1_0.routes),
    217 219
credential_exchange_create_free_offer() credential_exchange_send_free_offer()
    (in module aries_cloudagent.protocols.issue_credential.v2_0.routes), (in module aries_cloudagent.protocols.issue_credential.v2_0.routes),
    244 246
credential_exchange_id credential_exchange_send_proposal() (in
    (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange_v1_0.CredentialExchange
    attribute), 211 219
credential_exchange_id credential_exchange_send_proposal() (in
    (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange_v2_0.CredentialExchange
    attribute), 212 246
credential_exchange_issue() (in module credential_exchange_send_request() (in
    aries_cloudagent.protocols.issue_credential.v1_0.routes), module aries_cloudagent.protocols.issue_credential.v1_0.routes),
    218 220
credential_exchange_issue() (in module credential_exchange_send_request() (in
    aries_cloudagent.protocols.issue_credential.v2_0.routes), module aries_cloudagent.protocols.issue_credential.v2_0.routes),
    244 246
credential_exchange_list() (in module credential_exchange_store() (in module
    aries_cloudagent.protocols.issue_credential.v1_0.routes), aries_cloudagent.protocols.issue_credential.v1_0.routes),
    218 220
credential_exchange_list() (in module credential_exchange_store() (in module
    aries_cloudagent.protocols.issue_credential.v2_0.routes), aries_cloudagent.protocols.issue_credential.v2_0.routes),
    245 246
credential_exchange_problem_report() (in credential_id(aries_cloudagent.holder.routes.CredIdMatchInfoScheme
    module aries_cloudagent.protocols.issue_credential.v1_0.routes), attribute), 49
    218 credential_id(aries_cloudagent.protocols.issue_credential.v1_0.models
credential_exchange_problem_report() (in attribute), 212
    module aries_cloudagent.protocols.issue_credential.v2_0.routes), id(aries_cloudagent.protocols.issue_credential.v1_0.routes
    245 attribute), 215
credential_exchange_remove() (in module credential_id(aries_cloudagent.protocols.issue_credential.v1_0.routes
    aries_cloudagent.protocols.issue_credential.v1_0.routes), attribute), 217
    218 credential_id(aries_cloudagent.protocols.issue_credential.v2_0.routes
credential_exchange_remove() (in module attribute), 243
    aries_cloudagent.protocols.issue_credential.v2_0.routes), credential_id(aries_cloudagent.protocols.issue_credential.v2_0.routes
    245 attribute), 244

```

Index	407
--------------	------------

CredentialOffer.Meta (class in credentials_attach
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer), 204
 205
 CredentialOfferHandler (class in credentials_attach
 aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_offer_handler), 204
 200
 CredentialOfferSchema (class in credentials_attr_mime_types_get() (in
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer), 49
 205
 CredentialOfferSchema.Meta (class in aries_cloudagent.holder.routes), 49
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_offer), (in module
 205
 CredentialPreview (class in credentials_remove() (in module
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_preview), 50
 202
 CredentialPreview.Meta (class in aries_cloudagent.holder.routes), 50
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_preview), (in module
 202
 CredentialPreviewSchema (class in aries_cloudagent.holder.routes), 50
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_preview), (in module
 202
 CredentialPreviewSchema.Meta (class in CredExIdMatchInfoSchema (class in
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_preview), 215
 202
 CredentialProposal (class in CredExIdMatchInfoSchema (class in
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal), 316
 206
 CredentialProposal.Meta (class in aries_cloudagent.holder.routes), 49
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal), (class in
 206
 CredentialProposalHandler (class in aries_cloudagent.protocols.issue_credential.v1_0.routes),
 200
 aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_proposal_handler), (class in
 200
 CredentialProposalSchema (class in CredRevokedResultSchema (class in
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal), 49
 207
 CredentialProposalSchema.Meta (class in aries_cloudagent.revocation.routes), 316
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal), (class in
 207
 CredentialRequest (class in current_active(aries_cloudagent.utils.task_queue.TaskQueue
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request),
 207
 current_pending(aries_cloudagent.utils.task_queue.TaskQueue
 CredentialRequest.Meta (class in attribute), 354
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request), (in module
 208
 CredentialRequestHandler (class in aries_cloudagent.holder.routes), 49
 aries_cloudagent.protocols.issue_credential.v1_0.handlers.credential_request_handler),
 200
 CredentialRequestSchema (class in attribute), 83
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request), (class in
 208
 CredentialRequestSchema.Meta (class in data_json() (aries_cloudagent.messaging.decorators.attach_decorator
 aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request),
 208
 class method), 77
 class method), 77

[data_links\(\)](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator](#) class method), 78

[datetime_now\(\)](#) (in module [aries_cloudagent.messaging.util](#)), 113

[datetime_to_str\(\)](#) (in module [aries_cloudagent.messaging.util](#)), 113

[debug_middleware\(\)](#) (in module [aries_cloudagent.admin.server](#)), 8

[DebugGroup](#) (class in [aries_cloudagent.config.argparse](#)), 11

[decode\(\)](#) ([aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator](#) method), 87

[decode_inbound_message\(\)](#) (in module [aries_cloudagent.utils.tracing](#)), 355

[decode_pack_message\(\)](#) (in module [aries_cloudagent.wallet.crypto](#)), 362

[decode_pack_message_outer\(\)](#) (in module [aries_cloudagent.wallet.crypto](#)), 362

[decode_pack_message_payload\(\)](#) (in module [aries_cloudagent.wallet.crypto](#)), 362

[DecoratorError](#), 84

[DecoratorSet](#) (class in [aries_cloudagent.messaging.decorators.default](#)), 84

[decrypt_plaintext\(\)](#) (in module [aries_cloudagent.wallet.crypto](#)), 362

[default](#) ([aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form_param.MenuFormParamSchema](#) attribute), 129

[DEFAULT_CACHE_TTL](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) attribute), 101

[default_endpoint](#) ([aries_cloudagent.messaging.request_context.RequestContext](#) attribute), 111

[DEFAULT_FRESHNESS](#) ([aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig](#) attribute), 57

[DEFAULT_KEY](#) ([aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig](#) attribute), 58

[DEFAULT_KEY_DERIVATION](#) ([aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig](#) attribute), 58

[default_label](#) ([aries_cloudagent.messaging.request_context.RequestContext](#) attribute), 111

[DEFAULT_MEDIATOR_RECORD_TYPE](#) ([aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager](#) attribute), 170

[DEFAULT_NAME](#) ([aries_cloudagent.core.profile.Profile](#) attribute), 45

[DEFAULT_STORAGE_TYPE](#) ([aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig](#) attribute), 58

[DefaultContextBuilder](#) (class in [aries_cloudagent.config.default_context](#)), 16

[delay_milli](#) ([aries_cloudagent.messaging.decorators.timing_decorator.timing_decorator](#) attribute), 91

[delete_all_records\(\)](#) ([aries_cloudagent.storage.base.BaseStorage](#) method), 320

[delete_all_records\(\)](#) ([aries_cloudagent.storage.in_memory.InMemoryStorage](#) method), 323

[decode\(\)](#) ([aries_cloudagent.storage.indy.IndySdkStorage](#) method), 324

[delete_cred_ex_record\(\)](#) ([aries_cloudagent.protocols.issue_credential.v2_0.manager.V200Manager](#) method), 239

[delete_credential\(\)](#) ([aries_cloudagent.indy.holder.IndyHolder](#) method), 59

[delete_credential\(\)](#) ([aries_cloudagent.indy.sdk.holder.IndySdkHolder](#) method), 52

[delete_mediation_request\(\)](#) (in module [aries_cloudagent.protocols.coordinate_mediation.v1_0.routes](#)), 176

[delete_record\(\)](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) method), 323

[delete_record\(\)](#) ([aries_cloudagent.storage.base.BaseStorage](#) method), 320

[delete_record\(\)](#) ([aries_cloudagent.storage.in_memory.InMemoryStorage](#) method), 323

[delete_route_record\(\)](#) ([aries_cloudagent.protocols.routing.v1_0.manager.RoutingManager](#) method), 299

[dequeued_message\(\)](#) ([aries_cloudagent.transport.outbound.manager.OutboundTransportManager](#) method), 339

[DeliveryQueue](#) (class in [aries_cloudagent.transport.inbound.delivery_queue](#)), 329

[DemoIntroductionService](#) (class in [aries_cloudagent.protocols.introduction.v0_1.demo_service](#)), 187

[deny_request\(\)](#) ([aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager](#) method), 171

[dequeue\(\)](#) ([aries_cloudagent.transport.queue.base.BaseMessageQueue](#) method), 343

[dequeue\(\)](#) ([aries_cloudagent.transport.queue.basic.BasicMessageQueue](#) method), 343

[description](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator](#) attribute), 83

[description](#) ([aries_cloudagent.protocols.actionmenu.v1_0.messages.messages](#) attribute), 129

[attribute](#)), 125
[description](#) ([aries_cloudagent.protocols.actionmenu.v1_0.models.actionmenu.v1_0.model.MenuFormSchema](#)
[attribute](#)), 128
[description](#) ([aries_cloudagent.protocols.actionmenu.v1_0.models.actionmenu.v1_0.model.MenuFormParamSchema](#)
[attribute](#)), 129
[description](#) ([aries_cloudagent.protocols.actionmenu.v1_0.models.actionmenu.v1_0.model.MenuOptionSchema](#)
[attribute](#)), 130
[description](#) ([aries_cloudagent.protocols.actionmenu.v1_0.routes.actionmenu.v1_0.route.MenuFormSchema](#)
[attribute](#)), 132
[deserialize\(\)](#) ([aries_cloudagent.connections.models.diddoc.DIDDoc](#) [attribute](#)), 372
[class method](#)), 22
[deserialize\(\)](#) ([aries_cloudagent.connections.models.diddoc.DIDDoc](#) [attribute](#)), 372
[class method](#)), 26
[deserialize\(\)](#) ([aries_cloudagent.messaging.models.base.BaseModel](#) [attribute](#)), 143
[class method](#)), 98
[detail](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential.v2_0.message.FormatSpec](#)
[attribute](#)), 225
[detail](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential.v2_0.message.V20CredFormat.Format](#)
[attribute](#)), 226
[determine_roles\(\)](#)
[\(aries_cloudagent.protocols.actionmenu.v1_0.controllers.actionmenu.v1_0.controller\)](#) (in module
[method](#)), 131
[did](#) ([aries_cloudagent.connections.models.connection_target.ConnectionTargetSchema](#)
[attribute](#)), 37
[did](#) ([aries_cloudagent.connections.models.diddoc.DIDDoc](#) [attribute](#)), 23
[did](#) ([aries_cloudagent.connections.models.diddoc.DIDDoc](#) [attribute](#)), 26
[did](#) ([aries_cloudagent.connections.models.diddoc.PublicKey](#) [attribute](#)), 24
[did](#) ([aries_cloudagent.connections.models.diddoc.publickey.PublicKey](#) [attribute](#)), 28
[did](#) ([aries_cloudagent.connections.models.diddoc.Service](#) [attribute](#)), 25
[did](#) ([aries_cloudagent.connections.models.diddoc.service.Service](#) [attribute](#)), 29
[did](#) ([aries_cloudagent.ledger.routes.QueryStringDIDSchema](#) [attribute](#)), 71
[did](#) ([aries_cloudagent.ledger.routes.QueryStringEndpointSchema](#) [attribute](#)), 71
[did](#) ([aries_cloudagent.ledger.routes.RegisterLedgerNymQueryStringSchema](#) [attribute](#)), 71
[did](#) ([aries_cloudagent.protocols.connections.v1_0.messages.connection.v1_0.message.invitation.ConnectionInvitationSchema](#)
[attribute](#)), 140
[did](#) ([aries_cloudagent.protocols.connections.v1_0.models.connection.v1_0.model.ConnectionDetailSchema](#)
[attribute](#)), 143
[did](#) ([aries_cloudagent.protocols.connections.v1_0.models.connection.v1_0.model.ConnectionDetailSchema](#)
[attribute](#)), 144
[did](#) ([aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDRequestSchema](#) [attribute](#)), 182
[did](#) ([aries_cloudagent.protocols.didexchange.v1_0.messages.response.DIDResponseSchema](#) [attribute](#)), 182
[did](#) ([aries_cloudagent.protocols.out_of_band.v1_0.messages.out_of_band.v1_0.message.ServiceSchema](#) [attribute](#)), 255
[did](#) ([aries_cloudagent.wallet.base.DIDInfo](#) [attribute](#)), 360
[did](#) ([aries_cloudagent.wallet.routes.DIDEndpointSchema](#) [attribute](#)), 372
[did](#) ([aries_cloudagent.wallet.routes.DIDEndpointWithTypeSchema](#) [attribute](#)), 372
[did](#) ([aries_cloudagent.wallet.routes.DIDListQueryStringSchema](#) [attribute](#)), 372
[did](#) ([aries_cloudagent.wallet.routes.DIDQueryStringSchema](#) [attribute](#)), 372
[did](#) ([aries_cloudagent.wallet.routes.DIDSchema](#) [attribute](#)), 372
[did_doc](#) ([aries_cloudagent.protocols.connections.v1_0.models.connection.v1_0.model.ConnectionDetailSchema](#)
[attribute](#)), 143
[did_doc](#) ([aries_cloudagent.protocols.connections.v1_0.models.connection.v1_0.model.ConnectionDetailSchema](#)
[attribute](#)), 144
[did_doc_attach](#) ([aries_cloudagent.protocols.didexchange.v1_0.messages.request.DIDRequestSchema](#) [attribute](#)), 182
[did_doc_attach](#) ([aries_cloudagent.protocols.didexchange.v1_0.messages.response.DIDResponseSchema](#) [attribute](#)), 182
[did_key_to_naked\(\)](#) (in module
[aries_cloudagent.wallet.util](#)), 374
[did_to_nym\(\)](#) ([aries_cloudagent.ledger.base.BaseLedger](#)
[method](#)), 64
[DIDCommPrefix](#) (class in
[aries_cloudagent.protocols.didcomm_prefix](#)), 304
[DIDDoc](#) (class in [aries_cloudagent.connections.models.diddoc](#)), 22
[DIDDoc](#) (class in [aries_cloudagent.connections.models.diddoc.diddoc](#)), 25
[diddoc_connection_targets\(\)](#)
[\(aries_cloudagent.connections.base_manager.BaseConnectionManager\)](#) (class in
[method](#)), 38
[DIDInfo](#) (class in [aries_cloudagent.wallet.base](#)), 360
[DIDRequestSchema](#) (class in [aries_cloudagent.messaging.valid](#)), 114
[DIDResponseSchema](#) (class in
[aries_cloudagent.wallet.routes](#)), 372
[ServiceSchema](#) (class in
[aries_cloudagent.wallet.routes](#)), 372

[188](#)
 Disclose.Meta (class in [aries_cloudagent.connections.models.diddoc.PublicKeyType](#)
[aries_cloudagent.protocols.discovery.v1_0.messages.disclose](#)), 24
[188](#)
 DiscloseHandler (class in [aries_cloudagent.messaging.decorators.trace_decorator](#)
[aries_cloudagent.protocols.discovery.v1_0.handlers.disclose_handler](#)), 94
[188](#)
 DiscloseSchema (class in [aries_cloudagent.messaging.decorators.trace_decorator](#)
[aries_cloudagent.protocols.discovery.v1_0.messages.disclose](#)), 352
[189](#)
 DiscloseSchema.Meta (class in [aries_cloudagent.wallet.crypto.PackRecipientsSchema](#)
[aries_cloudagent.protocols.discovery.v1_0.messages.disclose](#)), 361
[189](#)
 dispatch_complete() (aries_cloudagent.core.conductor.Conductor
 method), 39
 dispatch_complete() (aries_cloudagent.transport.inbound.manager.InboundTransportManager
 method), 331
 Dispatcher (class in [aries_cloudagent.transport.inbound.session.InboundSession](#)
[aries_cloudagent.core.dispatcher](#)), 40
 DispatcherResponder (class in [aries_cloudagent.wallet.crypto](#)
[aries_cloudagent.core.dispatcher](#)), 42
 dns_resolvehost_end() (aries_cloudagent.transport.stats.StatsTracer
 method), 346
 dns_resolvehost_start() (aries_cloudagent.transport.stats.StatsTracer
 method), 346
 doc(aries_cloudagent.messaging.jsonld.routes.SignRequestSchema attribute), 361
 doc(aries_cloudagent.messaging.jsonld.routes.VerifyRequestSchema attribute), 294
 done (aries_cloudagent.cache.base.CacheKeyLock attribute), 9
 drain() (aries_cloudagent.utils.task_queue.TaskQueue
 method), 354
 DriverMenuService (class in [aries_cloudagent.connections.models.diddoc.service.Service](#)
[aries_cloudagent.protocols.actionmenu.v1_0.driver_menu](#)), 131
 DroppedAttributeException, 96
 dump_decorators() (aries_cloudagent.messaging.agent_message.AgentMessageSchema
 method), 109
E
 ED25519_SIG_2018 (aries_cloudagent.connections.models.diddoc.PublicKeyType
 attribute), 28
 ED25519_SIG_2018 (aries_cloudagent.connections.models.diddoc.PublicKeyType
 attribute), 24
 EDDSA_SA_SIG_SECP256K1 (aries_cloudagent.connections.models.diddoc.PublicKeyType
 attribute), 28
 EDDSA_SA_SIG_SECP256K1 (aries_cloudagent.connections.models.diddoc.PublicKeyType
 attribute), 114

`endpoint_type` (`aries_cloudagent.ledger.routes.QuerySchema.EndpointTypeSchema`), 71
`endpoint_type` (`aries_cloudagent.ledger.routes.QuerySchema.EndpointTypeSchema`), 114
`endpoint_type` (`aries_cloudagent.wallet.routes.DIDEndpointWithTypesSchema`), 114
`endpoint_type` (`aries_cloudagent.wallet.routes.DIDEndpointWithTypesSchema`), 114
`EndpointsResultSchema` (class in `EXAMPLE` (`aries_cloudagent.messaging.valid.Base64URL` `aries_cloudagent.protocols.connections.v1_0.routes`), attribute), 114
`152` `EXAMPLE` (`aries_cloudagent.messaging.valid.Base64URLNoPad` attribute), 114
`EndpointType` (class in `EXAMPLE` (`aries_cloudagent.messaging.valid.DIDKey` `aries_cloudagent.ledger.endpoint_type`), attribute), 114
`66` `EXAMPLE` (`aries_cloudagent.messaging.valid.DIDPosture` attribute), 114
`EndpointType` (class in `EXAMPLE` (`aries_cloudagent.messaging.valid.DIDPosture` `aries_cloudagent.messaging.valid`), attribute), 114
`EndpointTypeName` (class in `EXAMPLE` (`aries_cloudagent.messaging.valid.Endpoint` `aries_cloudagent.ledger.endpoint_type`), attribute), 115
`66` `EXAMPLE` (`aries_cloudagent.messaging.valid.EndpointType` attribute), 115
`enqueue()` (`aries_cloudagent.transport.queue.base.BaseMessageQueue` attribute), 115
`method`), 343 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyCredDefId` attribute), 115
`enqueue()` (`aries_cloudagent.transport.queue.basic.BasicMessageQueue` attribute), 115
`method`), 344 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyCredRevId` attribute), 115
`enqueue_message()` (`aries_cloudagent.transport.outbound.manager.OutboundTransportManager` attribute), 115
`method`), 339 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyDID` attribute), 115
`enqueue_webhook()` (`aries_cloudagent.transport.outbound.manager.OutboundTransportManager` attribute), 115
`method`), 339 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyExtraWQL` attribute), 115
`epoch_to_str()` (in `module` attribute), 115
`aries_cloudagent.messaging.util`), 113 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyISO8601DateTime` attribute), 115
`error_code` (`aries_cloudagent.messaging.error.MessageParseError` attribute), 115
`attribute`), 110 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyPredicate` attribute), 116
`error_code` (`aries_cloudagent.messaging.error.MessagePrepareError` attribute), 116
`attribute`), 110 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyRawPublicKey` attribute), 116
`error_code` (`aries_cloudagent.transport.error.WireFormatEncodeError` attribute), 116
`attribute`), 344 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyRevRegId` attribute), 116
`error_code` (`aries_cloudagent.transport.error.WireFormatParseError` attribute), 116
`attribute`), 345 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyRevRegSize` attribute), 116
`error_msg` (`aries_cloudagent.connections.models.conn_record.ConnRecordSchema` attribute), 35
`attribute`), 35 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyVersion` attribute), 116
`error_msg` (`aries_cloudagent.protocols.issue_credential.v1_0.models.issue_cred_v1_0.exchange.V10CredentialExchangeSchema` attribute), 212
`attribute`), 212 `EXAMPLE` (`aries_cloudagent.messaging.valid.IndyWQL` attribute), 116
`error_msg` (`aries_cloudagent.protocols.issue_credential.v2_0.models.issue_cred_v2_0.exchange.V20CredExRecordSchema` attribute), 238
`attribute`), 238 `EXAMPLE` (`aries_cloudagent.messaging.valid.IntEpoch` attribute), 116
`error_msg` (`aries_cloudagent.protocols.present_proof.v1_0.models.present_proof_v1_0.exchange.V10PresentationExchangeSchema` attribute), 273
`attribute`), 273 `EXAMPLE` (`aries_cloudagent.messaging.valid.JSONWebToken` attribute), 116
`error_msg` (`aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecordSchema` attribute), 310
`attribute`), 310 `EXAMPLE` (`aries_cloudagent.messaging.valid.JWSHeaderKid` attribute), 116
`errormsg` (`aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_v1_0.Schema` attribute), 125
`attribute`), 125 `EXAMPLE` (`aries_cloudagent.messaging.valid.NaturalNumber` attribute), 117
`errormsg` (`aries_cloudagent.protocols.actionmenu.v1_0.routes.MenuInSchema` attribute), 133
`attribute`), 133 `EXAMPLE` (`aries_cloudagent.messaging.valid.NumericStrNatural` attribute), 117
`escalation_uri` (`aries_cloudagent.protocols.problem_report.v1_0.models.problem_report_v1_0.ProblemReportSchema` attribute), 286
`attribute`), 286 `EXAMPLE` (`aries_cloudagent.messaging.valid.NumericStrWhole` attribute), 117
`establish_inbound()` (`aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager` attribute), 117
`method`), 148 `EXAMPLE` (`aries_cloudagent.messaging.valid.SHA256Hash` attribute), 117

EXAMPLE (aries_cloudagent.messaging.valid.UUIDFour attribute), 117	method), 83
EXAMPLE (aries_cloudagent.messaging.valid.WholeNumber attribute), 117	extract_pack_recipients() (in module aries_cloudagent.wallet.crypto), 363
execute() (in module aries_cloudagent.commands.help), 10	extract_payload_key() (in module aries_cloudagent.wallet.crypto), 363
execute() (in module aries_cloudagent.commands.provision), 10	F
execute() (in module aries_cloudagent.commands.start), 11	fetch() (aries_cloudagent.storage.base.BaseStorageSearchSession method), 322
EXISTING_CONNECTION_DOES_NOT_EXISTS (aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportReason attribute), 252	fetch() (aries_cloudagent.storage.in_memory.InMemoryStorageSearchSession method), 324
EXISTING_CONNECTION_NOT_ACTIVE (aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportReason attribute), 252	fetch() (aries_cloudagent.storage.indy.IndySdkStorageSearchSession method), 326
exists_for_connection_id() (aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord class method), 169	fetch() (in module aries_cloudagent.config.ledger), 19
expire_messages() (aries_cloudagent.transport.inbound.delivery_queue.DeliveryQueue method), 329	fetch_connection_targets() (aries_cloudagent.connections.base_manager.BaseConnectionManager method), 38
expires_time (aries_cloudagent.messaging.decorators.timing_decorator.TimingDecorator attribute), 91	fetch_credential_definition() (aries_cloudagent.ledger.indy.IndySdkLedger method), 68
explain (aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportSchema attribute), 143	fetch_did_document() (aries_cloudagent.connections.base_manager.BaseConnectionManager method), 38
explain (aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportSchema attribute), 181	fetch_error() (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator module aries_cloudagent.config.ledger), 19
explain (aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportSchema attribute), 252	fetch_schema_by_id() (aries_cloudagent.ledger.indy.IndySdkLedger method), 68
explain_l10n (aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReportSchema attribute), 286	fetch_schema_by_seq_no() (aries_cloudagent.ledger.indy.IndySdkLedger method), 68
explain_ltxt (aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_request.IssueCredentialRequest attribute), 216	fetch_txns() (aries_cloudagent.ledger.indy.IndySdkLedger method), 64
explain_ltxt (aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential_request.IssueCredentialRequest attribute), 243	fetch_txn_author_agreement() (aries_cloudagent.ledger.indy.IndySdkLedger method), 68
explain_ltxt (aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof.PresentationProblemReportRequestSchema attribute), 280	FetchError, 349
explain_ltxt (aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReportSchema attribute), 286	field() (aries_cloudagent.messaging.decorators.base.BaseDecoratorSet method), 83
extend() (aries_cloudagent.config.base.BaseSettings method), 15	fields (aries_cloudagent.messaging.decorators.base.BaseDecoratorSet attribute), 83
extend() (aries_cloudagent.config.settings.Settings method), 21	file_ext() (in module aries_cloudagent.indy.sdk.wallet_plugin), 57
extra_query (aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof.PresentationProblemReportRequestSchema attribute), 278	filename (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator attribute), 83
extract() (aries_cloudagent.utils.stats.Collector method), 352	filter (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.problem_report.ProblemReportSchema attribute), 163
extract() (aries_cloudagent.utils.stats.Stats method), 353	filter (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.problem_report.ProblemReportSchema attribute), 174
extract_decorators() (aries_cloudagent.messaging.agent_message.AgentMessageSchema method), 109	
extract_decorators() (aries_cloudagent.messaging.agent_message.AgentMessageSchema method), 109	

filter (aries_cloudagent.protocols.routing.v1_0.messages.route_query_string_schema.RouteRequestSchema attribute), 290	format (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 226
filter () (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 230	format_cred_proposal (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredProposal attribute), 226
filter_ (aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredentialCoreRoutes attribute), 244	format_cred_revoked_query_string_schema (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredRevokedQueryStringSchema attribute), 372
filters_attach (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredProposal attribute), 231	format_wallet_record () (in module aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredProposalSchema), 119
final (aries_cloudagent.utils.repeat.RepeatAttempt attribute), 351	formats (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 228
find_all_records () (aries_cloudagent.storage.base.BaseStorage method), 320	formats (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 229
find_all_records () (aries_cloudagent.storage.in_memory.InMemoryStorage method), 323	formats (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 231
find_all_records () (aries_cloudagent.storage.indy.IndySdkStorage method), 325	formats (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 232
find_connection () (aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager method), 148	FormatSpec (class in aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat), 225
find_connection () (aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager method), 148	fortran (aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate attribute), 275
find_did_for_key () (aries_cloudagent.connections.base_manager.BaseConnectionManager method), 38	Forward (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
find_existing_connection () (aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager method), 258	ForwardInvitation (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
find_inbound_connection () (aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager method), 148	ForwardInvitationHandler (class in aries_cloudagent.protocols.routing.v1_0.handlers.forward_invitation.InvitationHandler), 287
find_record () (aries_cloudagent.storage.base.BaseStorage method), 321	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
finished_deliver () (aries_cloudagent.transport.outbound.manager.OutboundTransportManager method), 339	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
finished_encode () (aries_cloudagent.transport.outbound.manager.OutboundTransportManager method), 339	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
fix_hint_txt (aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReport attribute), 286	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
flip () (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 32	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
flush () (aries_cloudagent.cache.base.BaseCache method), 8	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
flush () (aries_cloudagent.cache.in_memory.InMemoryCache method), 9	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
flush () (aries_cloudagent.transport.outbound.manager.OutboundTransportManager method), 339	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
flush () (aries_cloudagent.utils.task_queue.TaskQueue method), 354	ForwardInvitationSchema (class in aries_cloudagent.protocols.routing.v1_0.messages.forward_invitation.Invitation), 288
form (aries_cloudagent.protocols.actionmenu.v1_0.models.menu_option.MenuOption attribute), 130	fro (aries_cloudagent.holder.routes.CredRevokedQueryStringSchema attribute), 372
format (aries_cloudagent.protocols.issue_credential.v2_0.messages.credentials.V20CredFormat attribute), 226	fro (aries_cloudagent.revocation.models.indy.NonRevocationIntervalSchema attribute), 304

[from_definition\(\)](#) ([aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry](#) class method), 312
[from_json\(\)](#) ([aries_cloudagent.connections.models.diddoc.DIDDoc](#) class method), 8
[from_json\(\)](#) ([aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc](#) class method), 23
[from_json\(\)](#) ([aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc](#) class method), 26
[from_json\(\)](#) ([aries_cloudagent.messaging.models.base.BaseModel](#) class method), 98
[from_storage\(\)](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) class method), 101
[from_url\(\)](#) ([aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitation](#) class method), 139
[from_url\(\)](#) ([aries_cloudagent.protocols.out_of_band.v1_0.messages.aries_cloudagent_protocol_message.AriesCloudAgentProtocolMessage](#) class method), 250
[full_thread\(\)](#) ([aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator](#) (in module [aries_cloudagent.revocation.routes](#)), 318
[full_thread\(\)](#) ([aries_cloudagent.messaging.decorators.trace_decorator.TraceDecoratorSchema](#) (in module [aries_cloudagent.ledger.indy.IndySdkLedger](#) method), 68
[full_verkey\(\)](#) (in module [aries_cloudagent.wallet.util](#)), 374
[future](#) ([aries_cloudagent.cache.base.CacheKeyLock](#) attribute), 9
G
[GE](#) ([aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate](#) attribute), 275
[GeneralGroup](#) (class in [aries_cloudagent.config.argparse](#)), 12
[generate_pr_nonce\(\)](#) (in module [aries_cloudagent.indy.util](#)), 62
[generate_registry\(\)](#) ([aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord](#) class method), 309
[generate_wallet_key\(\)](#) ([aries_cloudagent.wallet.indy.IndySdkWallet](#) class method), 369
[get](#) ([aries_cloudagent.connections.models.conn_record.ConnRecord](#) attribute), 32
[get](#) ([aries_cloudagent.connections.models.conn_record.ConnRecord.State](#) attribute), 32
[get](#) ([aries_cloudagent.connections.models.diddoc.publickey.PublicKeyType](#) attribute), 28
[get](#) ([aries_cloudagent.connections.models.diddoc.PublicKeyType](#) attribute), 24
[get](#) ([aries_cloudagent.ledger.base.Role](#) attribute), 66
[get](#) ([aries_cloudagent.ledger.endpoint_type.EndpointType](#) attribute), 66
[get](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format_v20.CredFormat](#) attribute), 226
[get](#) ([aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.Invitation](#) attribute), 249
[get](#) ([aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate](#) attribute), 275
[get](#) ([aries_cloudagent.wallet.did_posture.DIDPosture](#) attribute), 364
[get](#) () ([aries_cloudagent.cache.base.BaseCache](#) class method), 314
[get](#) () ([aries_cloudagent.cache.in_memory.InMemoryCache](#) class method), 314
[get_active_issuer_rev_reg_record\(\)](#) ([aries_cloudagent.revocation.indy.IndyRevocation](#) method), 314
[get_active_menu\(\)](#) ([aries_cloudagent.protocols.actionmenu.v1_0.base_service.BaseService](#) method), 132
[get_active_menu\(\)](#) ([aries_cloudagent.protocols.actionmenu.v1_0.driver_service.DriverService](#) method), 132
[get_decorator\(\)](#) ([aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator](#) (in module [aries_cloudagent.revocation.routes](#)), 318
[get_decorator\(\)](#) ([aries_cloudagent.messaging.decorators.trace_decorator.TraceDecoratorSchema](#) (in module [aries_cloudagent.ledger.indy.IndySdkLedger](#) method), 68
[get_attachment_data\(\)](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_format_v20.CredFormat](#) class method), 226
[get_bool\(\)](#) ([aries_cloudagent.config.base.BaseSettings](#) class method), 15
[get_cached_key\(\)](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) class method), 101
[get_connection_targets\(\)](#) ([aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager](#) method), 148
[get_cred_rev_record\(\)](#) (in module [aries_cloudagent.revocation.routes](#)), 318
[get_cred_rev_record\(\)](#) ([aries_cloudagent.indy.holder.IndyHolder](#) method), 59
[get_credential\(\)](#) ([aries_cloudagent.indy.sdk.holder.IndySdkHolder](#) method), 52
[get_credential_definition\(\)](#) ([aries_cloudagent.ledger.base.BaseLedger](#) method), 64
[get_credential_definition\(\)](#) ([aries_cloudagent.ledger.indy.IndySdkLedger](#) method), 68
[get_credentials\(\)](#) ([aries_cloudagent.indy.sdk.holder.IndySdkHolder](#) method), 52
[get_credentials_for_presentation_request_by_referent\(\)](#) ([aries_cloudagent.indy.sdk.holder.IndySdkHolder](#) method), 52
[get_default_mediator\(\)](#) ([aries_cloudagent.multitenant.manager.MultitenantManager](#) method), 121
[get_default_mediator\(\)](#) ([aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.CoordinateMediationManager](#) method), 171

`get_default_mediator()` (in module `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager`), 176
`get_default_mediator_id()` (in module `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager`), 171
`get_detail_record()` (in module `aries_cloudagent.protocols.issue_credential.v2_0.manager`), 239
`get_did_endpoint()` (in module `aries_cloudagent.ledger.routes`), 72
`get_did_verkey()` (in module `aries_cloudagent.ledger.routes`), 72
`get_endpoint_for_did()` (in module `aries_cloudagent.ledger.base.BaseLedger`), 64
`get_endpoint_for_did()` (in module `aries_cloudagent.ledger.indy.IndySdkLedger`), 68
`get_endpoints()` (in module `aries_cloudagent.protocols.connections.v1_0.manager`), 148
`get_genesis_transactions()` (in module `aries_cloudagent.config.ledger`), 19
`get_indy_storage()` (in module `aries_cloudagent.ledger.indy.IndySdkLedger`), 68
`get_int()` (in module `aries_cloudagent.config.base.BaseSettings`), 15
`get_issuer_rev_reg_record()` (in module `aries_cloudagent.revocation.indy.IndyRevocation`), 314
`get_key_for_did()` (in module `aries_cloudagent.ledger.base.BaseLedger`), 64
`get_key_for_did()` (in module `aries_cloudagent.ledger.indy.IndySdkLedger`), 69
`get_keylist()` (in module `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager`), 172
`get_keylist()` (in module `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager`), 176
`get_latest_txn_author_acceptance()` (in module `aries_cloudagent.ledger.base.BaseLedger`), 64
`get_latest_txn_author_acceptance()` (in module `aries_cloudagent.ledger.indy.IndySdkLedger`), 69
`get_ledger_registry()` (in module `aries_cloudagent.revocation.indy.IndyRevocation`), 314
`get_local_did()` (in module `aries_cloudagent.wallet.base.BaseWallet`), 358
`get_local_did()` (in module `aries_cloudagent.wallet.in_memory.InMemoryWallet`), 358
`get_local_did_for_verkey()` (in module `aries_cloudagent.wallet.indy.IndySdkWallet`), 369
`get_local_dids()` (in module `aries_cloudagent.wallet.base.BaseWallet`), 358
`get_local_dids()` (in module `aries_cloudagent.wallet.in_memory.InMemoryWallet`), 366
`get_local_dids()` (in module `aries_cloudagent.wallet.indy.IndySdkWallet`), 369
`get_mime_type()` (in module `aries_cloudagent.indy.holder.IndyHolder`), 52
`get_mime_type()` (in module `aries_cloudagent.indy.sdk.holder.IndySdkHolder`), 52
`get_my_keylist()` (in module `aries_cloudagent.protocols.coordinate_mediation.v1_0.manager`), 172
`get_nym_role()` (in module `aries_cloudagent.ledger.base.BaseLedger`), 64
`get_nym_role()` (in module `aries_cloudagent.ledger.indy.IndySdkLedger`), 69
`get_nym_role()` (in module `aries_cloudagent.ledger.routes`), 72
`get_one_message_for_key()` (in module `aries_cloudagent.transport.inbound.delivery_queue.DeliveryQueue`), 329
`get_or_fetch_local_tails_path()` (in module `aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry`), 312
`get_posted_dids()` (in module `aries_cloudagent.wallet.base.BaseWallet`), 358
`get_profile_for_token()` (in module `aries_cloudagent.multitenant.manager.MultitenantManager`), 121
`get_provider()` (in module `aries_cloudagent.config.injector.Injector`), 18
`get_public_did()` (in module `aries_cloudagent.wallet.base.BaseWallet`), 358
`get_receiving_tails_local_path()` (in module `aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry`), 313
`get_recipient()` (in module `aries_cloudagent.protocols.routing.v1_0.manager`), 299
`get_recipient_keys()` (in module `aries_cloudagent.transport.pack_format.PackWireFormat`), 345

`get_recipient_keys()` (`aries_cloudagent.transport.wire_format.BaseWireFormat` method), 347
`get_recipient_keys()` (`aries_cloudagent.transport.wire_format.JsonWireFormat` method), 347
`get_record()` (`aries_cloudagent.storage.base.BaseStorage` method), 321
`get_record()` (`aries_cloudagent.storage.in_memory.InMemoryStorage` method), 323
`get_record()` (`aries_cloudagent.storage.indy.IndySdkStorage` method), 325
`get_registered()` (`aries_cloudagent.config.argsparse.group` class method), 14
`get_registered_transport_for_scheme()` (`aries_cloudagent.transport.outbound.manager.OutboundTransportManager` method), 339
`get_registry()` (`aries_cloudagent.revocation.models.issue_revoked()` method), 309
`get_rev_reg()` (`aries_cloudagent.revocation.routes` in module), 318
`get_rev_reg_issued()` (`aries_cloudagent.revocation.routes` in module), 319
`get_revoc_reg_def()` (`aries_cloudagent.ledger.base.BaseLedger` method), 64
`get_revoc_reg_def()` (`aries_cloudagent.ledger.indy.IndySdkLedger` method), 69
`get_revoc_reg_delta()` (`aries_cloudagent.ledger.base.BaseLedger` method), 64
`get_revoc_reg_delta()` (`aries_cloudagent.ledger.indy.IndySdkLedger` method), 69
`get_revoc_reg_entry()` (`aries_cloudagent.ledger.base.BaseLedger` method), 64
`get_revoc_reg_entry()` (`aries_cloudagent.ledger.indy.IndySdkLedger` method), 69
`get_routes()` (`aries_cloudagent.protocols.routing.v1_0.manager.RoutingManager` method), 299
`get_running_transport_for_endpoint()` (`aries_cloudagent.transport.outbound.manager.OutboundTransportManager` method), 339
`get_running_transport_for_scheme()` (`aries_cloudagent.transport.outbound.manager.OutboundTransportManager` method), 339
`get_schema()` (`aries_cloudagent.ledger.base.BaseLedger` method), 64
`get_schema()` (`aries_cloudagent.ledger.indy.IndySdkLedger` method), 69
`get_settings()` (`aries_cloudagent.config.argsparse.AdminGroup` method), 11
`get_settings()` (`aries_cloudagent.config.argsparse.ArgumentParser` method), 11
`get_settings()` (`aries_cloudagent.config.argsparse.DebugGroup` method), 12
`get_settings()` (`aries_cloudagent.config.argsparse.GeneralGroup` method), 12
`get_settings()` (`aries_cloudagent.config.argsparse.LedgerGroup` method), 12
`get_settings()` (`aries_cloudagent.config.argsparse.LoggingGroup` method), 12
`get_settings()` (`aries_cloudagent.config.argsparse.MediationGroup` method), 13
`get_settings()` (`aries_cloudagent.config.argsparse.MultitenantGroup` method), 13
`get_settings()` (`aries_cloudagent.config.argsparse.ProtocolGroup` method), 13
`get_settings()` (`aries_cloudagent.config.argsparse.StartupGroup` method), 13
`get_settings()` (`aries_cloudagent.config.argsparse.TransportGroup` method), 13
`get_settings()` (`aries_cloudagent.config.argsparse.WalletGroup` method), 14
`get_signature()` (`aries_cloudagent.messaging.agent_message.AgentMessage` method), 107
`get_signing_key()` (`aries_cloudagent.wallet.base.BaseWallet` method), 359
`get_signing_key()` (`aries_cloudagent.wallet.in_memory.InMemoryWallet` method), 366
`get_signing_key()` (`aries_cloudagent.wallet.indy.IndySdkWallet` method), 369
`get_stats()` (`aries_cloudagent.core.conductor.Conductor` method), 39
`get_str()` (`aries_cloudagent.config.base.BaseSettings` method), 15
`get_tag_map()` (`aries_cloudagent.messaging.models.base_record.BaseRecord` class method), 101
`get_tails_file()` (`aries_cloudagent.revocation.routes` in module), 319
`get_timer()` (`aries_cloudagent.utils.tracing` in module), 355
`get_transport_for_endpoint()` (`aries_cloudagent.transport.inbound.manager.InboundTransportManager` method), 331
`get_transport_for_scheme()` (`aries_cloudagent.transport.outbound.manager.OutboundTransportManager` method), 340
`get_txn_author_agreement()` (`aries_cloudagent.ledger.base.BaseLedger` method), 65
`get_txn_author_agreement()` (`aries_cloudagent.ledger.indy.IndySdkLedger` method), 65

<code>(aries_cloudagent.ledger.indy.IndySdkLedger</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_</code>
<code>method), 69</code>		<code>method), 124</code>
<code>get_value() (aries_cloudagent.config.base.BaseSettings</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.actionmenu.v1_0.handlers.perform</code>
<code>method), 15</code>		<code>method), 124</code>
<code>get_value() (aries_cloudagent.config.settings.Settings</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.basicmessage.v1_0.handlers.basic</code>
<code>method), 21</code>		<code>method), 135</code>
<code>get_wallet_profile()</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.connections.v1_0.handlers.conne</code>
<code>(aries_cloudagent.multitenant.manager.MultitenantManagem</code>	<code>method), 137</code>	
<code>method), 122</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.connections.v1_0.handlers.conne</code>
<code>get_wallets_by_message()</code>		<code>method), 138</code>
<code>(aries_cloudagent.multitenant.manager.MultitenantManager</code>	<code>method), 138</code>	
<code>method), 122</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.coordinate_mediation.v1_0.handle</code>
<code>get_webhook_urls()</code>		<code>method), 156</code>
<code>(aries_cloudagent.multitenant.manager.MultitenantManagem</code>	<code>method), 122</code>	<code>handle()</code>
<code>method), 122</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.coordinate_mediation.v1_0.handle</code>
<code>GetKeylistQuerySchema</code>	<code>(class in</code>	<code>method), 156</code>
<code>aries_cloudagent.protocols.coordinate_mediation.v1_0.routes</code>	<code>aries_cloudagent.protocols.coordinate_mediation.v1_0.handle</code>	
<code>174</code>	<code>method), 156</code>	
<code>grant_request() (aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.coordinate_mediation.v1_0.handle</code>	<code>method), 172</code>	
<code>method), 172</code>	<code>handle()</code>	<code>(aries_cloudagent.protocols.coordinate_mediation.v1_0.handle</code>
<code>group (class in aries_cloudagent.config.argparse), 14</code>		<code>method), 157</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.AdminGroup</code>	<code>attribute), 11</code>	<code>handle()</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.ArgumentGroup</code>	<code>attribute), 11</code>	<code>method), 157</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.DebugGroup</code>	<code>attribute), 12</code>	<code>handle()</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.GeneralGroup</code>	<code>attribute), 12</code>	<code>method), 178</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.LedgerGroup</code>	<code>attribute), 12</code>	<code>handle()</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.LoggingGroup</code>	<code>attribute), 12</code>	<code>method), 179</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.MediationGroup</code>	<code>attribute), 12</code>	<code>handle()</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.MultitenantGroup</code>	<code>attribute), 13</code>	<code>method), 188</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.ProtocolGroup</code>	<code>attribute), 13</code>	<code>handle()</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.StartupGroup</code>	<code>attribute), 13</code>	<code>method), 193</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.TransportGroup</code>	<code>attribute), 13</code>	<code>handle()</code>
<code>GROUP_NAME (aries_cloudagent.config.argparse.WalletGroup</code>	<code>attribute), 14</code>	<code>method), 193</code>
<code>GT (aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate</code>	<code>method), 275</code>	<code>handle()</code>
		<code>(aries_cloudagent.protocols.issue_credential.v1_0.handlers.cre</code>
		<code>method), 199</code>
H	<code>handle()</code>	<code>(aries_cloudagent.protocols.issue_credential.v1_0.handlers.cre</code>
<code>handle() (aries_cloudagent.messaging.base_handler.BaseHandler</code>	<code>method), 109</code>	<code>handle()</code>
<code>handle() (aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_handler.MenuHandler</code>	<code>method), 123</code>	<code>handle()</code>
		<code>(aries_cloudagent.protocols.issue_credential.v1_0.handlers.cre</code>
		<code>method), 200</code>

`handle()` (`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_ack_handler.V20CredAckHandler`
`method`), 221
`handle()` (`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_issue_handler.V20CredIssueHandler`
`method`), 221
`handle()` (`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_offer_handler.V20CredOfferHandler`
`method`), 221
`handle()` (`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_proposal_handler.V20CredProposalHandler`
`method`), 222
`handle()` (`aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_request_handler.V20CredRequestHandler`
`method`), 222
`handle()` (`aries_cloudagent.protocols.out_of_band.v1_0.handlers.problem_report_handler.OutOfBandProblemReportHandler`
`method`), 247
`handle()` (`aries_cloudagent.protocols.out_of_band.v1_0.handlers.resume_accept_handler.OutOfBandResumeAcceptHandler`
`method`), 248
`handle()` (`aries_cloudagent.protocols.out_of_band.v1_0.handlers.resume_decline_handler.OutOfBandResumeDeclineHandler`
`method`), 248
`handle()` (`aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack_handler.PresentationAckHandler`
`method`), 262
`handle()` (`aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_request_handler.PresentationRequestHandler`
`method`), 262
`handle()` (`aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_revocation_handler.PresentationRevocationHandler`
`method`), 263
`handle()` (`aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_request_handler.PresentationRequestHandler`
`method`), 263
`handle()` (`aries_cloudagent.protocols.problem_report.v1_0.handlers.problem_report_handler.OutOfBandProblemReportHandler`
`method`), 284
`handle()` (`aries_cloudagent.protocols.routing.v1_0.handlers.forward_handler.ForwardHandler`
`method`), 287
`handle()` (`aries_cloudagent.protocols.routing.v1_0.handlers.route_query_request_handler.RouteQueryRequestHandler`
`method`), 287
`handle()` (`aries_cloudagent.protocols.routing.v1_0.handlers.route_query_response_handler.RouteQueryResponseHandler`
`method`), 287
`handle()` (`aries_cloudagent.protocols.routing.v1_0.handlers.route_update_request_handler.RouteUpdateRequestHandler`
`method`), 288
`handle()` (`aries_cloudagent.protocols.routing.v1_0.handlers.route_update_response_handler.RouteUpdateResponseHandler`
`method`), 288
`handle()` (`aries_cloudagent.protocols.trustping.v1_0.handlers.ping_handler.PingHandler`
`method`), 301
`handle()` (`aries_cloudagent.protocols.trustping.v1_0.handlers.ping_response_handler.PingResponseHandler`
`method`), 301
`handle_message()` (`aries_cloudagent.core.dispatcher.Dispatcher`
`method`), 41
`handle_message()` (`aries_cloudagent.transport.outbound.base.OutboundBase`
`method`), 337
`handle_message()` (`aries_cloudagent.transport.outbound.http.HttpTransport`
`method`), 338
`handle_message()` (`aries_cloudagent.transport.outbound.websocket.WebsocketTransport`
`method`), 342
`handle_not_delivered()`
`(aries_cloudagent.core.conductor.Conductor`
`method`), 39
`handle_not_returned()`
`(aries_cloudagent.core.conductor.Conductor`
`method`), 39

[handler_class \(aries_cloudagent.protocols.discovery.v1_0.messages.query.Query.Meta attribute\), 191](#)
[handler_class \(aries_cloudagent.protocols.discovery.v1_0.messages.query.Query.Meta attribute\), 302](#)
[handler_class \(aries_cloudagent.protocols.introduction.v0_1.messages.forward_forwards_incoming_forwarding_incoming.Meta attribute\), 194](#)
[handler_class \(aries_cloudagent.protocols.introduction.v0_1.messages.invitation.Invitation.Meta attribute\), 195](#)
[handler_class \(aries_cloudagent.protocols.introduction.v0_1.messages.invitation.Invitation.Meta attribute\), 251](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_ack.CredentialAck.Meta attribute\), 203](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_issue.CredentialIssue.Meta attribute\), 204](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_reuse.CredentialReuse.Meta attribute\), 205](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_proposal.CredentialProposal.Meta attribute\), 207](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request.CredentialRequest.Meta attribute\), 208](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ack.V20CredAck.Meta attribute\), 225](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue.V20CredIssue.Meta attribute\), 227](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer.V20CredOffer.Meta attribute\), 229](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal.V20CredProposal.Meta attribute\), 230](#)
[handler_class \(aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_request.V20CredRequest.Meta attribute\), 231](#)
[handler_class \(aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReport.Meta attribute\), 252](#)
[handler_class \(aries_cloudagent.protocols.out_of_band.v1_0.messages.handshake_reuse.HandshakeReuse.Meta attribute\), 253](#)
[handler_class \(aries_cloudagent.protocols.out_of_band.v1_0.messages.handshake_reuse_accept.HandshakeReuseAccept.Meta attribute\), 253](#)
[handler_class \(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation.Presentation.Meta attribute\), 268](#)
[handler_class \(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposal.Meta attribute\), 269](#)
[handler_class \(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request.PresentationRequest.Meta attribute\), 270](#)
[handler_class \(aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReport.Meta attribute\), 286](#)
[handler_class \(aries_cloudagent.protocols.routing.v1_0.messages.forward.Forward.Meta attribute\), 288](#)
[handler_class \(aries_cloudagent.protocols.routing.v1_0.messages.route_query.RouteQuery.Meta attribute\), 289](#)
[handler_class \(aries_cloudagent.protocols.routing.v1_0.messages.route_query_response.RouteQueryResponse.Meta attribute\), 290](#)
[handler_class \(aries_cloudagent.protocols.routing.v1_0.messages.route_update_request.RouteUpdateRequest.Meta attribute\), 291](#)
[handler_class \(aries_cloudagent.protocols.routing.v1_0.messages.route_update_response.RouteUpdateResponse.Meta attribute\), 292](#)

<code>method()</code> , 270			<code>IndyRevocation</code> (class in	
<code>IndyCredDefId</code> (class in		<code>aries_cloudagent.revocation.indy)</code> , 314		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>IndyRevRegId</code> (class in		
<code>IndyCredInfoSchema</code> (class in		<code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 278		<code>IndyRevRegSize</code> (class in		
<code>IndyCredPrecisSchema</code> (class in		<code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 278		<code>IndySchemaId</code> (class in		
<code>IndyCredRevId</code> (class in		<code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>IndySdkHolder</code> (class in		
<code>IndyDID</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 115		<code>aries_cloudagent.indy.sdk.holder)</code> , 51		
<code>IndyErrorHandler</code> (class in		<code>IndySdkIssuer</code> (class in		
<code>aries_cloudagent.indy.sdk.error)</code> , 50		<code>aries_cloudagent.indy.sdk.issuer)</code> , 53		
<code>IndyExtraWQL</code> (class in		<code>IndySdkLedger</code> (class in		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>aries_cloudagent.ledger.indy)</code> , 67		
<code>IndyHolder</code> (class in <code>aries_cloudagent.indy.holder)</code> , 58		<code>IndySdkLedgerPool</code> (class in		
<code>IndyHolderError</code> , 60		<code>aries_cloudagent.ledger.indy)</code> , 70		
<code>IndyISO8601DateTime</code> (class in		<code>IndySdkLedgerPoolProvider</code> (class in		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>aries_cloudagent.ledger.indy)</code> , 70		
<code>IndyIssuer</code> (class in <code>aries_cloudagent.indy.issuer)</code> , 60		<code>IndySdkProfile</code> (class in		
<code>IndyIssuerError</code> , 62		<code>aries_cloudagent.indy.sdk.profile)</code> , 55		
<code>IndyIssuerRevocationRegistryFullError</code> , 62		<code>IndySdkProfileManager</code> (class in		
<code>IndyOpenWallet</code> (class in		<code>aries_cloudagent.indy.sdk.profile)</code> , 55		
<code>aries_cloudagent.indy.sdk.wallet_setup)</code> , 57		<code>IndySdkProfileSession</code> (class in		
<code>IndyPredicate</code> (class in		<code>aries_cloudagent.indy.sdk.profile)</code> , 55		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>IndySdkStorage</code> (class in		
<code>IndyProofReqAttrSpecSchema</code> (class in		<code>aries_cloudagent.storage.indy)</code> , 324		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 278		<code>IndySdkStorageSearch</code> (class in		
<code>IndyProofReqNonRevokedSchema</code> (class in		<code>aries_cloudagent.storage.indy)</code> , 326		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 279		<code>IndySdkVerifier</code> (class in		
<code>IndyProofReqPredSpecRestrictionsSchema</code> (class in		<code>aries_cloudagent.indy.sdk.verifier)</code> , 56		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 279		<code>IndySdkWallet</code> (class in		
<code>IndyProofReqPredSpecSchema</code> (class in		<code>aries_cloudagent.wallet.indy)</code> , 368		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 279		<code>IndyTailsServer</code> (class in		
<code>IndyProofRequestSchema</code> (class in		<code>aries_cloudagent.tails.indy_tails_server)</code> , 327		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 279		<code>IndyVerifier</code> (class in		
<code>IndyRawPublicKey</code> (class in		<code>aries_cloudagent.indy.verifier)</code> , 63		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>IndyVersion</code> (class in		
<code>IndyRequestedCredsRequestedAttrSchema</code> (class in		<code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 279		<code>IndyWalletConfig</code> (class in		
<code>IndyRequestedCredsRequestedPredSchema</code> (class in		<code>aries_cloudagent.indy.sdk.wallet_setup)</code> , 57		
<code>aries_cloudagent.protocols.present_proof.v1_0.routes)</code> , 280		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyRevocation</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.revocation.indy)</code> , 314		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyRevRegId</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.messaging.valid)</code> , 115		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyRevRegSize</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySchemaId</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkHolder</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.holder)</code> , 51		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkIssuer</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.issuer)</code> , 53		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkLedger</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.ledger.indy)</code> , 67		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkLedgerPool</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.ledger.indy)</code> , 70		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkLedgerPoolProvider</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.ledger.indy)</code> , 70		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkProfile</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.profile)</code> , 55		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkProfileManager</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.profile)</code> , 55		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkProfileSession</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.profile)</code> , 55		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkStorage</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.storage.indy)</code> , 324		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkStorageSearch</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.storage.indy)</code> , 326		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkVerifier</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.verifier)</code> , 56		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndySdkWallet</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.wallet.indy)</code> , 368		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyTailsServer</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.tails.indy_tails_server)</code> , 327		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyVerifier</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.verifier)</code> , 63		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyVersion</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWalletConfig</code> (class in		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>aries_cloudagent.indy.sdk.wallet_setup)</code> , 57		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		
<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> , 116		<code>IndyWQL</code> (class in <code>aries_cloudagent.messaging.valid)</code> ,		

[illegible]

[aries_cloudagent.protocols.introduction.v0_1.messages.invitation](#),
[194](#)

[InvitationHandler](#) (class in [aries_cloudagent.protocols.introduction.v0_1.handlers.invitation](#),
[194](#)

[Invitation.Meta](#) (class in [aries_cloudagent.protocols.introduction.v0_1.messages.invitation](#),
[195](#)

[InvitationMessage](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[249](#)

[invitation_create\(\)](#) (in module [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[261](#)

[InvitationMessage.Meta](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[250](#)

[invitation_id\(aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation_record.Schema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[257](#)

[invitation_id\(aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation_record.Schema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[257](#)

[invitation_key\(aries_cloudagent.connections.models.conn_record.ConnRecordSchema](#) (class in [aries_cloudagent.connections.v1_0.messages.invitation](#),
[35](#)

[InvitationMessageSchema.Meta](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[251](#)

[invitation_key\(aries_cloudagent.protocols.connections.v1_0.messages.invitation_query_string.Schema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[251](#)

[invitation_mode\(aries_cloudagent.connections.models.conn_record.ConnRecordSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[35](#)

[INVITATION_MODE_MULTI](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[InvitationReceiveRequestSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[31](#)

[INVITATION_MODE_ONCE](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[261](#)

[InvitationRecord](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[31](#)

[INVITATION_MODE_STATIC](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[31](#)

[InvitationRecord.Meta](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[256](#)

[invitation_msg_id](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[35](#)

[InvitationRecordSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[257](#)

[INVITATION_NOT_ACCEPTED](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[142](#)

[InvitationReportProblemReason](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[257](#)

[INVITATION_NOT_ACCEPTED](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[180](#)

[InvitationReportProblemReason](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[180](#)

[invitation_receive\(\)](#) (in module [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[261](#)

[InvitationRequest.Meta](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[196](#)

[invitation_url\(aries_cloudagent.protocols.connections.v1_0.messages.invitation_result.Schema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[152](#)

[InvitationRequestHandler](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[193](#)

[invitation_url\(aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation_result.Schema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[257](#)

[InvitationCreateQueryStringSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[InvitationCreateRequestSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[InvitationCreateRequestSchema.Meta](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[InvitationCreateRequestSchema.AttachmentDefSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[InvitationCreateRequestSchema.ResultSchema](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[InvitationHandler](#) (class in [aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation](#),
[260](#)

[195](#)
 InvitationSchema.Meta (class in [aries_cloudagent.revocation.models.issuer_rev_reg_record](#)),
[aries_cloudagent.protocols.introduction.v0_1.messages.invitation](#)
[195](#)
 invite_message_handler() [309](#)
 (aries_cloudagent.transport.inbound.http.HttpTransport
 method), [330](#)
 is_managed(aries_cloudagent.wallet.models.wallet_record.WalletRecord
 attribute), [356](#)
 is_multiuse_invitation
 (aries_cloudagent.connections.models.conn_record.ConnRecord
 attribute), [33](#)
 is_ready(aries_cloudagent.connections.models.conn_record.ConnRecord
 attribute), [33](#)
 is_transaction(aries_cloudagent.core.profile.ProfileSession
 attribute), [47](#)
 issue_credential()
 (aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential
 method), [213](#)
 issue_credential()
 (aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential
 method), [239](#)
 IssueCredentialModuleResponseSchema
 (class in aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential
[215](#)
 issuer_did(aries_cloudagent.messaging.credential_definition_util.CredDefQueryStringSchema
 attribute), [76](#)
 issuer_did(aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_proposal
 attribute), [207](#)
 issuer_did(aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_proposal
 attribute), [215](#)
 issuer_did(aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_proposal
 attribute), [217](#)
 issuer_did(aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential
 attribute), [242](#)
 issuer_did(aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof_req_pred_spec_restrictions
 attribute), [279](#)
 issuer_did(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecordSchema
 attribute), [310](#)
 issuer_did(aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry
 attribute), [313](#)
 IssuerCredRevRecord (class in [aries_cloudagent.revocation.models.issuer_cred_rev_record](#)),
[305](#)
 IssuerCredRevRecord.Meta (class in [aries_cloudagent.revocation.models.issuer_cred_rev_record](#)),
[306](#)
 IssuerCredRevRecordSchema (class in [aries_cloudagent.revocation.models.issuer_cred_rev_record](#)),
[307](#)
 IssuerCredRevRecordSchema.Meta (class in [aries_cloudagent.revocation.models.issuer_cred_rev_record](#)),
[307](#)
 IssuerRevRegRecord (class in [aries_cloudagent.revocation.models.issuer_rev_reg_record](#)),
[307](#)

KEY_DERIVATION_ARGON2I_MOD	KeylistQueryPaginateSchema (class in	
(aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
attribute), 58	159	
KEY_DERIVATION_RAW	KeylistQueryPaginateSchema.Meta (class in	
(aries_cloudagent.indy.sdk.wallet_setup.IndyWalletConfig	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
attribute), 58	159	
key_management_mode	KeylistQuerySchema (class in	
(aries_cloudagent.multitenant.admin.routes.CreateWalletRequestSchema	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
attribute), 118	162	
key_management_mode	KeylistQuerySchema.Meta (class in	
(aries_cloudagent.wallet.models.wallet_record.WalletRecordSideEntry	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
attribute), 357	162	
KeyInfo (class in aries_cloudagent.wallet.base), 360	KeylistSchema (class in	
Keylist (class in aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
161	162	
Keylist.Meta (class in KeylistSchema (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	175	
161		
KeylistHandler (class in KeylistSchema.Meta (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.	162	
156		
KeylistKey (class in KeylistUpdate (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	163	
157		
KeylistKey.Meta (class in KeylistUpdate.Meta (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	163	
158		
KeylistKeySchema (class in KeylistUpdated (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	160	
158		
KeylistKeySchema.Meta (class in KeylistUpdated.Meta (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	160	
158		
KeylistQuery (class in KeylistUpdatedSchema (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	161	
162		
KeylistQuery.Meta (class in KeylistUpdatedSchema.Meta (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	161	
162		
KeylistQueryFilterRequestSchema (class in KeylistUpdateHandler (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.	156	
174		
KeylistQueryHandler (class in KeylistUpdateRequestSchema (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),	
aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.	175	
156		
KeylistQueryPaginate (class in KeylistUpdateResponse (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	164	
158		
KeylistQueryPaginate.Meta (class in KeylistUpdateResponse.Meta (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.	164	
159		
KeylistQueryPaginateQuerySchema (class in KeylistUpdateResponseHandler (class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers.	
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.	156	
174		

[KeylistUpdateResponseSchema](#) (class in [ledger_get_taa\(\)](#) (in module [aries_cloudagent.ledger.routes](#)), 72
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response](#)), 164
[LedgerConfigError](#), 67
[KeylistUpdateResponseSchema.Meta](#) (class in [LedgerError](#), 67
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist_update_response](#)), 164
[aries_cloudagent.config.argparse](#)), 12
[KeylistUpdateRule](#) (class in [LedgerModulesResultSchema](#) (class in [aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 71
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 159
[LedgerTransactionError](#), 67
[KeylistUpdateRule.Meta](#) (class in [limit](#) ([aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 159
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 159
[aries_cloudagent.config.argparse](#)), 12
[limit](#) ([aries_cloudagent.protocols.routing.v1_0.models.paginate.Paginate](#)), 293
[KeylistUpdateRuleSchema](#) (class in [attribute](#)), 293
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 160
[aries_cloudagent.config.argparse](#)), 12
[attribute](#)), 294
[KeylistUpdateRuleSchema.Meta](#) (class in [LINKED_DOMAINS](#) ([aries_cloudagent.ledger.endpoint_type.EndpointType](#)), 160
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 160
[LinkedDataKeySpec](#) (class in [aries_cloudagent.connections.models.diddoc](#)), 163
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 163
[LinkedDataKeySpec](#) (class in [aries_cloudagent.connections.models.diddoc](#)), 163
[aries_cloudagent.config.argparse](#)), 12
[links](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator](#)), 163
[aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 163
[links_](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator](#)), 163
[aries_cloudagent.config.argparse](#)), 12
[keys](#) ([aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.linked_data_key_spec](#)), 163
[aries_cloudagent.config.argparse](#)), 12
[links_](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator](#)), 163
[aries_cloudagent.config.argparse](#)), 12
[kid](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator](#)), 82
[aries_cloudagent.config.argparse](#)), 12
[list_issuer_registries\(\)](#)
[kid](#) ([aries_cloudagent.wallet.crypto.PackRecipientHeaderSchema](#) ([aries_cloudagent.revocation.indy.IndyRevocation](#)
[aries_cloudagent.config.argparse](#)), 361
[method](#)), 314
[list_mediation_requests\(\)](#) (in module [aries_cloudagent.protocols.coordinate_mediation.v1_0.routes](#)), 176
[label](#) ([aries_cloudagent.connections.models.connection_target.ConnectionTargetSchema](#)
[aries_cloudagent.config.argparse](#)), 37
[list_plain\(\)](#) ([aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_v1](#)), 201
[label](#) ([aries_cloudagent.multitenant.admin.routes.CreateWalletRequestSchema](#) ([aries_cloudagent.config.argparse](#)), 118
[aries_cloudagent.config.argparse](#)), 118
[list_plain\(\)](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential_v2](#)), 223
[label](#) ([aries_cloudagent.multitenant.admin.routes.UpdateWalletRequestSchema](#) ([aries_cloudagent.config.argparse](#)), 119
[aries_cloudagent.config.argparse](#)), 119
[list_plain\(\)](#) ([aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof_v1](#)), 264
[label](#) ([aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitationSchema](#)
[aries_cloudagent.config.argparse](#)), 140
[liveliness_handler\(\)](#)
[label](#) ([aries_cloudagent.protocols.connections.v1_0.messages.connection_invitation.ConnectionInvitationSchema](#)
[aries_cloudagent.config.argparse](#)), 141
[method](#)), 6
[label](#) ([aries_cloudagent.protocols.didexchange.v1_0.messages.load_argument.DDEXArgumentSchema](#) (in module [aries_cloudagent.config.argparse](#)), 182
[aries_cloudagent.config.argparse](#)), 182
[load_argument.DDEXArgumentSchema](#) (in module [aries_cloudagent.config.argparse](#)), 182
[label](#) ([aries_cloudagent.protocols.out_of_band.v1_0.messages.load_argument.DDEXArgumentSchema](#) (in module [aries_cloudagent.config.argparse](#)), 251
[aries_cloudagent.config.argparse](#)), 251
[load_argument.DDEXArgumentSchema](#) (in module [aries_cloudagent.config.argparse](#)), 251
[lastmod_time](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema](#) module
[aries_cloudagent.config.argparse](#)), 83
[aries_cloudagent.config.argparse](#)), 83
[LE](#) ([aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate](#) ([aries_cloudagent.messaging.decorators.base.BaseDecorator](#)
[aries_cloudagent.config.argparse](#)), 275
[method](#)), 83
[ledger_accept_taa\(\)](#) (in module [load_module\(\)](#) ([aries_cloudagent.utils.classloader.ClassLoader](#)
[aries_cloudagent.ledger.routes](#)), 72
[class method](#)), 348
[ledger_config\(\)](#) (in module [load_plugins\(\)](#) ([aries_cloudagent.config.default_context.DefaultContext](#)
[aries_cloudagent.config.ledger](#)), 19
[method](#)), 16

[load_postgres_plugin\(\)](#) (in module [aries_cloudagent.commands.provision](#)), 57
[load_protocol_version\(\)](#) (in module [aries_cloudagent.commands.start](#)), 11
[load_protocols\(\)](#) ([aries_cloudagent.core.plugin_registry.PluginRegistry](#) attribute), 45
[load_protocols\(\)](#) ([aries_cloudagent.core.plugin_registry.PluginRegistry](#) method), 45
[load_resource\(\)](#) (in module [aries_cloudagent.config.logging](#)), 19
[load_subclass_of\(\)](#) ([aries_cloudagent.utils.classloader.ClassLoader](#) class method), 349
[locale](#) ([aries_cloudagent.messaging.decorators.localization_decorator.LocalizationDecoratorSchema](#) attribute), 85
[locale](#) ([aries_cloudagent.protocols.problem_report.v1_0.message.ProblemReportSchema](#) attribute), 286
[localizable](#) ([aries_cloudagent.messaging.decorators.localization_decorator.LocalizationDecoratorSchema](#) attribute), 85
[LocalizationDecorator](#) (class in [aries_cloudagent.messaging.decorators.localization_decorator](#)), 84
[LocalizationDecorator](#) ([aries_cloudagent.messaging.decorators.localization_decorator](#) method), 344
[LocalizationDecorator.Meta](#) (class in [aries_cloudagent.messaging.decorators.localization_decorator](#)), 85
[LocalizationDecoratorSchema](#) (class in [aries_cloudagent.messaging.decorators.localization_decorator](#)), 85
[LocalizationDecoratorSchema](#) ([aries_cloudagent.messaging.decorators.localization_decorator](#) method), 352
[LocalizationDecoratorSchema.Meta](#) (class in [aries_cloudagent.messaging.decorators.localization_decorator](#)), 85
[log\(\)](#) ([aries_cloudagent.utils.stats.Collector](#) method), 352
[log\(\)](#) ([aries_cloudagent.utils.stats.Stats](#) method), 353
[log_state\(\)](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) class method), 101
[LOG_STATE_FLAG](#) ([aries_cloudagent.connections.models.conn_record.ConnRecord](#) attribute), 32
[LOG_STATE_FLAG](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) attribute), 101
[LOG_STATE_FLAG](#) ([aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord](#) attribute), 309
[log_task\(\)](#) ([aries_cloudagent.core.dispatcher.Dispatcher](#) method), 41
[LoggingConfigurator](#) (class in [aries_cloudagent.config.logging](#)), 19
[LoggingGroup](#) (class in [aries_cloudagent.config.argparse](#)), 12
[LT](#) ([aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Predicate](#) attribute), 275

M
[main\(\)](#) (in module [aries_cloudagent.commands.help](#)), 10

```

attribute), 150
mediation_id(aries_cloudagent.protocols.connections.v1_0.routes.CreateInvitationRequestSchema)
attribute), 152
mediation_id(aries_cloudagent.protocols.connections.v1_0.routes.ReceiveInvitationQueryStringSchema)
attribute), 152
mediation_id(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord)
attribute), 169
MediationGrantSchema (class in
mediation_id(aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.create_invitation_create_invitation_request_schema)
attribute), 170
mediation_id(aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.receive_invitation_receive_invitation_query_string_schema)
attribute), 175
mediation_id(aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.match_info_match_info_schema)
attribute), 185
MediationGroup (class in
mediation_id(aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXAcceptRequestQueryStringSchema)
attribute), 186
MediationIdMatchInfoSchema (class in
mediation_id(aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXRelayRequestQueryStringSchema)
attribute), 186
MediationListSchema (class in
mediation_id(aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationCreateRequestSchema)
attribute), 260
mediation_id(aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationReceiveQueryStringSchema)
attribute), 261
MediationRecord (class in
mediation_record_if_id() (in module aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),
aries_cloudagent.connections.util), 38
mediation_request_deny() (in module MediationManager (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),
aries_cloudagent.protocols.coordinate_mediation.v1_0.manager)
176
mediation_request_grant() (in module MediationManagerError, 174
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),
NotGrantedError, 174
176
MediationRecord (class in
mediation_sort_key() (in module aries_cloudagent.protocols.coordinate_mediation.v1_0.models.m
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),
176
MediationRecord.Meta (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.models.m
MediationAlreadyExists, 170
MediationCreateRequestSchema (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),
175
RecordSchema (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.models.m
MediationDeny (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messagesmediate_deny).Meta (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.models.m
164
MediationDeny.Meta (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messagesmediate_deny), (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages
165
MediationDenyHandler (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.handlersmediate_handler)(class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages
157
MediationDenySchema (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messagesmediate_deny_handler) (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.handlers
165
MediationDenySchema.Meta (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messagesmediate_deny_handler).Meta (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages
165
MediationGrant (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messagesmediate_grant).Meta (class in
aries_cloudagent.protocols.coordinate_mediation.v1_0.messages

```

167
mediator_terms (aries_cloudagent.protocols.coordinate_mediation_v1_0.messages.mediate_request.MediationRequestSchema attribute), 165
mediator_terms (aries_cloudagent.protocols.coordinate_mediation_v1_0.messages.mediate_request.MediationRequestSchema attribute), 167
mediator_terms (aries_cloudagent.protocols.coordinate_mediation_v1_0.messages.mediate_request.MediationRequestSchema attribute), 170
mediator_terms (aries_cloudagent.protocols.coordinate_mediation_v1_0.messages.mediate_request.MediationRequestSchema attribute), 174
mediator_terms (aries_cloudagent.protocols.coordinate_mediation_v1_0.messages.mediate_request.MediationRequestSchema attribute), 175
mediator_terms (aries_cloudagent.protocols.coordinate_mediation_v1_0.messages.mediate_request.MediationRequestSchema attribute), 175
menu (aries_cloudagent.protocols.actionmenu.v1_0.routes.AdminMenuSchema handler (class in aries_cloudagent.protocols.actionmenu.v1_0.handlers.menu_request.MenuRequestSchema)), 133
Menu (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 124
Menu (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 124
Menu.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 126
MenuForm (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 126
MenuForm (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 126
MenuForm.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 127
MenuFormParam (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 127
MenuFormParam (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 127
MenuFormParam.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 128
MenuFormParamSchema (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 128
MenuFormParamSchema (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 129
MenuFormParamSchema.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 129
MenuFormSchema (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 129
MenuFormSchema (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 129
MenuFormSchema.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 128
MenuHandler (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 123
MenuJsonSchema (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 132
MenuOption (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 129
MenuOption.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_request.MenuRequestSchema), 129

<code>attribute()</code> , 47	<code>attribute()</code> , 36
<code>MessageParseError</code> , 110	<code>model_class (aries_cloudagent.messaging.ack.message.AckSchema.MetaData)</code> , 74
<code>MessagePrepareError</code> , 110	<code>model_class (aries_cloudagent.messaging.agent_message.AgentMessage)</code> , 109
<code>MessageReceipt</code> (class in <code>aries_cloudagent.transport.inbound.receipt</code>), 333	<code>model_class (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema)</code> , 80
<code>metadata (aries_cloudagent.protocols.connections.v1_0.routes.ConnectionMetadata)</code> , 151	<code>model_class (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema)</code> , 80
<code>metadata (aries_cloudagent.protocols.connections.v1_0.routes.CreateConnectionRequestSchema)</code> , 152	<code>model_class (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema)</code> , 82
<code>metadata (aries_cloudagent.protocols.out_of_band.v1_0.routes.InvitationRequestSchema)</code> , 260	<code>model_class (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema)</code> , 82
<code>metadata (aries_cloudagent.wallet.base.DIDInfo attribute)</code> , 360	<code>model_class (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema)</code> , 83
<code>metadata (aries_cloudagent.wallet.base.KeyInfo attribute)</code> , 360	<code>model_class (aries_cloudagent.messaging.decorators.localization_decorator.LocalizationDecoratorSchema)</code> , 85
<code>metadata (aries_cloudagent.wallet.did_posture.DIDPosture attribute)</code> , 364	<code>model_class (aries_cloudagent.messaging.decorators.please_ack_decorator.PleaseAckDecoratorSchema)</code> , 86
<code>metadata_delete()</code> (<code>aries_cloudagent.connections.models.conn_record.ConnRecord</code> method), 33	<code>model_class (aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecoratorSchema)</code> , 89
<code>metadata_get()</code> (<code>aries_cloudagent.connections.models.conn_record.ConnRecord</code> method), 33	<code>model_class (aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecoratorSchema)</code> , 90
<code>metadata_get_all()</code> (<code>aries_cloudagent.connections.models.conn_record.ConnRecord</code> method), 34	<code>model_class (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecoratorSchema)</code> , 95
<code>metadata_set()</code> (<code>aries_cloudagent.connections.models.conn_record.ConnRecord</code> method), 34	<code>model_class (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecoratorSchema)</code> , 99
<code>mime_type (aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorSchema attribute)</code> , 83	<code>model_class (aries_cloudagent.messaging.decorators.transport_decorator.TransportDecoratorSchema)</code> , 100
<code>mime_type (aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_preview.CredAttrSpecSchema attribute)</code> , 202	<code>model_class (aries_cloudagent.messaging.models.base.BaseModelSchema)</code> , 102
<code>mime_type (aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential_preview.V20CredAttrSpecSchema attribute)</code> , 223	<code>model_class (aries_cloudagent.messaging.models.base_record.BaseRecordSchema)</code> , 102
<code>mime_type (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_preview.PresAttrSpecSchema attribute)</code> , 265	<code>model_class (aries_cloudagent.messaging.models.base_record.BaseRecordSchema)</code> , 102
<code>mime_types()</code> (<code>aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_preview.CredentialPreview</code> method), 202	<code>model_class (aries_cloudagent.messaging.models.openapi.OpenAPISchema)</code> , 102
<code>mime_types()</code> (<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_credential_preview.V20CredPreview</code> method), 224	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 126
<code>MIN_SIZE (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry attribute)</code> , 312	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 127
<code>MockResponder</code> (class in <code>aries_cloudagent.messaging.responder</code>), 112	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 128
<code>MODE_MANAGED (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute)</code> , 356	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 129
<code>MODE_UNMANAGED (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute)</code> , 356	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 130
<code>Model (aries_cloudagent.messaging.models.base.BaseModelSchema attribute)</code> , 99	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 130
<code>model_class (aries_cloudagent.connections.models.conn_record.ConnRecord)</code> , 35	<code>model_class (aries_cloudagent.protocols.actionmenu.v1_0.messages.actionmenu_preview.ActionMenuPreview)</code> , 136
<code>model_class (aries_cloudagent.connections.models.connection_target.CommunicationTargetSchema)</code> , 136	

[illegible]

attribute), 268	msg (aries_cloudagent.protocols.routing.v1_0.messages.forward.ForwardS
model_class (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack.PresentationAckSchema.Meta	msg_catalog (aries_cloudagent.protocols.problem_report.v1_0.messages
attribute), 269	msg_id (aries_cloudagent.messaging.decorators.trace_decorator.TraceRe
model_class (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposalSchema.Meta	msg_id (aries_cloudagent.messaging.decorators.trace_decorator.TraceRe
attribute), 270	msg_id (aries_cloudagent.messaging.decorators.trace_decorator.TraceRe
model_class (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request.PresentationRequestSchema.Meta	multi_use (aries_cloudagent.protocols.connections.v1_0.routes.CreateIn
attribute), 271	multi_use (aries_cloudagent.protocols.connections.v1_0.routes.CreateIn
model_class (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_exchange.V10PresentationExchangeSchema.Meta	multi_use (aries_cloudagent.protocols.out_of_band.v1_0.routes.Invitati
attribute), 273	MultiUse (aries_cloudagent.protocols.out_of_band.v1_0.routes.Invitati
model_class (aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReportSchema.Meta	MultitenantGroup (class in
attribute), 286	MultitenantManager (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.forward.ForwardSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 289	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_query_request.RoutingQueryRequestSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 290	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_query_response.RoutingQueryResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 290	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_request.RoutingUpdateRequestSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 291	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 292	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 293	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 294	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 294	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 296	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 297	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.routing.v1_0.messages.routing_update_response.RoutingUpdateResponseSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 298	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.trustping.v1_0.messages.ping.PingSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 302	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.protocols.trustping.v1_0.messages.ping.PingSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 303	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.revocation.models.indy.NonRevocationIntervalSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 305	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.revocation.models.issue_credential_report.IssueCredentialReportSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 307	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.revocation.models.issue_credential_report.IssueCredentialReportSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 310	MultitenantModuleResponseSchema (class in
model_class (aries_cloudagent.wallet.models.wallet_record.WalletRecordSchema.Meta	MultitenantModuleResponseSchema (class in
attribute), 357	MultitenantModuleResponseSchema (class in
models (aries_cloudagent.messaging.decorators.base.BaseDecoratorSent (aries_cloudagent.protocols.didexchange.v1_0.routes.DID	MultitenantModuleResponseSchema (class in
attribute), 84	MultitenantModuleResponseSchema (class in
module (aries_cloudagent.transport.inbound.base.InboundTransportConfiguration	MultitenantModuleResponseSchema (class in
attribute), 328	MultitenantModuleResponseSchema (class in
ModuleLoadError, 349	MultitenantModuleResponseSchema (class in
moniker (aries_cloudagent.wallet.did_posture.DIDPosture	MultitenantModuleResponseSchema (class in
attribute), 364	MultitenantModuleResponseSchema (class in
moniker (aries_cloudagent.wallet.did_posture.DIDPostureSpec	MultitenantModuleResponseSchema (class in
attribute), 365	MultitenantModuleResponseSchema (class in

N

- naked_to_did_key() (in module aries_cloudagent.wallet.util), 374
- name (aries_cloudagent.config.injection_context.Scope attribute), 18
- name (aries_cloudagent.core.profile.Profile attribute), 46
- name (aries_cloudagent.indy.sdk.profile.IndySdkProfile attribute), 55
- name (aries_cloudagent.indy.sdk.wallet_setup.IndyOpenWallet attribute), 57
- name (aries_cloudagent.messaging.schemas.routes.SchemaSchema attribute), 105
- name (aries_cloudagent.protocols.actionmenu.v1_0.messages.PerformPerformSchema attribute), 127
- name (aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form_param.MenuFormParamSchema attribute), 129
- name (aries_cloudagent.protocols.actionmenu.v1_0.models.menu_option_menu_option_schema.MenuOptionSchema attribute), 130
- name (aries_cloudagent.protocols.actionmenu.v1_0.routes.PerformPerformReqAttribSpecSchema attribute), 133
- name (aries_cloudagent.protocols.issue_credential.v1_0.messages.inner_cred_attach_preview.CredAttachPreviewSchema attribute), 202
- name (aries_cloudagent.protocols.issue_credential.v2_0.messages.inner_cred_attach_preview.CredAttachPreviewSchema attribute), 223
- name (aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.NISPProtocolSpec (class in aries_cloudagent.messaging.valid), 117
- name (aries_cloudagent.protocols.present_proof.v1_0.messages.inner_presentation_preview.PresAttrSpecSchema attribute), 265
- name (aries_cloudagent.protocols.present_proof.v1_0.messages.inner_presentation_preview.PresPresSpecSchema attribute), 266
- name (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqAttribSpecSchema attribute), 278
- name (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqPredSpecSchema attribute), 279
- name (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqSalienceSpecSchema attribute), 279
- names (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqAttribSpecSchema attribute), 278
- NaturalNumber (class in aries_cloudagent.messaging.valid), 117
- NETWORK_MONITOR (aries_cloudagent.ledger.base.Role attribute), 65
- NEW (aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix class attribute), 304
- next() (aries_cloudagent.utils.repeat.RepeatAttempt method), 351
- next_interval(aries_cloudagent.utils.repeat.RepeatAttempt attribute), 352
- next_interval() (aries_cloudagent.utils.repeat.RepeatSequence attribute), 352
- no (aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Relation attribute), 125
- non_revoc_intervals() (aries_cloudagent.indy.sdk.verifier.IndySdkVerifier method), 56
- non_revoked(aries_cloudagent.protocols.present_proof.v1_0.routes.Ina attribute), 278
- non_revoked(aries_cloudagent.protocols.present_proof.v1_0.routes.Ina attribute), 279
- non_revoked(aries_cloudagent.protocols.present_proof.v1_0.routes.Ina attribute), 279
- nonce (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRe attribute), 279
- NonRevocationInterval (class in aries_cloudagent.revocation.models.indy), 305
- NonRevocationIntervalMeta (class in aries_cloudagent.revocation.models.indy), 305
- NonRevocationIntervalSchema (class in aries_cloudagent.revocation.models.indy), 305
- NonRevocationIntervalSchema.Meta (class in aries_cloudagent.revocation.models.indy), 305
- notify_fatal_error() (aries_cloudagent.admin.server_admin_server method), 6
- now() (aries_cloudagent.utils.stats.Timer class attribute), 353
- NumericStrWholeNISPProtocolSpec (class in aries_cloudagent.messaging.valid), 117
- NumericStrWholePresentationPreviewPresAttrSpecSchema (class in aries_cloudagent.messaging.valid), 117
- nym_to_did() (aries_cloudagent.ledger.base.BaseLedger method), 65
- nym_to_id() (aries_cloudagent.ledger.indy.IndySdkLedger method), 69
- offer_attach(aries_cloudagent.protocols.issue_credential.v2_0.messages.cre attribute), 229
- offset(aries_cloudagent.protocols.coordinate_mediation.v1_0.messages attribute), 159
- offset(aries_cloudagent.protocols.routing.v1_0.models.paginate.Pagina attribute), 293
- ok_did() (in module aries_cloudagent.connections.models.diddoc.util), 30
- OLD (aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix attribute), 304

P

paginated (aries_cloudagent.protocols.routing.v1_0.messages.route_query_response.RouteQueryResponseSchema attribute), 290
 PATTERN (aries_cloudagent.messaging.valid.Endpoint attribute), 115
 Paginated (class in aries_cloudagent.protocols.routing.v1_0.models.paginated), 293
 PATTERN (aries_cloudagent.messaging.valid.IndyCredDefId attribute), 115
 Paginated.Meta (class in PATTERN (aries_cloudagent.messaging.valid.IndyCredRevId aries_cloudagent.protocols.routing.v1_0.models.paginated)attribute), 115
 PATTERN (aries_cloudagent.messaging.valid.IndyDID attribute), 115
 PaginatedSchema (class in PATTERN (aries_cloudagent.messaging.valid.IndyExtraWQL aries_cloudagent.protocols.routing.v1_0.models.paginated)attribute), 115
 PATTERN (aries_cloudagent.messaging.valid.IndyISO8601DateTime attribute), 115
 PaginatedSchema.Meta (class in PATTERN (aries_cloudagent.messaging.valid.IndyRawPublicKey aries_cloudagent.protocols.routing.v1_0.models.paginated)attribute), 115
 PATTERN (aries_cloudagent.messaging.valid.IndyRevRegId attribute), 116
 PaginateSchema (class in PATTERN (aries_cloudagent.messaging.valid.IndySchemaId aries_cloudagent.protocols.routing.v1_0.models.paginate), attribute), 116
 PATTERN (aries_cloudagent.messaging.valid.IndyVersion attribute), 116
 pagination (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.keylist.KeylistSchema attribute), 162
 PATTERN (aries_cloudagent.messaging.valid.IndyWQL attribute), 115
 params (aries_cloudagent.protocols.actionmenu.v1_0.messages.perform_menu_form.Schema attribute), 127
 PATTERN (aries_cloudagent.messaging.valid.JSONWebToken attribute), 115
 params (aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form.MenuFormSchema attribute), 128
 PATTERN (aries_cloudagent.messaging.valid.JWSHeaderKid attribute), 115
 params (aries_cloudagent.protocols.actionmenu.v1_0.routes.PerformMenuRouteSchema attribute), 133
 PATTERN (aries_cloudagent.messaging.valid.NumericStrNatural attribute), 117
 parent (aries_cloudagent.cache.base.CacheKeyLock attribute), 9
 PATTERN (aries_cloudagent.messaging.valid.NumericStrWhole attribute), 117
 parent_thread_id (aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_exchange.V10CredentialExchangeSchema attribute), 212
 PATTERN (aries_cloudagent.messaging.valid.SHA256Hash attribute), 115
 parent_thread_id (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred_ex_record.V20CredExRecordSchema attribute), 238
 PATTERN (aries_cloudagent.messaging.valid.UUIDFour attribute), 115
 parse_inbound() (aries_cloudagent.transport.inbound.session.InboundSession method), 336
 pending_pub (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord attribute), 110
 parse_message() (aries_cloudagent.transport.pack_format.PackFormat method), 345
 PendingTask (class in aries_cloudagent.utils.task_queue), 353
 parse_message() (aries_cloudagent.transport.wire_format.BaseWireFormat method), 347
 Perform (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.perform.Perform), 126
 parse_message() (aries_cloudagent.transport.wire_format.JsonWireFormat method), 348
 Perform.Meta (class in aries_cloudagent.protocols.actionmenu.v1_0.messages.perform), 126
 parse_type_string() (aries_cloudagent.core.protocol_registry.ProtocolRegistry method), 47
 perform_encode() (aries_cloudagent.transport.outbound.manager.OutboundManager method), 340
 PATTERN (aries_cloudagent.messaging.valid.Base58SHA256Hash attribute), 114
 perform_menu_action() (aries_cloudagent.protocols.actionmenu.v1_0.base_service.BaseService method), 131
 PATTERN (aries_cloudagent.messaging.valid.Base64 attribute), 114
 perform_menu_action() (aries_cloudagent.protocols.actionmenu.v1_0.driver_service.DriverService method), 132
 PATTERN (aries_cloudagent.messaging.valid.Base64URL attribute), 114
 PerformHandler (class in aries_cloudagent.protocols.actionmenu.v1_0.handlers.perform_handler.PerformHandler), 132
 PATTERN (aries_cloudagent.messaging.valid.Base64URLNoPad attribute), 114
 PATTERN (aries_cloudagent.messaging.valid.DIDKey attribute), 115

124	PleaseAckDecoratorSchema.Meta	(class in	aries_cloudagent.messaging.decorators.please_ack_decorator),	86
133	PerformRequestSchema	(class in	aries_cloudagent.protocols.actionmenu.v1_0.routes),	86
133	PerformSchema	(class in	aries_cloudagent.protocols.actionmenu.v1_0.messages.perform),	45
126	PerformSchema.Meta	(class in	aries_cloudagent.core.plugin_registry.PluginRegistry),	45
126	PerformSchema.Meta	(class in	aries_cloudagent.protocols.actionmenu.v1_0.messages.perform),	45
126	plugins_handler()			
301	Ping	(class in	aries_cloudagent.protocols.trustping.v1_0.messages.ping),	6
301	Ping.Meta	(class in	aries_cloudagent.ledger.indy.IndySdkLedger),	69
302	PingHandler	(class in	aries_cloudagent.ledger.indy.IndySdkLedger),	69
301	PingHandler	(class in	aries_cloudagent.messaging.agent_message.AgentMessageSchema),	109
303	PingRequestResponseSchema	(class in	aries_cloudagent.transport.inbound.base.InboundTransportConfiguration),	328
303	PingRequestSchema	(class in	aries_cloudagent.protocols.out_of_band.v1_0.messages.inbound_ping),	251
303	PingResponse	(class in	aries_cloudagent.protocols.out_of_band.v1_0.messages.out_of_band_ping_response),	255
302	PingResponse	(class in	aries_cloudagent.core.plugin_registry.PluginRegistry),	45
302	PingResponse.Meta	(class in	aries_cloudagent.ledger.routes),	50
301	PingResponseHandler	(class in	aries_cloudagent.ledger.routes),	73
301	PingResponseSchema	(class in	aries_cloudagent.messaging.credential_definitions.routes),	75
302	PingResponseSchema	(class in	aries_cloudagent.messaging.schemas.routes),	95
302	PingResponseSchema.Meta	(class in	aries_cloudagent.messaging.schemas.routes),	95
302	PingSchema	(class in	aries_cloudagent.multitenant.admin.routes),	119
302	PingSchema.Meta	(class in	aries_cloudagent.protocols.actionmenu.v1_0.routes),	134
86	PleaseAckDecorator	(class in	aries_cloudagent.protocols.basicmessage.v1_0.routes),	86
86	PleaseAckDecorator.Meta	(class in	aries_cloudagent.protocols.connections.v1_0.routes),	86
86	PleaseAckDecoratorSchema	(class in	aries_cloudagent.protocols.coordinate_mediation.v1_0.routes),	86

```

aries_cloudagent.protocols.didexchange.v1_0.routes).prefix_tag_filter()
187
post_process_routes() (in module aries_cloudagent.messaging.models.base_record.BaseRecord
    aries_cloudagent.protocols.discovery.v1_0.routes).prepare_disclosed()
192
post_process_routes() (in module aries_cloudagent.core.protocol_registry.ProtocolRegistry
    aries_cloudagent.protocols.introduction.v0_1.routes).prepare_keylist_query()
198
post_process_routes() (in module aries_cloudagent.protocols.coordinate_mediation.v1_0.manager
    aries_cloudagent.protocols.issue_credential.v1_0.routes).prepare_pack_recipient_keys() (in module
220
post_process_routes() (in module aries_cloudagent.wallet.crypto), 363
    aries_cloudagent.protocols.issue_credential.v2_0.routes).prepare_request()
247
post_process_routes() (in module aries_cloudagent.protocols.coordinate_mediation.v1_0.manager
    aries_cloudagent.protocols.out_of_band.v1_0.routes).prepare_send() (aries_cloudagent.protocols.issue_credential.v1_0.ma
261
post_process_routes() (in module aries_cloudagent.protocols.issue_credential.v2_0.ma
    aries_cloudagent.protocols.present_proof.v1_0.routes).pres_ex_id(aries_cloudagent.protocols.present_proof.v1_0.routes.Pres
281
post_process_routes() (in module PresAttrSpec (class in
    aries_cloudagent.protocols.trustping.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
303
post_process_routes() (in module PresAttrSpec.Meta (class in
    aries_cloudagent.revocation.routes), 319
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
264
post_process_routes() (in module PresAttrSpec.Posture (class in
    aries_cloudagent.wallet.routes), 372
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
264
post_save() (aries_cloudagent.connections.models.conn_record.ConnectionRecord
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
264
post_save() (aries_cloudagent.messaging.models.base_record.BaseRecordSchema (class in
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
265
POSTED (aries_cloudagent.wallet.did_posture.DIDPosture
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
265
posted (aries_cloudagent.wallet.did_posture.DIDPostureSpec
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
265
posture (aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.PresentationPreview
    aries_cloudagent.protocols.present_proof.v1_0.messages.inner.pr
264
posture (aries_cloudagent.wallet.routes.DIDListQueryStringSchema (class in
    aries_cloudagent.protocols.present_proof.v1_0.messages.presentat
267
posture (aries_cloudagent.wallet.routes.DIDSchema
    aries_cloudagent.protocols.present_proof.v1_0.messages.presentat
268
pre_load() (aries_cloudagent.protocols.out_of_band.v1_0.messages.v1_0.messages.presentat
    aries_cloudagent.protocols.out_of_band.v1_0.messages.presentat
251
pre_verify() (aries_cloudagent.indy.sdk.verifier.IndySdkVerifier
    aries_cloudagent.protocols.present_proof.v1_0.routes)
56
predicate (aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.PresPredSpecSchema
    aries_cloudagent.protocols.present_proof.v1_0.routes)
266
Predicate (class in aries_cloudagent.protocols.present_proof.v1_0.routes)
    aries_cloudagent.protocols.present_proof.v1_0.util.predicate
274
predicates (aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.PresPredSpecSchema
    aries_cloudagent.protocols.present_proof.v1_0.messages.presentat
267
prefix (aries_cloudagent.messaging.decorators.base_decorator.BaseDecorator
    aries_cloudagent.protocols.present_proof.v1_0.messages.presentat
84

```


attribute), 274
presentation_exchange_list() (in module PresentationAckSchema (class in
aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 282
presentation_exchange_problem_report() PresentationAckSchema.Meta (class in
(in module aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 282
presentation_exchange_remove() (in module PresentationHandler (class in
aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack), 282
presentation_exchange_retrieve() (in module PresentationManager (class in
ule aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.manager), 282
presentation_exchange_send_bound_request() PresentationManagerError, 277
(in module aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_ack), 282
presentation_exchange_send_free_request() 266
(in module aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_ack), 283
presentation_exchange_send_presentation() 266
(in module aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_ack), 283
presentation_exchange_send_proposal() 267
(in module aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_ack), 283
presentation_exchange_verify_presentation() 267
(in module aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_ack), 283
presentation_proposal 269
(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 270
presentation_proposal 269
(aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack), 281
presentation_proposal_dict 263
(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 274
presentation_referents 269
(aries_cloudagent.protocols.present_proof.v1_0.routes), aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 278
presentation_request 269
(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 274
presentation_request_dict 270
(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 274
PresentationAck (class in 270
aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 268
PresentationAck.Meta (class in 263
aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_ack), 268
PresentationAckHandler (class in 270
aries_cloudagent.protocols.present_proof.v1_0.handlers.presentation_ack), 269

`aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request),`
270 `ProblemReport.Meta` (class in
presentations_attach `aries_cloudagent.protocols.connections.v1_0.messages.problem_`
`(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation.PresentationSchema`
attribute), 268 `ProblemReport.Meta` (class in
PresentationSchema (class in `aries_cloudagent.protocols.didexchange.v1_0.messages.problem_`
`aries_cloudagent.protocols.present_proof.v1_0.messages.presentation),`
268 `ProblemReport.Meta` (class in
PresentationSchema.Meta (class in `aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_`
`aries_cloudagent.protocols.present_proof.v1_0.messages.presentation),`
268 `ProblemReport.Meta` (class in
PresentProofModuleResponseSchema (class in `aries_cloudagent.protocols.problem_report.v1_0.message),`
`aries_cloudagent.protocols.present_proof.v1_0.routes),` 286
280 `ProblemReportHandler` (class in
PresExIdMatchInfoSchema (class in `aries_cloudagent.protocols.problem_report.v1_0.handler),`
`aries_cloudagent.protocols.present_proof.v1_0.routes),` 284
280 `ProblemReportReason` (class in
PresPredSpec (class in `aries_cloudagent.protocols.connections.v1_0.messages.problem_`
`aries_cloudagent.protocols.present_proof.v1_0.messages.inbound.presentation_preview),`
265 `ProblemReportReason` (class in
PresPredSpec.Meta (class in `aries_cloudagent.protocols.didexchange.v1_0.messages.problem_`
`aries_cloudagent.protocols.present_proof.v1_0.messages.inbound.presentation_preview),`
265 `ProblemReportReason` (class in
PresPredSpecSchema (class in `aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_`
`aries_cloudagent.protocols.present_proof.v1_0.messages.inbound.presentation_preview),`
265 `ProblemReportSchema` (class in
PresPredSpecSchema.Meta (class in `aries_cloudagent.protocols.connections.v1_0.messages.problem_`
`aries_cloudagent.protocols.present_proof.v1_0.messages.inbound.presentation_preview),`
265 `ProblemReportSchema` (class in
print_banner() (`aries_cloudagent.config.logging.LoggingConfiguration` or `aries_cloudagent.protocols.didexchange.v1_0.messages.problem_`
class method), 19 181
priority (`aries_cloudagent.connections.models.diddoc.Service` or `ProblemReportSchema` (class in
attribute), 25 `aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_`
priority (`aries_cloudagent.connections.models.diddoc.service.Service` or `ProblemReportSchema` (class in
attribute), 29 `ProblemReportSchema` (class in
problem_code (`aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportSchema` in
attribute), 143 286
problem_code (`aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportSchema` in
attribute), 181 `aries_cloudagent.protocols.connections.v1_0.messages.problem_`
problem_code (`aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportSchema`
attribute), 252 `ProblemReportSchema.Meta` (class in
problem_items (`aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReportSchema` in
attribute), 286 181
ProblemReport (class in `ProblemReportSchema.Meta` (class in
`aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportSchema` in
142 252
ProblemReport (class in `ProblemReportSchema.Meta` (class in
`aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportSchema` in
180 286
ProblemReport (class in `process_inbound()`
`aries_cloudagent.protocols.out_of_band.v1_0.messages.problem_report.ProblemReportSchema` in
251 `method`), 336
ProblemReport (class in `process_queued()` (`aries_cloudagent.transport.outbound.manager.OutboundManager` or
`aries_cloudagent.protocols.problem_report.v1_0.message`), method), 340

[process_undelivered\(\)](#) (method), 20
[process_undelivered\(\)](#) (aries_cloudagent.transport.inbound.manager.InboundTransportManager method), 331
[profile](#) (aries_cloudagent.admin.request_context.AdminRequestContext attribute), 4
[profile](#) (aries_cloudagent.config.provider.InstanceProvider method), 20
[profile](#) (aries_cloudagent.config.provider.StatsProvider method), 20
[profile](#) (aries_cloudagent.core.profile.ProfileSession provide()) (aries_cloudagent.config.provider.StatsProvider method), 20
[PROFILE](#) (aries_cloudagent.ledger.endpoint_type.EndpointType provide()) (aries_cloudagent.core.profile.ProfileManagerProvider method), 46
[profile](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 70
[profile](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 111
[profile](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 214
[profile](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 240
[Profile](#) (class in aries_cloudagent.core.profile), 45
[ProfileDuplicateError](#), 42
[ProfileError](#), 43
[ProfileManager](#) (class in aries_cloudagent.core.profile), 46
[ProfileManagerProvider](#) (class in aries_cloudagent.core.profile), 46
[ProfileNotFoundError](#), 43
[ProfileSession](#) (class in aries_cloudagent.core.profile), 46
[ProfileSessionInactiveError](#), 43
[proof_request](#) (aries_cloudagent.protocols.present_proof.present_proof attribute), 280
[protected](#) (aries_cloudagent.messaging.decorators.attach_decorator attribute), 80
[protected](#) (aries_cloudagent.messaging.decorators.attach_decorator attribute), 82
[protected](#) (aries_cloudagent.messaging.decorators.attach_decorator attribute), 361
[ProtocolDefinitionValidationError](#), 43
[ProtocolDescriptorSchema](#) (class in aries_cloudagent.connections.models.diddoc), 189
[ProtocolGroup](#) (class in aries_cloudagent.connections.models.diddoc), 13
[ProtocolMinorVersionNotSupported](#), 43
[ProtocolRegistry](#) (class in aries_cloudagent.connections.models.diddoc), 47
[protocols](#) (aries_cloudagent.core.protocol_registry.ProtocolRegistry attribute), 48
[protocols](#) (aries_cloudagent.protocols.discovery.v1_0.messages.disclosure attribute), 189
[protocols_matching_query\(\)](#) (aries_cloudagent.core.protocol_registry.ProtocolRegistry method), 48
[provide\(\)](#) (aries_cloudagent.config.base.BaseProvider method), 14
[provide\(\)](#) (aries_cloudagent.config.provider.CachedProvider method), 20
[provide\(\)](#) (aries_cloudagent.config.provider.InstanceProvider method), 20
[provide\(\)](#) (aries_cloudagent.config.provider.StatsProvider method), 20
[provide\(\)](#) (aries_cloudagent.core.profile.ProfileManagerProvider method), 46
[provide\(\)](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 70
[provide\(\)](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 111
[provide\(\)](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 214
[provide\(\)](#) (aries_cloudagent.ledger.indy.IndySdkLedgerPoolProvider method), 240
[provision\(\)](#) (aries_cloudagent.indy.sdk.profile.IndySdkProfileManager method), 55
[provision\(\)](#) (in module aries_cloudagent.commands.provision), 10
[ProvisionError](#), 10
[pthid](#) (aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecorator attribute), 89
[pthid](#) (aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecorator attribute), 89
[pubkey](#) (aries_cloudagent.connections.models.diddoc.DIDDoc attribute), 23
[public_key](#) (aries_cloudagent.connections.models.diddoc.DIDDoc attribute), 26
[public_key](#) (aries_cloudagent.connections.models.diddoc.DIDDoc attribute), 152
[public_key](#) (aries_cloudagent.connections.models.diddoc.DIDDoc attribute), 364
[public_key](#) (aries_cloudagent.connections.models.diddoc.DIDDoc attribute), 365
[PublicKey](#) (class in aries_cloudagent.connections.models.diddoc), 27
[PublicKeyType](#) (class in aries_cloudagent.connections.models.diddoc), 24
[PublicKeyType](#) (class in aries_cloudagent.connections.models.diddoc), 28
[publish_pending_revocations\(\)](#) (aries_cloudagent.revocation.manager.RevocationManager method), 315
[publish_revocations\(\)](#) (in module aries_cloudagent.revocation.routes), 319
[publish_revocations_schema](#) (class in aries_cloudagent.revocation.routes), 319

`aries_cloudagent.revocation.routes)`, 316
`purge()` (`aries_cloudagent.revocation.routes.ClearPendingRevocationsRequestSchema` (class in `aries_cloudagent.protocols.discovery.v1_0.messages.query`), attribute), 316
`put()` (`aries_cloudagent.utils.task_queue.TaskQueue` (class in `aries_cloudagent.protocols.discovery.v1_0.messages.query`), method), 354
`put_file()` (in module `aries_cloudagent.utils.http`), 350
`put_task()` (`aries_cloudagent.core.dispatcher.Dispatcher` (class in `aries_cloudagent.ledger.routes`), method), 41
`PutError`, 349
Q
`qualify()` (`aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix` (class in `aries_cloudagent.messaging.decorators.transport_decorator.TransportDecorator`), method), 304
`qualify()` (in module `aries_cloudagent.protocols.didcomm_prefix`), 304
`qualify_all()` (`aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix` (class in `aries_cloudagent.messaging.decorators.transport_decorator.TransportDecorator`), attribute), 304
`qualify_current()` (`aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix` (class in `aries_cloudagent.messaging.decorators.transport_decorator.TransportDecorator`), attribute), 304
`query()` (`aries_cloudagent.protocols.discovery.v1_0.messages.query.QuerySchema` (class in `aries_cloudagent.ledger.routes`), attribute), 191
`query()` (`aries_cloudagent.protocols.discovery.v1_0.routes.QueryFeaturesQuerySchema` (class in `aries_cloudagent.wallet.crypto`), attribute), 191
`Query` (class in `aries_cloudagent.protocols.discovery.v1_0.messages.query`), 190
`query()` (`aries_cloudagent.messaging.models.base_record.BaseRecord` (class in `aries_cloudagent.messaging.decorators.attach_decorator`), class method), 102
`Query.Meta` (class in `aries_cloudagent.protocols.discovery.v1_0.messages.query`), 191
`query_by_cred_def_id()` (`aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord` (class in `aries_cloudagent.ledger.base.BaseLedger`), class method), 309
`query_by_ids()` (`aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), class method), 306
`query_by_pending()` (`aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord` (class in `aries_cloudagent.ledger.base.BaseLedger`), class method), 310
`query_features()` (in module `aries_cloudagent.protocols.discovery.v1_0.routes`), 192
`QueryFeaturesQueryStringSchema` (class in `aries_cloudagent.ledger.routes`), 191
`QueryHandler` (class in `aries_cloudagent.protocols.discovery.v1_0.handlers.query_handler`), 188
`QueryResultSchema` (class in `aries_cloudagent.protocols.discovery.v1_0.routes`), 192
`QuerySchema` (class in `aries_cloudagent.protocols.discovery.v1_0.messages.query`), 191
`queue_message()` (`aries_cloudagent.core.dispatcher.Dispatcher` (class in `aries_cloudagent.ledger.routes`), method), 41
`queue_outbound()` (`aries_cloudagent.core.conductor.Conductor` (class in `aries_cloudagent.ledger.routes`), method), 40
R
`read_only()` (`aries_cloudagent.ledger.base.BaseLedger` (class in `aries_cloudagent.ledger.routes`), attribute), 7
`ready()` (`aries_cloudagent.admin.server.AdminStatusReadinessSchema` (class in `aries_cloudagent.ledger.routes`), attribute), 7
`ready_middleware()` (in module `aries_cloudagent.admin.server`), 8
`receive()` (`aries_cloudagent.transport.inbound.session.InboundSession` (class in `aries_cloudagent.ledger.routes`), method), 336
`receive_credential()` (`aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager` (class in `aries_cloudagent.ledger.routes`), method), 214
`receive_credential_ack()` (`aries_cloudagent.protocols.issue_credential.v2_0.manager.V200CredentialManager` (class in `aries_cloudagent.ledger.routes`), method), 240

(aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager
method), 214
receive_credential_ack()
(aries_cloudagent.protocols.issue_credential.v2_0.manager.V20CredManager
method), 240
receive_inbound()
(aries_cloudagent.transport.inbound.session.InboundSession
method), 336
receive_invitation()
(aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager
method), 148
receive_invitation()
(aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXMManager
method), 184
receive_invitation()
(aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager
method), 258
receive_offer()
(aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager
method), 214
receive_offer()
(aries_cloudagent.protocols.issue_credential.v2_0.manager.V20CredManager
method), 240
receive_presentation()
(aries_cloudagent.protocols.present_proof.v1_0.manager.PresentationManager
method), 277
receive_presentation_ack()
(aries_cloudagent.protocols.present_proof.v1_0.manager.PresentationManager
method), 277
receive_problem_report()
(aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager
method), 258
receive_proposal()
(aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager
method), 214
receive_proposal()
(aries_cloudagent.protocols.issue_credential.v2_0.manager.V20CredManager
method), 240
receive_proposal()
(aries_cloudagent.protocols.present_proof.v1_0.manager.PresentationManager
method), 277
receive_request()
(aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager
method), 149
receive_request()
(aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediationManager
method), 172
receive_request()
(aries_cloudagent.protocols.didexchange.v1_0.manager.DIDXMManager
method), 184
receive_request()
(aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager
method), 214
receive_request()
(aries_cloudagent.protocols.issue_credential.v2_0.manager.V20CredManager
method), 240

[recipient_terms \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecordSchema attribute\), 170](#)
[recipient_terms \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.admin_mediation_deny_schema.AdminMediationDenySchema attribute\), 174](#)
[recipient_terms \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_create_request_schema.MediationCreateRequestSchema attribute\), 175](#)
[recipient_terms \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_list_query_string_schema.MediationListQueryStringSchema attribute\), 175](#)
[recipient_verkey \(aries_cloudagent.transport.inbound.receipt.MessageReceipt attribute\), 334](#)
[RecipientKeysError, 344](#)
[recipients \(aries_cloudagent.wallet.crypto.PackRecipientsSchema attribute\), 361](#)
[record \(aries_cloudagent.protocols.connections.v1_0.models.route_record.RouteRecord attribute\), 151](#)
[record_id \(aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord attribute\), 295](#)
[record_id \(aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord attribute\), 296](#)
[record_id \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 307](#)
[record_id \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 307](#)
[record_id \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 310](#)
[record_id \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 310](#)
[RECORD_ID_NAME \(aries_cloudagent.connections.models.conn_record.ConnRecord attribute\), 32](#)
[RECORD_ID_NAME \(aries_cloudagent.messaging.models.base_record.BaseRecord attribute\), 101](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord attribute\), 169](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute\), 211](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.issue_credential.v2_0.models.issue_credential_v2_0_base_connection_manager.BaseConnectionManager attribute\), 237](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.issue_credential.v2_0.models.issue_credential_v2_0_cred_ex_record.DIF attribute\), 233](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.issue_credential.v2_0.models.issue_credential_v2_0_cred_ex_record.Indy attribute\), 234](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.Invitation attribute\), 256](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute\), 273](#)
[RECORD_ID_NAME \(aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord attribute\), 295](#)
[RECORD_ID_NAME \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 306](#)
[RECORD_ID_NAME \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 309](#)
[RECORD_ID_NAME \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute\), 34](#)
[RECORD_ID_NAME \(aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute\), 356](#)
[record_tags \(aries_cloudagent.messaging.models.base_record.BaseRecord attribute\), 102](#)

attribute), 169
 record_value (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange
 attribute), 211
 record_value (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord
 attribute), 237
 record_value (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecordDiff.v1_0.routes),
 attribute), 233
 record_value (aries_cloudagent.protocols.issue_credential.v2_0.models.detail.indy.V20CredExRecordIndy module
 attribute), 234
 record_value (aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.InvitationRecord
 attribute), 257
 record_value (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange
 attribute), 273
 record_value (aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord module
 attribute), 296
 record_value (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord
 attribute), 310
 record_value (aries_cloudagent.wallet.models.wallet_record.WalletRecord aries_cloudagent.protocols.didexchange.v1_0.routes),
 attribute), 356
 redirect_handler () register () (in module
 (aries_cloudagent.admin.server.AdminServer aries_cloudagent.protocols.discovery.v1_0.routes),
 method), 6
 ref_id (aries_cloudagent.protocols.connections.v1_0.routes.ConnectionRefIdMatchInfoSchema module
 attribute), 150
 ref_id (aries_cloudagent.protocols.didexchange.v1_0.routes.DIDXCNIdRefIdMatchInfoSchema
 attribute), 186
 referent (aries_cloudagent.holder.routes.CredBriefSchema aries_cloudagent.protocols.issue_credential.v1_0.routes),
 attribute), 49
 referent (aries_cloudagent.protocols.present_proof.v1_0.messages.inner.presentation_preview.PresAttrSpansSchema
 attribute), 265
 referent (aries_cloudagent.protocols.present_proof.v1_0.routes.CredentialsFetchQueryStringSchema
 attribute), 278
 referent (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyCredentialInfoSchema aries_cloudagent.protocols.out_of_band.v1_0.routes),
 attribute), 278
 reg_def (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry (in module
 attribute), 313
 reg_def_type (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry
 attribute), 313
 register () (aries_cloudagent.transport.inbound.manager.InboundTransportManager aries_cloudagent.protocols.trustping.v1_0.routes),
 method), 332
 register () (aries_cloudagent.transport.outbound.manager.OutboundTransportManager (in module
 method), 340
 register () (in module register () (in module
 aries_cloudagent.holder.routes), 50
 register () (in module register_admin_routes ()
 aries_cloudagent.ledger.routes), 73
 register () (in module method), 45
 aries_cloudagent.messaging.credential_definitions.requests, 76
 register_class () (aries_cloudagent.transport.outbound.manager.OutboundTransportManager, 340
 method), 340
 register () (in module register_controllers ()
 aries_cloudagent.messaging.jsonld.routes),
 97
 register () (in module register_ledger_nym () (in module
 aries_cloudagent.messaging.schemas.routes),
 aries_cloudagent.ledger.routes), 73

`register_message_types()` (method), 6
`(aries_cloudagent.core.protocol_registry.ProtocolRegistry, 48)` (class in `aries_cloudagent.multitenant.admin.routes`), 48
`register_nym()` (method), 65
`(aries_cloudagent.ledger.base.BaseLedger, 118)` (class in `RepeatAttempt` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), 351
`(aries_cloudagent.ledger.indy.IndySdkLedger, 69)` (method), 69
`RepeatSequence` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), 351
`register_package()` (method), 45
`(aries_cloudagent.core.plugin_registry.PluginRegistry, 45)` (method), 45
`register_plugin()` (method), 45
`(aries_cloudagent.core.plugin_registry.PluginRegistry, 45)` (method), 45
`register_transport()` (method), 332
`(aries_cloudagent.transport.inbound.manager.InboundTransportManager, 332)` (method), 332
`RegisterLedgerNymQueryStringSchema` (class in `aries_cloudagent.ledger.routes`), 71
`registry_id` (attribute), 313
`(aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry, 313)` (attribute), 313
`Relation` (class in `aries_cloudagent.protocols.present_proof.v1_0.util.predicate`), 275
`release()` (method), 8
`(aries_cloudagent.cache.base.BaseCache, 8)` (method), 8
`release()` (method), 9
`(aries_cloudagent.cache.base.CacheKeyLock, 9)` (method), 9
`remove()` (method), 46
`(aries_cloudagent.core.profile.Profile, 46)` (method), 46
`remove()` (method), 55
`(aries_cloudagent.indy.sdk.profile.IndySdkProfile, 55)` (method), 55
`remove_field()` (method), 84
`(aries_cloudagent.messaging.decorators.base.BaseDecorator, 84)` (method), 84
`remove_key()` (method), 173
`(aries_cloudagent.protocols.coordinate_mediation.v1_0.util.predicate, 173)` (method), 173
`remove_keys_for_did()` (method), 38
`(aries_cloudagent.connections.base_manager.BaseConnectionManager, 38)` (method), 38
`remove_message_for_key()` (method), 330
`(aries_cloudagent.transport.inbound.delivery_queue.DeliveryQueue, 330)` (method), 330
`remove_model()` (method), 84
`(aries_cloudagent.messaging.decorators.base.BaseDecorator, 84)` (method), 84
`remove_skipped_values()` (method), 99
`(aries_cloudagent.messaging.models.base.BaseModelSchema, 99)` (method), 99
`remove_wallet()` (method), 58
`(aries_cloudagent.indy.sdk.wallet_setup.IndyWalletSetup, 58)` (method), 58
`remove_wallet()` (method), 122
`(aries_cloudagent.multitenant.manager.MultitenantManager, 122)` (method), 122
`remove_webhook_target()` (method), 3
`(aries_cloudagent.admin.base_server.BaseAdminServer, 3)` (method), 3
`remove_webhook_target()` (method), 346
`(aries_cloudagent.admin.server.AdminServer, 346)` (method), 346
`replace_local_did_metadata()` (method), 359
`(aries_cloudagent.wallet.base.BaseWallet, 359)` (method), 359
`replace_local_did_metadata()` (method), 367
`(aries_cloudagent.wallet.in_memory.InMemoryWallet, 367)` (method), 367
`replace_local_did_metadata()` (method), 370
`(aries_cloudagent.wallet.indy.IndySdkWallet, 370)` (method), 370
`replace_signatures()` (method), 109
`(aries_cloudagent.messaging.agent_message.AgentMessageSchema, 109)` (method), 109
`replace_signing_key_metadata()` (method), 359
`(aries_cloudagent.wallet.base.BaseWallet, 359)` (method), 359
`replace_signing_key_metadata()` (method), 367
`(aries_cloudagent.wallet.in_memory.InMemoryWallet, 367)` (method), 367
`replace_signing_key_metadata()` (method), 370
`(aries_cloudagent.wallet.indy.IndySdkWallet, 370)` (method), 370
`replacement_id` (attribute), 228
`(aries_cloudagent.protocols.issue_credential.v2_0.message, 228)` (attribute), 228
`replacement_id` (attribute), 229
`(aries_cloudagent.protocols.issue_credential.v2_0.message, 229)` (attribute), 229
`reply_mode` (attribute), 336
`(aries_cloudagent.transport.inbound.session.InboundSession, 336)` (attribute), 336
`REPLY_MODE_NONE` (attribute), 333
`(aries_cloudagent.transport.inbound.receipt.MessageReceipt, 333)` (attribute), 333
`REPLY_MODE_THREAD` (attribute), 333
`(aries_cloudagent.transport.inbound.receipt.MessageReceipt, 333)` (attribute), 333
`reply_thread_ids` (attribute), 336
`(aries_cloudagent.transport.inbound.session.InboundSession, 336)` (attribute), 336
`reply_verkeys` (attribute), 336
`(aries_cloudagent.transport.inbound.session.InboundSession, 336)` (attribute), 336
`REQUEST` (attribute), 322
`(aries_cloudagent.connections.models.conn_record.ConnRecord, 322)` (attribute), 322
`request_attach` (attribute), 251
`(aries_cloudagent.protocols.out_of_band.v1_0.message, 251)` (attribute), 251
`request_denied()` (method), 173
`(aries_cloudagent.protocols.coordinate_mediation.v1_0.util.predicate, 173)` (method), 173
`request_end()` (method), 346
`(aries_cloudagent.transport.stats.StatsTracer, 346)` (method), 346

`request_granted()` (method), 344
`(aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.ConnectionManager)` (method), 173
`request_id(aries_cloudagent.connections.models.conn_record.ConnRecordSchema)` (in module `aries_cloudagent.messaging.models.base`), 35
`request_mediation()` (in module `aries_cloudagent.protocols.coordinate_mediation.v1_0.routes`), 99
`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.inbound_connection()` (method), 177
`REQUEST_NOT_ACCEPTED` (method), 149
`(aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportReason)` (attribute), 142
`REQUEST_NOT_ACCEPTED` (method), 48
`(aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportReason)` (in module `aries_cloudagent.messaging.models.base`), 180
`request_presentations_attach` (method), 99
`(aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof_request.PresentProofRequestSchema)` (attribute), 271
`REQUEST_PROCESSING_ERROR` (resource) (in module `aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportReason`), 142
`(aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportReason)` (attribute), 180
`REQUEST_PROCESSING_ERROR` (RESPONDER) (aries_cloudagent.connections.models.conn_record.ConnRecordSchema) (attribute), 32
`request_start()` (aries_cloudagent.transport.stats.StatsTracer) (method), 346
`RequestContext` (class in `RESPONSE` (aries_cloudagent.connections.models.conn_record.ConnRecordSchema) (attribute), 32
`aries_cloudagent.messaging.request_context`), 110
`response_buffered` (aries_cloudagent.transport.inbound.session.InboundSession) (method), 235
`requested_attributes` (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRequestSchema) (attribute), 279
`RESPONSE_NOT_ACCEPTED` (aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportReason) (attribute), 142
`requested_attributes` (aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentProofRequestSchema) (attribute), 281
`RESPONSE_NOT_ACCEPTED` (aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportReason) (attribute), 180
`requested_predicates` (aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRequestSchema) (attribute), 279
`RESPONSE_PROCESSING_ERROR` (aries_cloudagent.protocols.connections.v1_0.messages.problem_report.ProblemReportReason) (attribute), 142
`requested_predicates` (aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentProofRequestSchema) (attribute), 281
`RESPONSE_PROCESSING_ERROR` (aries_cloudagent.protocols.didexchange.v1_0.messages.problem_report.ProblemReportReason) (attribute), 180
`REQUESTER` (aries_cloudagent.connections.models.conn_record.ConnRecordSchema) (attribute), 32
`requests_attach(aries_cloudagent.protocols.issue_credential.v1_0.messages.credential_request.CredentialRequestSchema)` (attribute), 208
`(aries_cloudagent.protocols.trustping.v1_0.messages.ping.PingSchema)` (attribute), 262
`requests_attach(aries_cloudagent.protocols.issue_credential.v1_0.messages.cred_request.V20CredRequestSchema)` (attribute), 232
`restrictions(aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRequestSchema)` (attribute), 279
`required(aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form.MenuFormParamSchema)` (attribute), 129
`restrictions(aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRequestSchema)` (attribute), 279
`requires_external_key` (aries_cloudagent.wallet.models.wallet_record.WalletRecordSchema) (attribute), 356
`(aries_cloudagent.admin.server.AdminModulesSchema)` (attribute), 5
`reset()` (aries_cloudagent.transport.queue.base.BaseMessageQueue) (method), 343
`(aries_cloudagent.cache.base.CacheKeyLock)` (attribute), 9
`reset()` (aries_cloudagent.transport.queue.basic.BasicMessageQueue) (method), 343
`(aries_cloudagent.ledger.routes.TAAResultSchema)` (attribute), 279

`aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.tribute)`, 311
`177` `revoc_reg_id(aries_cloudagent.protocols.issue_credential.v1_0.model`
`retrieve_request()` `attribute)`, 212
`(aries_cloudagent.connections.models.conn_record.ConnRecord`
`method)`, 35 `attribute)`, 311
`retrieve_tails()` `(aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry`
`method)`, 313 `attribute)`, 212
`return_invitation()` `revocation_registry_size`
`(aries_cloudagent.protocols.introduction.v0_1.base_service(BriefIntroductionService`
`method)`, 196 `attribute)`, 75
`return_invitation()` `RevocationError`, 313
`(aries_cloudagent.protocols.introduction.v0_1.demovoice.DemoIntroductionService` `class` `in`
`method)`, 197 `aries_cloudagent.revocation.manager)`, 314
`return_route(aries_cloudagent.messaging.decorators.ResponseDecoratorResponseDecorator`
`attribute)`, 95 `RevocationModuleResponseSchema` `(class` `in`
`return_route_thread` `aries_cloudagent.revocation.routes)`, 317
`(aries_cloudagent.messaging.decorators.transport_decorator.TransportDecorator`
`attribute)`, 95 `RevocationRegistry` `(class` `in`
`return_to_session()` `aries_cloudagent.revocation.models.revocation_registry)`,
`(aries_cloudagent.transport.inbound.manager.InboundTransportManager`
`method)`, 332 `RevocationRegistryBadSizeError`, 314
`return_undelivered()` `revoke()` `(in` `module`
`(aries_cloudagent.transport.inbound.manager.InboundTransportManager`
`method)`, 332 `aries_cloudagent.revocation.routes)`, 319
`REV_REG_CACHE(aries_cloudagent.revocation.indy.IndyRevocation`
`attribute)`, 314 `method)`, 315
`rev_reg_id(aries_cloudagent.holder.routes.CredBriefSchema`
`attribute)`, 49 `revoke_credential_by_cred_ex_id()`
`rev_reg_id(aries_cloudagent.protocols.issue_credential.v2_0.model`
`attribute)`, 235 `(aries_cloudagent.revocation.manager.RevocationManager`
`rev_reg_id(aries_cloudagent.protocols.present_proof.v1_0.routes`
`attribute)`, 278 `method)`, 62
`rev_reg_id(aries_cloudagent.revocation.models.issuer_cred_rev_record`
`attribute)`, 307 `(aries_cloudagent.indy.sdk.issuer.IndySdkIssuer`
`rev_reg_id(aries_cloudagent.revocation.routes.CredRevRecordQueryStringSchema`
`attribute)`, 316 `revoked(aries_cloudagent.holder.routes.CredRevokedResultSchema`
`rev_reg_id(aries_cloudagent.revocation.routes.RevRegIdMatchInfoSchema)`, 49
`attribute)`, 317 `RevokeRequestSchema` `(class` `in`
`rev_reg_ids(aries_cloudagent.revocation.routes.RevRegsCreatedSchema`
`attribute)`, 317 `aries_cloudagent.revocation.routes)`, 318
`rev_regs_created()` `(in` `module` `aries_cloudagent.revocation.routes)`, 317
`aries_cloudagent.revocation.routes)`, 319 `RevRegCreateRequestSchema` `(class` `in`
`revealed(aries_cloudagent.protocols.present_proof.v1_0.routes`
`attribute)`, 279 `RevRegIdMatchInfoSchema` `(class` `in`
`REVEALED_CLAIM(aries_cloudagent.protocols.present_proof.v1_0.messages`
`attribute)`, 264 `RevRegIssuedResultSchema` `(class` `in`
`revoc_def_type(aries_cloudagent.revocation.models.issuer_rev_reg_def`
`attribute)`, 310 `RevRegResultSchema` `(class` `in`
`REVOC_DEF_TYPE_CL` `aries_cloudagent.revocation.routes)`, 317
`(aries_cloudagent.revocation.models.issuer_rev_reg_def.RevRegRecord` `(class` `in`
`attribute)`, 309 `aries_cloudagent.revocation.routes)`, 317
`revoc_reg_def(aries_cloudagent.revocation.models.issuer_rev_reg_def.RevRegRecordSchema` `class` `in`
`attribute)`, 310 `aries_cloudagent.revocation.routes)`, 317
`revoc_reg_entry(aries_cloudagent.revocation.models.issuer_rev_reg_def.RevRegRecordSchema` `messages.invitation.HS`

[attribute\), 249](#)
[rfc \(aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.HSProtoSpec](#)
[attribute\), 249](#)
[rfc160 \(aries_cloudagent.connections.models.conn_record.ConnRecordState\), 65](#)
[attribute\), 32](#)
[rfc160 \(aries_cloudagent.connections.models.conn_record.ConnRecordState\), 169](#)
[attribute\), 33](#)
[RFC160 \(aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.HSProto](#)
[attribute\), 248](#)
[rfc23 \(aries_cloudagent.connections.models.conn_record.ConnRecordState\), 273](#)
[attribute\), 32](#)
[rfc23 \(aries_cloudagent.connections.models.conn_record.ConnRecordState\), 42](#)
[attribute\), 33](#)
[RFC23 \(aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation.HSProto](#)
[attribute\), 248](#)
[rfc23_state \(aries_cloudagent.connections.models.conn_record.ConnRecordState\), 7](#)
[attribute\), 35](#)
[rfc23_state \(aries_cloudagent.connections.models.conn_record.ConnRecordState\), 359](#)
[attribute\), 35](#)
[rfc23strict \(\) \(aries_cloudagent.connections.models.conn_record.ConnRecordState](#)
[method\), 33](#)
[role \(aries_cloudagent.ledger.routes.RegisterLedgerNymQueryStringSchema\), 367](#)
[attribute\), 71](#)
[role \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecordState\), 370](#)
[attribute\), 170](#)
[role \(aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.GetKeyListQueryStringSchema\) \(](#)
[attribute\), 174](#)
[role \(aries_cloudagent.protocols.issue_credential.v1_0.models.cred_ex_record.V10CredentialExchangeSchema](#)
[attribute\), 212](#)
[role \(aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeListQueryStringSchemaMemoryWallet](#)
[attribute\), 216](#)
[role \(aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecordSchema](#)
[attribute\), 238](#)
[role \(aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredExRecordListQueryStringSchema](#)
[attribute\), 242](#)
[role \(aries_cloudagent.protocols.present_proof.v1_0.models.present_proof.v1_0.messages.present_proof.v1_0.messages.route_query_record.RouteQueryRecordState\), 65](#)
[attribute\), 274](#)
[role \(aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationExchangeListQueryStringSchema](#)
[attribute\), 280](#)
[role \(aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecordState\), 65](#)
[attribute\), 296](#)
[Role \(class in aries_cloudagent.ledger.base\), 65](#)
[ROLE_CLIENT \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.mediation_record.MediationRecord](#)
[attribute\), 169](#)
[ROLE_CLIENT \(aries_cloudagent.protocols.routing.v1_0.models.route_query_record.RouteQueryRecordState\), 289](#)
[attribute\), 295](#)
[ROLE HOLDER \(aries_cloudagent.protocols.issue_credential.v1_0.models.cred_ex_record.V10CredentialExchange](#)
[attribute\), 211](#)
[ROLE HOLDER \(aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord](#)
[attribute\), 237](#)
[ROLE_ISSUER \(aries_cloudagent.protocols.issue_credential.v1_0.models.cred_ex_record.V10CredentialExchange](#)
[attribute\), 211](#)
[ROLE_ISSUER \(aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord](#)
[attribute\), 237](#)

[289](#) `RouteUpdated.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_updated`),
`RouteQueryRequestSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_query_request`),
[289](#) `RouteUpdatedSchema` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_updated`),
`RouteQueryResponse` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_query_response`),
[290](#) `RouteUpdatedSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_updated`),
`RouteQueryResponse.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_query_response`),
[290](#) `RouteUpdateRequest` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_request`),
`RouteQueryResponseHandler` (class in `aries_cloudagent.protocols.routing.v1_0.handlers.route_query_response_handler`),
[287](#) `RouteUpdateRequest.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_request`),
`RouteQueryResponseSchema` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_query_response`),
[290](#) `RouteUpdateRequestHandler` (class in `aries_cloudagent.protocols.routing.v1_0.handlers.route_update_request_handler`),
`RouteQueryResponseSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_query_response`),
[290](#) `RouteUpdateRequestSchema` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_request`),
`RouteQueryResult` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_query_result`),
[294](#) `RouteUpdateRequestSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_request`),
`RouteQueryResult.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_query_result`),
[294](#) `RouteUpdateResponse` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_response`),
`RouteQueryResultSchema` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_query_result`),
[294](#) `RouteUpdateResponse.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_response`),
`RouteQueryResultSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_query_result`),
[294](#) `RouteUpdateResponseHandler` (class in `aries_cloudagent.protocols.routing.v1_0.handlers.route_update_response_handler`),
`RouteRecord` (class in `aries_cloudagent.protocols.routing.v1_0.handlers.route_update_request_handler`),
[295](#) `RouteUpdateResponseSchema` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_response`),
`RouteRecord.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_record`),
[295](#) `RouteUpdateResponseSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.messages.route_update_response`),
`RouteRecordSchema` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_record`),
[296](#) `RouteUpdateSchema` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_update`),
`RouteRecordSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_record`),
[296](#) `RouteUpdateSchema.Meta` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_update`),
`routes` (`aries_cloudagent.protocols.routing.v1_0.messages.route_query_response` attribute), [290](#)
`RouteUpdate` (class in `ROUTING_DID_RECORD_TYPE`), [297](#)
`RouteUpdate.Meta` (class in `routing_keys` (`aries_cloudagent.connections.models.connection_target` attribute)), [37](#)
`RouteUpdated` (class in `aries_cloudagent.protocols.routing.v1_0.models.route_updated`), [298](#)

Index 455

`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 264`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 266`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 265`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 268`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 268`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 269`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 270`
`schema_class(aries_cloudagent.protocols.present_proof.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.CredDefMeta`
`attribute), 273`
`schema_class(aries_cloudagent.protocols.problem_report.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.V10CredDefMeta`
`attribute), 286`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.V20CredDefMeta`
`attribute), 288`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 289`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 290`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 291`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 292`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 293`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 294`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 294`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 295`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 297`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 298`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 302`
`schema_class(aries_cloudagent.protocols.routing.v1_0.messages.issue_credential_response_v1_0.messages.issue_credential_response_v1_0.routes.IndyCredDefMeta`
`attribute), 302`
`schema_class(aries_cloudagent.revocation.models.indy_revocation_interval.RevocationInterval`
`attribute), 305`
`schema_class(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord`
`attribute), 306`
`schema_class(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord`
`attribute), 309`
`schema_class(aries_cloudagent.wallet.models.wallet_record.WalletRecord`
`attribute), 356`
`schema_id(aries_cloudagent.holder.routes.CredBriefSchema`
`attribute), 49`

<code>attribute</code>), 207	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_name</code> (<code>aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeSchema</code>), <code>attribute</code>), 216	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_name</code> (<code>aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeSchema</code>), <code>attribute</code>), 217	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_name</code> (<code>aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredentialExchangeSchema</code>), <code>attribute</code>), 242	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_name</code> (<code>aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentProofSchema</code>), <code>attribute</code>), 279	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.messaging.credential_definition.is101.CredDefQueryStringSchema</code>), <code>attribute</code>), 76	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.messaging.schemas.routes.SchemasAndRequestsSchema</code>), <code>attribute</code>), 105	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.messaging.schemas.util.SchemaQueryStringSchema</code>), <code>attribute</code>), 106	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.protocols.issue_credential.v1_0.messages.CredentialProposalCredentialProposalSchema</code>), <code>attribute</code>), 207	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeSchema</code>), <code>attribute</code>), 216	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeSchema</code>), <code>attribute</code>), 217	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredentialExchangeSchema</code>), <code>attribute</code>), 243	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schema_version</code> (<code>aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentProofSchema</code>), <code>attribute</code>), 279	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>SchemaGetResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 104	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schemaId</code> (<code>aries_cloudagent.messaging.credential_definition.is101.CredDefQueryStringSchema</code>), <code>attribute</code>), 74	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>SchemaIdMatchInfoSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 104	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>SchemaQueryStringSchema</code> (class in <code>aries_cloudagent.messaging.schemas.util</code>), 106	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schemas_created</code> () (in module <code>aries_cloudagent.messaging.schemas.routes</code>), 105	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schemas_get_schema</code> () (in module <code>aries_cloudagent.messaging.schemas.routes</code>), 106	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>schemas_send_schema</code> () (in module <code>aries_cloudagent.messaging.schemas.routes</code>), 106	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>SchemaSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>SchemasCreatedResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105
<code>SchemaSendRequestSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105	<code>SchemaSendResultsSchema</code> (class in <code>aries_cloudagent.messaging.schemas.routes</code>), 105

`send_outbound()` (`aries_cloudagent.messaging.responder.BaseResponder` (class in `aries_cloudagent.messaging.decorators.thread_decorator`), 112)
`send_outbound()` (`aries_cloudagent.messaging.responder.MockResponder` (class in `aries_cloudagent.transport.inbound.receipt.MessageReceipt`), 113)
`send_presentation_ack()` (`aries_cloudagent.protocols.present_proof.v1_0.manager.PresentationManager` (class in `aries_cloudagent.protocols.actionmenu.v1_0.routes`), 277)
`send_reply()` (`aries_cloudagent.messaging.responder.BaseResponder` (class in `aries_cloudagent.protocols.basicmessage.v1_0.routes`), 112)
`send_reply()` (`aries_cloudagent.messaging.responder.MockResponder` (class in `aries_cloudagent.transport.inbound.receipt.MessageReceipt`), 113)
`SEND_REQ_AFTER_CONNECTION` (`aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.CoordinationManager` (class in `aries_cloudagent.messaging.schemas.routes.SchemaSchema`), 171)
`send_rev_reg_def()` (`aries_cloudagent.revocation.routes`), 319
`send_rev_reg_entry()` (`aries_cloudagent.revocation.routes`), 320
`send_revoc_reg_def()` (`aries_cloudagent.ledger.base.BaseLedger` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), 65)
`send_revoc_reg_def()` (`aries_cloudagent.ledger.indy.IndySdkLedger` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), 70)
`send_revoc_reg_entry()` (`aries_cloudagent.ledger.base.BaseLedger` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), 65)
`send_revoc_reg_entry()` (`aries_cloudagent.ledger.indy.IndySdkLedger` (class in `aries_cloudagent.ledger.indy.IndySdkLedger`), 70)
`send_webhook()` (`aries_cloudagent.admin.base_server.BaseAdminServer` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 3)
`send_webhook()` (`aries_cloudagent.admin.server.AdminServer` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 5)
`send_webhook()` (`aries_cloudagent.admin.server.AdminServer` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 6)
`send_webhook()` (`aries_cloudagent.core.dispatcher.Dispatcher` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 42)
`send_webhook()` (`aries_cloudagent.messaging.models.base_record.BaseRecord` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 103)
`send_webhook()` (`aries_cloudagent.messaging.responder.BaseResponder` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 112)
`send_webhook()` (`aries_cloudagent.messaging.responder.MockResponder` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 113)
`send_webhook()` (`aries_cloudagent.protocols.actionmenu.v1_0.driver.MenuService` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 132)
`sender` (`aries_cloudagent.wallet.crypto.PackRecipientHeaderSchema` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 361)
`sender_did` (`aries_cloudagent.transport.inbound.receipt.MessageReceipt` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 334)
`sender_key` (`aries_cloudagent.connections.models.connection_target.ConnectionTarget` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 37)
`sender_order` (`aries_cloudagent.messaging.decorators.thread_decorator` (class in `aries_cloudagent.connections.models.diddoc.DIDDoc`), 89)

ServiceSchema.Meta (class in set_rev_reg_state() (in module
aries_cloudagent.protocols.out_of_band.v1_0.messages.service), 320
255 set_signature() (aries_cloudagent.messaging.agent_message.AgentM
session (aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager
attribute), 149 set_state() (aries_cloudagent.revocation.models.issuer_cred_rev_reco
session (aries_cloudagent.protocols.didexchange.v1_0.manager.DIDExchangeManager
attribute), 185 set_state() (aries_cloudagent.revocation.models.issuer_rev_reg_reco
session (aries_cloudagent.protocols.out_of_band.v1_0.manager.OutOfBandManager
attribute), 259 set_tails_file_public_uri()
session() (aries_cloudagent.admin.request_context.AdminRequestContext), 310
method), 4 set_value() (aries_cloudagent.config.settings.Settings
session() (aries_cloudagent.core.in_memory.InMemoryProfile), 44
method), 44 (aries_cloudagent.protocols.coordinate_mediation.v1_0.manager
session() (aries_cloudagent.core.profile.Profile attribute), 171
method), 46 set_urlsafes_b64() (in module
session() (aries_cloudagent.indy.sdk.profile.IndySdkProfile aries_cloudagent.wallet.util), 374
method), 55 set_value() (aries_cloudagent.config.settings.Settings
session() (aries_cloudagent.messaging.request_context.RequestContext), 21
method), 111 SetRevRegStateQueryStringSchema (class in
set (aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix aries_cloudagent.revocation.routes), 318
attribute), 304 settings (aries_cloudagent.admin.request_context.AdminRequestContext
set () (aries_cloudagent.cache.base.BaseCache attribute), 4
method), 8 settings (aries_cloudagent.config.injection_context.InjectionContext
set () (aries_cloudagent.cache.in_memory.InMemoryCache attribute), 17
method), 9 settings (aries_cloudagent.config.injector.Injector at-
set () (aries_cloudagent.connections.models.diddoc.DIDDoc attribute), 18
method), 23 settings (aries_cloudagent.core.profile.Profile at-
set () (aries_cloudagent.connections.models.diddoc.diddoc.DIDDoc attribute), 46
method), 26 settings (aries_cloudagent.core.profile.ProfileSession
set_cached_key() (aries_cloudagent.messaging.models.base_record.BaseRecord
class method), 103 settings (aries_cloudagent.messaging.request_context.RequestContext
set_default() (aries_cloudagent.config.settings.Settings attribute), 111
method), 21 settings (aries_cloudagent.wallet.models.wallet_record.WalletRecord
set_default_mediator() attribute), 357
(aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediatorManager
method), 173 attribute), 357
set_default_mediator() (in module Settings (class in aries_cloudagent.config.settings),
aries_cloudagent.protocols.coordinate_mediation.v1_0.routes), 177
177 SettingsError, 15
set_default_mediator_by_id() setup() (aries_cloudagent.core.conductor.Conductor
(aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.MediatorManager
method), 173 setup() (aries_cloudagent.core.dispatcher.Dispatcher
method), 42
set_did_endpoint() setup() (aries_cloudagent.transport.inbound.manager.InboundTransport
(aries_cloudagent.wallet.base.BaseWallet method), 332
method), 359 method), 340
set_did_endpoint() setup() (aries_cloudagent.transport.outbound.manager.OutboundTransp
(aries_cloudagent.wallet.indy.IndySdkWallet method), 340
method), 370 sha256 (aries_cloudagent.messaging.decorators.attach_decorator.AttachL
set_public_did() (aries_cloudagent.wallet.base.BaseWallet attribute), 79
method), 360 sha256_ (aries_cloudagent.messaging.decorators.attach_decorator.AttachL
set_response() (aries_cloudagent.transport.inbound.session.InboundSession), 332
method), 336 SHA256Hash (class in
set_result() (aries_cloudagent.cache.base.CacheKeyLock aries_cloudagent.messaging.valid), 117
method), 9 shutdown_app() (in module

`aries_cloudagent.commands.start()`, 11
`shutdown_handler()`
 (`aries_cloudagent.admin.server.AdminServer`
 method), 6
`sig_data(aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator`
 attribute), 88
`sign()` (`aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator` attribute), 79
`sign()` (in module `aries_cloudagent.messaging.jsonld.routes`), 97
`sign_credential()` (in module `aries_cloudagent.messaging.jsonld.credential`), 96
`sign_field()` (`aries_cloudagent.messaging.agent_message.AgentMessage` attribute), 107
`sign_message()` (`aries_cloudagent.wallet.base.BaseWallet` method), 360
`sign_message()` (`aries_cloudagent.wallet.in_memory.InMemoryWallet` method), 367
`sign_message()` (`aries_cloudagent.wallet.indy.IndySdkWallet` method), 370
`sign_message()` (in module `aries_cloudagent.wallet.crypto`), 364
`sign_pk_from_sk()` (in module `aries_cloudagent.wallet.crypto`), 364
`signature(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator` attribute), 80
`signature(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator` attribute), 82
`signature(aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator` attribute), 88
`signature_type(aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator` attribute), 88
`SignatureDecorator` (class in `aries_cloudagent.messaging.decorators.signature_decorator`), 86
`SignatureDecorator.Meta` (class in `aries_cloudagent.messaging.decorators.signature_decorator`), 87
`SignatureDecoratorSchema` (class in `aries_cloudagent.messaging.decorators.signature_decorator`), 87
`SignatureDecoratorSchema.Meta` (class in `aries_cloudagent.messaging.decorators.signature_decorator`), 88
`signatures(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator` attribute), 80
`signatures(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator` attribute), 82
`signed(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator` attribute), 80
`signed_doc(aries_cloudagent.messaging.jsonld.routes.SignResponseSchema` attribute), 97
`signed_fields(aries_cloudagent.messaging.agent_message.AgentMessage` attribute), 109
`signed_fields(aries_cloudagent.protocols.connections.v1_0.messages` attribute), 142
`signer(aries_cloudagent.messaging.decorators.signature_decorator.SignatureDecorator` attribute), 88
`SignRequestSchema` (class in `aries_cloudagent.messaging.jsonld.routes`), 97
`SignResponseSchema` (class in `aries_cloudagent.messaging.jsonld.routes`), 97
`skip_dump_only()` (`aries_cloudagent.messaging.models.base.BaseModelSchema` method), 99
`socket_connect_start()` (`aries_cloudagent.transport.stats.StatsTracer` method), 346
`some_dif_criterion` (`aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CreateCredential` attribute), 242
`specification()` (`aries_cloudagent.connections.models.diddoc.PublicKey` method), 28
`specification()` (`aries_cloudagent.connections.models.diddoc.PublicKey` method), 24
`split_credential_data(JWSSchema` attribute), 23
`split_credential_data(JWSSchema` attribute), 27
`split_credential_data(JWSSchema` attribute), 28
`split_credential_data(JWSSchema` attribute), 24
`stage_pending_registry()` (`aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord` method), 310
`stale_time(aries_cloudagent.messaging.decorators.timing_decorator.TimingDecorator` attribute), 91
`start(aries_cloudagent.holder.routes.CredentialsListQueryStringSchema` attribute), 49
`start(aries_cloudagent.protocols.present_proof.v1_0.routes.CredentialsListQueryStringSchema` attribute), 278
`start(aries_cloudagent.protocols.routing.v1_0.models.paginated.PaginatedCredentialsListQueryStringSchema` attribute), 294
`start()` (`aries_cloudagent.admin.base_server.BaseAdminServer` method), 90
`start()` (`aries_cloudagent.admin.server.AdminServer` method), 90
`start()` (`aries_cloudagent.core.conductor.Conductor` method), 90
`start()` (`aries_cloudagent.transport.inbound.base.BaseInboundTransport` method), 28
`start()` (`aries_cloudagent.transport.inbound.http.HttpTransport` method), 28
`start()` (`aries_cloudagent.transport.inbound.http.HttpTransport` method), 28

[start \(\) \(aries_cloudagent.transport.inbound.manager.InboundTransportManager attribute\), 332](#)
[start \(\) \(aries_cloudagent.transport.inbound.ws.WsTransport attribute\), 336](#)
[start \(\) \(aries_cloudagent.transport.outbound.base.BaseOutboundTransportManager attribute\), 337](#)
[start \(\) \(aries_cloudagent.transport.outbound.http.HttpTransport attribute\), 311](#)
[start \(\) \(aries_cloudagent.transport.outbound.manager.OutboundTransportManager attribute\), 340](#)
[start \(\) \(aries_cloudagent.transport.outbound.ws.WsTransport attribute\), 318](#)
[start \(\) \(aries_cloudagent.transport.outbound.ws.WsTransport attribute\), 343](#)
[start \(\) \(aries_cloudagent.utils.repeat.RepeatSequence attribute\), 352](#)
[start \(\) \(aries_cloudagent.utils.stats.Timer attribute\), 353](#)
[start_app \(\) \(in module aries_cloudagent.commands.start\), 11](#)
[start_introduction \(\) \(aries_cloudagent.protocols.introduction.v0_1.base_service.BaseIntroductionService attribute\), 197](#)
[start_introduction \(\) \(aries_cloudagent.protocols.introduction.v0_1.demo_service.DemoIntroductionService attribute\), 197](#)
[start_scope \(\) \(aries_cloudagent.config.injection_context.Injector attribute\), 17](#)
[start_transport \(\) \(aries_cloudagent.transport.inbound.manager.InboundTransportManager attribute\), 332](#)
[start_transport \(\) \(aries_cloudagent.transport.outbound.manager.OutboundTransportManager attribute\), 340](#)
[StartupError, 43](#)
[StartupGroup \(class in aries_cloudagent.config.argparse\), 13](#)
[state \(aries_cloudagent.messaging.models.base_record.BaseRecord attribute\), 103](#)
[state \(aries_cloudagent.protocols.connections.v1_0.routes.ConnectionsListQueryStringSchema attribute\), 151](#)
[state \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.coordinate_mediation.v1_0.record.MediationRecord attribute\), 169](#)
[state \(aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.coordinate_mediation.v1_0.record.MediationRecord attribute\), 175](#)
[state \(aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.v1_0.CredentialExchangeSchema attribute\), 212](#)
[state \(aries_cloudagent.protocols.issue_credential.v1_0.routes.V10CredentialExchangeListQueryStringSchema attribute\), 216](#)
[state \(aries_cloudagent.protocols.issue_credential.v2_0.models.credential_exchange.v2_0.CredExRecordSchema attribute\), 238](#)
[state \(aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredExRecordListQueryStringSchema attribute\), 242](#)
[state \(aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.v1_0.InvitationRecordSchema attribute\), 257](#)
[state \(aries_cloudagent.protocols.present_proof.v1_0.models.present_proof.v1_0.PresentationExchangeSchema attribute\), 274](#)
[state \(aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationExchangeSchema attribute\), 280](#)
[state \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredentialRevRecord attribute\), 307](#)
[state \(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute\), 311](#)
[state \(aries_cloudagent.revocation.routes.RevRegsCreatedQueryStringSchema attribute\), 318](#)
[state \(aries_cloudagent.revocation.routes.SetRevRegStateQueryStringSchema attribute\), 318](#)
[STATE_ACKED \(aries_cloudagent.protocols.issue_credential.v1_0.models.issue_credential.v1_0.models.credential.v1_0.models.credential attribute\), 211](#)
[STATE_ACTIVE \(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute\), 309](#)
[STATE_AWAIT_RESPONSE \(aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.v1_0.models.invitation attribute\), 256](#)
[STATE_CREDENTIAL_RECEIVED \(aries_cloudagent.protocols.issue_credential.v1_0.models.credential.v1_0.models.credential attribute\), 211](#)
[STATE_CREDENTIAL_RECEIVED \(aries_cloudagent.protocols.issue_credential.v2_0.models.credential.v2_0.models.credential attribute\), 237](#)
[STATE_DENIED \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.coordinate_mediation.v1_0.models.coordinate_mediation attribute\), 169](#)
[STATE_DONE \(aries_cloudagent.protocols.issue_credential.v2_0.models.credential.v2_0.models.credential attribute\), 237](#)
[STATE_DONE \(aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.v1_0.models.invitation attribute\), 256](#)
[STATE_DONE \(aries_cloudagent.transport.outbound.manager.OutboundTransportManager attribute\), 341](#)
[STATE_ENCODE \(aries_cloudagent.transport.outbound.manager.OutboundTransportManager attribute\), 341](#)
[STATE_FULL \(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute\), 309](#)
[STATE_GENERATED \(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute\), 309](#)
[STATE_GRANTED \(aries_cloudagent.protocols.coordinate_mediation.v1_0.models.coordinate_mediation.v1_0.models.coordinate_mediation attribute\), 169](#)
[STATE_INIT \(aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute\), 309](#)
[STATE_INITIAL \(aries_cloudagent.protocols.out_of_band.v1_0.models.invitation.v1_0.models.invitation attribute\), 256](#)
[STATE_ISSUED \(aries_cloudagent.protocols.issue_credential.v1_0.models.issue_credential.v1_0.models.credential.v1_0.models.credential attribute\), 211](#)
[STATE_ISSUED \(aries_cloudagent.protocols.issue_credential.v2_0.models.credential.v2_0.models.credential attribute\), 237](#)
[STATE_ISSUED \(aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredentialRevRecord attribute\), 307](#)
[STATE_NEW \(aries_cloudagent.transport.outbound.manager.OutboundTransportManager attribute\), 341](#)

STATE_OFFER_RECEIVED (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute), 211
STATE_OFFER_RECEIVED (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord attribute), 237
STATE_OFFER_SENT (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute), 211
STATE_OFFER_SENT (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord attribute), 237
STATE_PENDING (aries_cloudagent.transport.outbound.manager.OutboundManager attribute), 341
STATE_POSTED (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecord attribute), 309
STATE_PRESENTATION_ACKED (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute), 273
STATE_PRESENTATION_RECEIVED (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute), 273
STATE_PRESENTATION_SENT (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute), 273
STATE_PROPOSAL_RECEIVED (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute), 211
STATE_PROPOSAL_RECEIVED (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord attribute), 237
STATE_PROPOSAL_RECEIVED (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute), 273
STATE_PROPOSAL_SENT (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute), 211
STATE_PROPOSAL_SENT (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord attribute), 237
STATE_PROPOSAL_SENT (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute), 273
STATE_REQUEST (aries_cloudagent.protocols.coordinate_mediation.v1_0.models.coordinate_mediation_base.CoordinateMediationBase attribute), 169
STATE_REQUEST_RECEIVED (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute), 211
STATE_REQUEST_RECEIVED (aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex_record.V20CredExRecord attribute), 237
STATE_REQUEST_RECEIVED (aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange.V10PresentationExchange attribute), 273
STATE_REQUEST_SENT (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange.V10CredentialExchange attribute), 211

storage (aries_cloudagent.core.in_memory.InMemoryProfileSession method), 70
 attribute), 44
 storage_record (aries_cloudagent.messaging.models.base_record.BaseRecord attribute), 103
 StorageDuplicateError, 322
 StorageError, 322
 StorageNotFoundError, 322
 StorageRecord (class in aries_cloudagent.storage.record), 326
 StorageSearchError, 322
 store_credential () (aries_cloudagent.indy.holder.IndyHolder method), 60
 store_credential () (aries_cloudagent.indy.sdk.holder.IndySdkHolder method), 52
 store_credential () (aries_cloudagent.protocols.issue_credential.v1_0.manager.CredentialManager method), 214
 store_credential () (aries_cloudagent.protocols.issue_credential.v2_0.manager.CredentialManager method), 241
 store_did_document () (aries_cloudagent.connections.base_manager.BaseConnectionManager method), 38
 store_update_results () (aries_cloudagent.protocols.coordinate_mediation.v1_0.manager.CoordinationManager method), 173
 str_time (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator attribute), 94
 str_time (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator attribute), 94
 str_to_b64 () (in aries_cloudagent.wallet.util), 374
 str_to_datetime () (in aries_cloudagent.messaging.util), 113
 str_to_epoch () (in aries_cloudagent.messaging.util), 114
 strip_tag_prefix () (aries_cloudagent.messaging.models.base_record.BaseRecord class method), 103
 submit_label (aries_cloudagent.protocols.actionmenu.v1_0.model.ActionMenuFormSchema attribute), 128
 support_revocation (aries_cloudagent.messaging.credential_definitions_routes.CredentialDefinitionRoutes attribute), 75
 taa_accepted (aries_cloudagent.ledger.routes.TAAInfoSchema attribute), 72
 taa_digest () (aries_cloudagent.ledger.base.BaseLedger method), 65
 taa_digest () (aries_cloudagent.ledger.indy.IndySdkLedger method), 70
 taa_record (aries_cloudagent.ledger.routes.TAAInfoSchema attribute), 72
 taa_required (aries_cloudagent.ledger.routes.TAAInfoSchema attribute), 72
 taa_rough_timestamp () (aries_cloudagent.ledger.indy.IndySdkLedger method), 70
 TAAAcceptanceSchema (class in aries_cloudagent.ledger.routes), 71
 TAAAcceptSchema (class in aries_cloudagent.ledger.routes), 71
 TAAInfoSchema (class in aries_cloudagent.ledger.routes), 72
 TAARecordSchema (class in aries_cloudagent.ledger.routes), 72
 TAAResultSchema (class in aries_cloudagent.ledger.routes), 72
 tag (aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionRoutes attribute), 74
 tag (aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionRoutes attribute), 75
 tag (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute), 311
 tag (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry attribute), 313
 tag (aries_cloudagent.messaging.pack_message_schema.PackMessageSchema attribute), 361
 TAG_NAMES (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 33
 TAG_NAMES (aries_cloudagent.messaging.models.base_record.BaseRecord attribute), 101
 TAG_NAMES (aries_cloudagent.protocols.coordinate_mediation.v1_0.model.CoordinationManager attribute), 169
 TAG_NAMES (aries_cloudagent.protocols.issue_credential.v1_0.models.credential_definition_routes.CredentialDefinitionRoutes attribute), 211
 TAG_NAMES (aries_cloudagent.protocols.issue_credential.v2_0.models.credential_definition_routes.CredentialDefinitionRoutes attribute), 237
 TAG_NAMES (aries_cloudagent.protocols.issue_credential.v2_0.models.credential_definition_routes.CredentialDefinitionRoutes attribute), 233
 TAG_NAMES (aries_cloudagent.protocols.issue_credential.v2_0.models.credential_definition_routes.CredentialDefinitionRoutes attribute), 234
 TAG_NAMES (aries_cloudagent.protocols.out_of_band.v1_0.models.invitation_routes.InvitationRoutes attribute), 257
 TAG_NAMES (aries_cloudagent.messaging.pack_message_schema.PackMessageSchema attribute), 273
 TAG_NAMES (aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecord attribute), 295
 TAG_NAMES (aries_cloudagent.revocation.models.issuer_cred_rev_record.IssuerCredRevRecord attribute), 306
 TAG_NAMES (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevocationRecord attribute), 309
 TAG_NAMES (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute), 311

attribute), 356
 tag_query_match() (in module aries_cloudagent.storage.in_memory), 324
 tag_value_match() (in module aries_cloudagent.storage.in_memory), 324
 tags (aries_cloudagent.messaging.models.base_record.BaseRecord attribute), 151
 tails_hash (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecordSchema attribute), 311
 tails_hash (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry attribute), 313
 tails_local_path (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecordSchema attribute), 311
 tails_local_path (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry attribute), 313
 tails_path() (in module aries_cloudagent.indy.util), 62
 tails_public_uri (aries_cloudagent.revocation.models.issuer_rev_reg_record.IssuerRevRegRecordSchema attribute), 311
 tails_public_uri (aries_cloudagent.revocation.models.revocation_registry.RevocationRegistry attribute), 313
 tails_public_uri (aries_cloudagent.revocation.routes.rev_reg_update_tail.FileUriSchema attribute), 317
 TailsServerNotConfiguredError, 327
 target (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator attribute), 92
 target (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator attribute), 92
 target_connection_id (aries_cloudagent.protocols.introduction.v0_1.routes.IntroStrategy.StringSchema attribute), 198
 task (aries_cloudagent.utils.task_queue.PendingTask attribute), 353
 task_done() (aries_cloudagent.transport.queue.base.BaseMessageQueue method), 343
 task_done() (aries_cloudagent.transport.queue.basic.BasicMessageQueue method), 344
 task_exc_info() (in module aries_cloudagent.utils.task_queue), 355
 TaskQueue (class in aries_cloudagent.utils.task_queue), 353
 test_context() (aries_cloudagent.admin.request_context.AdminRequestContext class method), 4
 test_context() (aries_cloudagent.messaging.request_context.RequestContext class method), 111
 test_profile() (aries_cloudagent.core.in_memory.InMemoryProfile class method), 44
 TEST_PROFILE_NAME (aries_cloudagent.core.in_memory.InMemoryProfile attribute), 44
 test_session() (aries_cloudagent.core.in_memory.InMemoryProfile class method), 44
 text (aries_cloudagent.ledger.routes.TAAAcceptSchema attribute), 71
 text (aries_cloudagent.ledger.routes.TAARecordSchema attribute), 72
 their_did (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 35
 their_did (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_did (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_endpoint (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_endpoint (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_label (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 35
 their_label (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_public_did (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 35
 their_public_did (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_public_did (aries_cloudagent.protocols.didexchange.v1_0.routes.DidExchange attribute), 151
 their_role (aries_cloudagent.connections.models.conn_record.ConnRecord attribute), 35
 their_role (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_role (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_verkey (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 their_verkey (aries_cloudagent.protocols.connections.v1_0.routes.Connection attribute), 151
 thid (aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecorator attribute), 89
 thid (aries_cloudagent.messaging.decorators.thread_decorator.ThreadDecorator attribute), 89
 thread_id (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator attribute), 94
 thread_id (aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator attribute), 94
 thread_id (aries_cloudagent.protocols.issue_credential.v1_0.models.issue_credential_v1_0.models.issue_credential_v1_0.routes.V10IssueCredential attribute), 216
 thread_id (aries_cloudagent.protocols.issue_credential.v1_0.routes.V10IssueCredential attribute), 216
 thread_id (aries_cloudagent.protocols.issue_credential.v2_0.models.issue_credential_v2_0.models.issue_credential_v2_0.routes.V20IssueCredential attribute), 238
 thread_id (aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredential attribute), 242
 thread_id (aries_cloudagent.protocols.present_proof.v1_0.models.present_proof_v1_0.models.present_proof_v1_0.routes.V10PresentProof attribute), 274
 thread_id (aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentProof attribute), 280
 thread_id (aries_cloudagent.protocols.trustping.v1_0.routes.PingRequest attribute), 303

[thread_id \(aries_cloudagent.transport.inbound.receipt.MessageReceipt attribute\), 130](#)
[title \(aries_cloudagent.protocols.actionmenu.v1_0.routes.MenuJsonSchema attribute\), 125](#)
[ThreadDecorator \(class in aries_cloudagent.messaging.decorators.thread_decorator attribute\), 133](#)
[ThreadDecorator \(aries_cloudagent.holder.routes.CredRevokedQueryStringSchema attribute\), 49](#)
[ThreadDecorator.Meta \(class in to \(aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofReqNo aries_cloudagent.messaging.decorators.thread_decorator\), attribute\), 279](#)
[ThreadDecoratorSchema \(class in to \(aries_cloudagent.protocols.routing.v1_0.messages.forward.ForwardSch aries_cloudagent.messaging.decorators.thread_decorator attribute\), 289](#)
[ThreadDecoratorSchema \(aries_cloudagent.revocation.models.indy.NonRevocationIntervalScheme attribute\), 305](#)
[ThreadDecoratorSchema.Meta \(class in to_dict \(\) \(aries_cloudagent.connections.models.diddoc.PublicKey aries_cloudagent.messaging.decorators.thread_decorator\), method\), 24](#)
[threshold \(aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_preview.PresPredSpecSchema attribute\), 266](#)
[time \(aries_cloudagent.ledger.routes.TAAAcceptanceSchema method\), 25](#)
[time_noticed \(aries_cloudagent.protocols.problem_report.v1_0.messages.ProblemReportSchema attribute\), 286](#)
[time_now \(\) \(in module aries_cloudagent.messaging.util\), 114](#)
[timeout \(\) \(aries_cloudagent.utils.repeat.RepeatAttempt \(aries_cloudagent.ledger.base.Role method\), 352](#)
[Timer \(class in aries_cloudagent.utils.stats\), 353](#)
[timer \(\) \(aries_cloudagent.utils.stats.Collector method\), 352](#)
[timestamp \(aries_cloudagent.messaging.decorators.trace_decorator attribute\), 94](#)
[timestamp \(aries_cloudagent.messaging.decorators.trace_decorator attribute\), 94](#)
[timestamp \(aries_cloudagent.protocols.present_proof.v1_0.routes.IndyReqv1_0.CredsRequestedPredSchema attribute\), 280](#)
[timestamp \(\) \(aries_cloudagent.revocation.models.indy.NonRevocationInterval method\), 305](#)
[TimingDecorator \(class in aries_cloudagent.messaging.decorators.timing_decorator attribute\), 118](#)
[TimingDecorator.Meta \(class in aries_cloudagent.messaging.decorators.timing_decorator attribute\), 118](#)
[TimingDecoratorSchema \(class in aries_cloudagent.messaging.decorators.timing_decorator attribute\), 90](#)
[TimingDecoratorSchema.Meta \(class in total \(aries_cloudagent.protocols.routing.v1_0.models.paginated.Pagina aries_cloudagent.messaging.decorators.timing_decorator\), attribute\), 294](#)
[title \(aries_cloudagent.protocols.actionmenu.v1_0.messages.menu_form.MenuFormSchema attribute\), 128](#)
[title \(aries_cloudagent.protocols.actionmenu.v1_0.models.menu_form.MenuFormParamSchema attribute\), 129](#)
[title \(aries_cloudagent.protocols.actionmenu.v1_0.models.menu_option.MenuOptionSchema attribute\), 120](#)

[trace \(aries_cloudagent.protocols.issue_credential.v2_0.routes.V20IssueCredentialSchemaCredential.IndySdkProfile.IndySdkProfile attribute\), 244](#)
[trace \(aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationCreateRequestSchemaCreateRequestSchema.request_context.RequestContext attribute\), 280](#)
[trace \(aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationProposalRequestSchemaPresentationProposalRequestSchema in attribute\), 281](#)
[trace \(aries_cloudagent.protocols.present_proof.v1_0.routes.V10PresentationRequestSchemaPresentationRequestSchema in attribute\), 281](#)
[trace \(aries_cloudagent.utils.tracing.AdminAPIMessageTracingSchemaAdminAPIMessageTracingSchema in attribute\), 355](#)
[trace_event\(\) \(in module TransportDecoratorSchema \(class in aries_cloudagent.utils.tracing\), 355](#)
[trace_reports \(aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 92](#)
[trace_reports \(aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 92](#)
[traced_type \(aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 94](#)
[traced_type \(aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 95](#)
[TraceDecorator \(class in aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 66](#)
[TraceDecorator \(class in aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 74](#)
[TraceDecorator.Meta \(class in type \(aries_cloudagent.wallet.crypto.PackRecipientsSchema \(class in attribute\), 362](#)
[TraceDecoratorSchema \(class in type \(aries_cloudagent.connections.models.diddoc.PublicKey \(class in attribute\), 24](#)
[TraceDecoratorSchema \(class in type \(aries_cloudagent.connections.models.diddoc.PublicKey \(class in attribute\), 28](#)
[TraceDecoratorSchema.Meta \(class in type \(aries_cloudagent.connections.models.diddoc.Service \(class in attribute\), 25](#)
[TraceDecoratorSchema.Meta \(class in type \(aries_cloudagent.connections.models.diddoc.service.Service \(class in attribute\), 29](#)
[TraceReport \(class in aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 92](#)
[TraceReport \(class in aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 87](#)
[TraceReport.Meta \(class in aries_cloudagent.messaging.decorators.trace_decorator.TraceDecorator \(class in attribute\), 93](#)
[TraceReportSchema \(class in unknown \(aries_cloudagent.connections.models.connection_target.ConnectionTarget \(class in attribute\), 37](#)
[TraceReportSchema \(class in unknown \(aries_cloudagent.messaging.ack.message.AckSchema.Meta \(class in attribute\), 74](#)
[TraceReportSchema \(class in unknown \(aries_cloudagent.messaging.agent_message.AgentMessageSchema \(class in attribute\), 109](#)
[tracing_enabled\(\) \(in module unknown \(aries_cloudagent.messaging.decorators.attach_decorator.AttachDecorator \(class in attribute\), 80](#)
[tracking_uri \(aries_cloudagent.protocols.problem_report.v1_0.message.ProblemReportSchemaProblemReportSchema in attribute\), 286](#)
[transaction\(\) \(aries_cloudagent.admin.request_context.AdminRequestContextAdminRequestContext in attribute\), 5](#)
[transaction\(\) \(aries_cloudagent.admin.request_context.AdminRequestContextAdminRequestContext in attribute\), 82](#)
[transaction\(\) \(aries_cloudagent.core.in_memory.InMemoryProfileInMemoryProfile in attribute\), 44](#)
[transaction\(\) \(aries_cloudagent.core.in_memory.InMemoryProfileInMemoryProfile in attribute\), 82](#)
[transaction\(\) \(aries_cloudagent.core.profile.ProfileProfile in attribute\), 46](#)
[transaction\(\) \(aries_cloudagent.core.profile.ProfileProfile in attribute\), 83](#)

- unknown (aries_cloudagent.messaging.decorators.localization_decorator (aries_cloudagent.protocols.didexchange.v1_0.messages.request attribute), 85
- unknown (aries_cloudagent.messaging.decorators.please_ack_decorator (aries_cloudagent.protocols.didexchange.v1_0.messages.response attribute), 86
- unknown (aries_cloudagent.messaging.decorators.signature_decorator (aries_cloudagent.protocols.didexchange.v1_0.messages.disclose.D attribute), 88
- unknown (aries_cloudagent.messaging.decorators.thread_decorator (aries_cloudagent.protocols.didexchange.v1_0.messages.query.Q attribute), 89
- unknown (aries_cloudagent.messaging.decorators.timing_decorator (aries_cloudagent.protocols.didexchange.v1_0.messages.invitation attribute), 91
- unknown (aries_cloudagent.messaging.decorators.trace_decorator (aries_cloudagent.protocols.didexchange.v1_0.messages.invitation attribute), 92
- unknown (aries_cloudagent.messaging.decorators.trace_decorator (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 94
- unknown (aries_cloudagent.messaging.decorators.transport_decorator (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 95
- unknown (aries_cloudagent.messaging.models.base.OpenAPISchema (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 99
- unknown (aries_cloudagent.messaging.models.openapi.OpenAPISchema (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 104
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 125
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 126
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 127
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 128
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 129
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 130
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 136
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 140
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 141
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 142
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 143
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 144
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 158
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 161
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 170
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 180
- unknown (aries_cloudagent.protocols.actionmenu.v1_0.messages.menu (aries_cloudagent.protocols.issue_credential.v1_0.messages.cred attribute), 181

unknown (aries_cloudagent.protocols.out_of_band.v1_0.messages.send_offer.Schema.Meta attribute), 255 update_endpoint_for_did() (aries_cloudagent.protocols.out_of_band.v1_0.messages.send_offer.Schema.Meta attribute), 255

unknown (aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof.Schema.Meta attribute), 268 presentation_ack(PresentationAckSchema.Meta attribute), 269 (aries_cloudagent.ledger.indy.IndySdkLedger attribute), 269

unknown (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_proposal.PresentationProposalSchema.Meta attribute), 270 update_inbound() (aries_cloudagent.protocols.connections.v1_0.manager.ConnectionManager attribute), 270

unknown (aries_cloudagent.protocols.present_proof.v1_0.messages.presentation_request.PresentationRequestSchema.Meta attribute), 271 update_keylist() (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.coordinate_mediation_request.CoordinateMediationRequestSchema.Meta attribute), 271

unknown (aries_cloudagent.protocols.problem_report.v1_0.messages.problem_report.ProblemReportSchema.Meta attribute), 286 update_record() (aries_cloudagent.storage.base.BaseStorage attribute), 286

unknown (aries_cloudagent.protocols.routing.v1_0.messages.forward_ethereum_request.ForwardEthereumRequestSchema.Meta attribute), 289 update_record() (aries_cloudagent.storage.in_memory.InMemoryStorage attribute), 289

unknown (aries_cloudagent.protocols.routing.v1_0.messages.route_query_request.RouteQueryRequestSchema.Meta attribute), 290 update_record() (aries_cloudagent.storage.indy.IndySdkStorage attribute), 290

unknown (aries_cloudagent.protocols.routing.v1_0.messages.route_query_response.RouteQueryResponseSchema.Meta attribute), 290 update_rev_reg() (in module aries_cloudagent.protocols.routing.v1_0.messages.route_query_response.RouteQueryResponseSchema.Meta attribute), 291

unknown (aries_cloudagent.protocols.routing.v1_0.messages.route_update_request.RouteUpdateRequestSchema.Meta attribute), 291 update_routes() (aries_cloudagent.protocols.routing.v1_0.manager.RouteManager attribute), 291

unknown (aries_cloudagent.protocols.routing.v1_0.messages.route_update_response.RouteUpdateResponseSchema.Meta attribute), 292 update_settings() (aries_cloudagent.config.settings.Settings attribute), 292

unknown (aries_cloudagent.protocols.routing.v1_0.models.paginate_request.PaginateRequestSchema.Meta attribute), 293 (aries_cloudagent.ledger.indy.IndySdkLedger attribute), 293

unknown (aries_cloudagent.protocols.routing.v1_0.models.paginate_response.PaginateResponseSchema.Meta attribute), 294 (aries_cloudagent.config.base_context.ContextBuilder attribute), 294

unknown (aries_cloudagent.protocols.routing.v1_0.models.route_query_result.RouteQueryResultSchema.Meta attribute), 295 update_settings() (aries_cloudagent.config.settings.Settings attribute), 295

unknown (aries_cloudagent.protocols.routing.v1_0.models.route_update_request.RouteUpdateRequestSchema.Meta attribute), 296 (aries_cloudagent.config.settings.Settings attribute), 296

unknown (aries_cloudagent.protocols.routing.v1_0.models.route_update_response.RouteUpdateResponseSchema.Meta attribute), 297 (aries_cloudagent.messaging.request_context.RequestContext attribute), 297

unknown (aries_cloudagent.protocols.routing.v1_0.models.route_updated.RouteUpdatedSchema.Meta attribute), 298 update_settings() (aries_cloudagent.config.settings.Settings attribute), 298

unknown (aries_cloudagent.protocols.trustping.v1_0.messages.ping.PingSchema.Meta attribute), 302 (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute), 302

unknown (aries_cloudagent.protocols.trustping.v1_0.messages.ping_response.PingResponseSchema.Meta attribute), 303 (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute), 303

unknown (aries_cloudagent.wallet.models.wallet_record.WalletRecordSchema.Meta attribute), 357 (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.coordinate_mediation_request.CoordinateMediationRequestSchema.Meta attribute), 357

unpack() (aries_cloudagent.transport.pack_format.PackFormat attribute), 346 (aries_cloudagent.protocols.routing.v1_0.messages.route_update_request.RouteUpdateRequestSchema.Meta attribute), 346

unpack_message() (aries_cloudagent.wallet.base.BaseWallet attribute), 360 (aries_cloudagent.messaging.models.base_record.BaseRecord attribute), 360

unpack_message() (aries_cloudagent.wallet.in_memory.InMemoryWallet attribute), 367 (aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.coordinate_mediation_request.CoordinateMediationRequestSchema.Meta attribute), 367

unpack_message() (aries_cloudagent.wallet.indy.IndySdkWallet attribute), 371 (aries_cloudagent.protocols.coordinate_mediation.v1_0.routes.routes attribute), 371

unpad() (in module aries_cloudagent.wallet.util), 374 updates (aries_cloudagent.protocols.routing.v1_0.messages.route_update_request.RouteUpdateRequestSchema.Meta attribute), 374

unqualify(aries_cloudagent.protocols.didcomm_prefix.DIDCommPrefix attribute), 291 UpdateWalletRequestSchema (class in aries_cloudagent.protocols.coordinate_mediation.v1_0.messages.coordinate_mediation_request.CoordinateMediationRequestSchema attribute), 291

UNREVEALED_CLAIM(aries_cloudagent.protocols.present_proof.v1_0.messages.present_proof.Schema.Meta attribute), 264 119

update() (aries_cloudagent.config.settings.Settings attribute), 292 upload_tails_file() (aries_cloudagent.ledger.indy.IndySdkLedger attribute), 292

`(aries_cloudagent.tails.base.BaseTailsServer method), 326`
`upload_tails_file()` `(aries_cloudagent.tails.indy_tails_server.IndyTailsServer method), 327`
`upload_tails_file()` `(in module aries_cloudagent.revocation.routes), 320`
`use_existing_connection` `(aries_cloudagent.protocols.out_of_band.v1_0.routes attribute), 261`
`use_public_did` `(aries_cloudagent.protocols.out_of_band.v1_0.routes attribute), 260`
`USER` `(aries_cloudagent.ledger.base.Role attribute), 66`
`UUIDFour` `(class in aries_cloudagent.messaging.valid), 117`
V
`V10CredentialCreateSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 215`
`V10CredentialExchange` `(class in aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange), 209`
`V10CredentialExchange.Meta` `(class in aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange), 211`
`V10CredentialExchangeListQueryStringSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 216`
`V10CredentialExchangeListResultSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 216`
`V10CredentialExchangeSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange), 211`
`V10CredentialExchangeSchema.Meta` `(class in aries_cloudagent.protocols.issue_credential.v1_0.models.credential_exchange), 211`
`V10CredentialIssueRequestSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 216`
`V10CredentialOfferRequestSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 216`
`V10CredentialProblemReportRequestSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 216`
`V10CredentialProposalRequestMandSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 216`
`V10CredentialProposalRequestOptSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 217`
`V10CredentialProposalRequestSchemaBase` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 217`
`V10CredentialStoreRequestSchema` `(class in aries_cloudagent.protocols.issue_credential.v1_0.routes), 217`
`V10PresentationCreateRequestRequestSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 280`
`V10PresentationReceiveQueryStringSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange), 271`
`V10PresentationCreateRequestSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 273`
`V10PresentationExchange.Meta` `(class in aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange), 273`
`V10PresentationExchangeListQueryStringSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 280`
`V10PresentationExchangeListSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 280`
`V10PresentationExchangeSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange), 273`
`V10PresentationExchangeSchema.Meta` `(class in aries_cloudagent.protocols.present_proof.v1_0.models.presentation_exchange), 273`
`V10PresentationProblemReportRequestSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 280`
`V10PresentationProposalRequestSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 280`
`V10PresentationRequestSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 281`
`V10PresentationSendRequestRequestSchema` `(class in aries_cloudagent.protocols.present_proof.v1_0.routes), 281`
`V20CredAck` `(class in aries_cloudagent.protocols.issue_credential.v2_0.messages.credential_exchange), 224`
`V20CredAck.Meta` `(class in aries_cloudagent.protocols.issue_credential.v2_0.messages.credential_exchange), 225`
`V20CredAckHandler` `(class in aries_cloudagent.protocols.issue_credential.v2_0.handlers.credential_exchange), 221`
`V20CredAckSchema` `(class in aries_cloudagent.protocols.issue_credential.v2_0.messages.credential_exchange), 225`
`V20CredAckSchema.Meta` `(class in aries_cloudagent.protocols.issue_credential.v2_0.messages.credential_exchange), 225`
`V20CredAttrSpec` `(class in aries_cloudagent.protocols.issue_credential.v2_0.messages.credential_exchange), 225`

<code>aries_cloudagent.protocols.issue_credential.v2_0.messages</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
222	242
<code>V20CredAttrSpec.Meta</code> (class in <code>V20CredExRecordSchema</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.messages</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex</code>
223	237
<code>V20CredAttrSpecSchema</code> (class in <code>V20CredExRecordSchema.Meta</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.messages</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex</code>
223	237
<code>V20CredAttrSpecSchema.Meta</code> (class in <code>V20CredFilter</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.messages</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
223	242
<code>V20CredCreateSchema</code> (class in <code>V20CredFilterDIF</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
241	242
<code>V20CredExIdMatchInfoSchema</code> (class in <code>V20CredFilterIndy</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
241	242
<code>V20CredExRecord</code> (class in <code>V20CredFormat</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
235	225
<code>V20CredExRecord.Meta</code> (class in <code>V20CredFormat.Format</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.cred_ex</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
237	226
<code>V20CredExRecordDetailSchema</code> (class in <code>V20CredFormat.Meta</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
241	226
<code>V20CredExRecordDIF</code> (class in <code>V20CredFormatSchema</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_dif</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
232	226
<code>V20CredExRecordDIF.Meta</code> (class in <code>V20CredFormatSchema.Meta</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_dif</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
233	226
<code>V20CredExRecordDIFSchema</code> (class in <code>V20CredIdMatchInfoSchema</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_dif</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
233	243
<code>V20CredExRecordDIFSchema.Meta</code> (class in <code>V20CredIssue</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_dif</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
233	226
<code>V20CredExRecordIndy</code> (class in <code>V20CredIssue.Meta</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_indy</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
233	227
<code>V20CredExRecordIndy.Meta</code> (class in <code>V20CredIssueHandler</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_indy</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_ex</code>
234	221
<code>V20CredExRecordIndySchema</code> (class in <code>V20CredIssueProblemReportRequestSchema</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_indy</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
234	243
<code>V20CredExRecordIndySchema.Meta</code> (class in <code>V20CredIssueRequestSchema</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.models.detail_indy</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>
234	243
<code>V20CredExRecordListQueryStringSchema</code> <code>V20CredIssueSchema</code> (class in <code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_ex</code>
242	227
<code>V20CredExRecordListResultSchema</code> (class in <code>V20CredIssueSchema.Meta</code> (class in	

<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue</code>	Schema	(class in	227
<code>V20CredManager</code>	(class in	230	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_issue.ProposalSchema.Meta</code>	(class in	238	
<code>V20CredManagerError</code>	241	230	
<code>V20CredOffer</code>	(class in	<code>V20CredRequest</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	228	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	231
<code>V20CredOffer.Meta</code>	(class in	<code>V20CredRequest.Meta</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	228	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	231
<code>V20CredOfferHandler</code>	(class in	<code>V20CredRequestHandler</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_offer_handler</code>	221	<code>aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_offer_handler</code>	222
<code>V20CredOfferRequestSchema</code>	(class in	<code>V20CredRequestSchema</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	243	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	232
<code>V20CredOfferSchema</code>	(class in	<code>V20CredRequestSchema.Meta</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	229	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	232
<code>V20CredOfferSchema.Meta</code>	(class in	<code>V20CredStoreRequestSchema</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	229	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	243
<code>V20CredPreview</code>	(class in	<code>V20IssueCredentialModuleResponseSchema</code>	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_preview</code>	223	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	244
<code>V20CredPreview.Meta</code>	(class in	<code>V20IssueCredSchemaCore</code> (class in	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_preview</code>	224	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	244
<code>V20CredPreviewSchema</code>	(class in	<code>valid(aries_cloudagent.messaging.jsonld.routes.VerifyResponseSchema</code>	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_preview</code>	224	<code>validate() (aries_cloudagent.messaging.models.base.BaseModel</code>	
<code>V20CredPreviewSchema.Meta</code>	(class in	<code>method</code>), 98	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_preview</code>	224	<code>aries_cloudagent.messaging.decorators.attach_decorator.Attach</code>	
<code>V20CredProposal</code>	(class in	<code>method</code>), 82	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal</code>	229	<code>aries_cloudagent.multitenant.admin.routes.CreateWalletRequest</code>	
<code>V20CredProposal.Meta</code>	(class in	<code>method</code>), 118	
<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_proposal</code>	230	<code>aries_cloudagent.protocols.connections.v1_0.messages.connection</code>	
<code>V20CredProposalHandler</code>	(class in	<code>method</code>), 140	
<code>aries_cloudagent.protocols.issue_credential.v2_0.handlers.cred_proposal_handler</code>	222	<code>aries_cloudagent.protocols.connections.v1_0.routes.ReceiveInvitation</code>	
<code>V20CredProposalRequestPreviewMandSchema</code>	(class in	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	
<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	243	<code>aries_cloudagent.protocols.issue_credential.v2_0.messages.cred_offer</code>	
<code>V20CredProposalRequestPreviewOptSchema</code>	(class in	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	
<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	243	<code>aries_cloudagent.protocols.issue_credential.v2_0.routes.V20CredProposalRequestSchemaBase</code>	
<code>V20CredProposalRequestSchemaBase</code>	(class in	<code>method</code>), 242	
<code>aries_cloudagent.protocols.issue_credential.v2_0.routes</code>	243	<code>aries_cloudagent.protocols.out_of_band.v1_0.messages.invitation</code>	

[method](#)), 251
[validate_fields\(\)](#) ([aries_cloudagent.protocols.present_proof.v1_0.routes.IndyProofRequestSpecSchema](#) module), 278
[validate_fields\(\)](#) ([aries_cloudagent.protocols.routing.v1_0.models.route_record.RouteRecordSchema](#) module), 296
[validate_fields\(\)](#) ([aries_cloudagent.revocation.routes.CredRevRecordQueryStringSchema](#) module), 316
[validate_filter\(\)](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.V20CredFormat.Format](#) module), 226
[validate_record\(\)](#) ([aries_cloudagent.storage.base](#) module), 322
[validate_seed\(\)](#) ([aries_cloudagent.wallet.crypto](#) module), 364
[validate_single_xor_multi_sig\(\)](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataJWSSchema](#) module), 82
[validate_version\(\)](#) ([aries_cloudagent.core.plugin_registry.PluginRegistry](#) module), 45
[value](#) ([aries_cloudagent.connections.models.diddoc.PublicKey](#) attribute), 24
[value](#) ([aries_cloudagent.connections.models.diddoc.publickey.PublicKey](#) attribute), 28
[value](#) ([aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionSchema](#) attribute), 74
[value](#) ([aries_cloudagent.messaging.models.base_record.BaseRecord](#) attribute), 103
[value](#) ([aries_cloudagent.protocols.issue_credential.v1_0.messages.issue_credential_v1_0.messages.AgentMessage](#) attribute), 202
[value](#) ([aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_cred_v2_0.messages.V20CredAttrSpecSchema](#) attribute), 223
[value](#) ([aries_cloudagent.protocols.present_proof.v1_0.messages.issue_present_proof_v1_0.messages.PresAuthSpecSchema](#) attribute), 265
[ver](#) ([aries_cloudagent.messaging.credential_definitions.routes.CredentialDefinitionSchema](#) attribute), 74
[ver](#) ([aries_cloudagent.messaging.schemas.routes.SchemaSchema](#) attribute), 105
[ver_type](#) ([aries_cloudagent.connections.models.diddoc.LinkedDataKeySpec](#) attribute), 23
[ver_type](#) ([aries_cloudagent.connections.models.diddoc.publickey.LinkedDataKeySpec](#) attribute), 27
[ver_type](#) ([aries_cloudagent.connections.models.diddoc.publickey.PublicKeyType](#) attribute), 28
[ver_type](#) ([aries_cloudagent.connections.models.diddoc.PublicKeyType](#) attribute), 24
[verified](#) ([aries_cloudagent.protocols.present_proof.v1_0.models.present_proof_v1_0.models.VerificationExchangeV1_0.VerificationExchangeSchema](#) attribute), 274
[verify\(\)](#) ([aries_cloudagent.messaging.decorators.attach_decorator.AttachDecoratorDataJWSSchema](#) module), 80
[verify\(\)](#) ([aries_cloudagent.messaging.decorators.sign_decorator.SignDecoratorDataJWSSchema](#) module), 87
[verify\(\)](#) ([aries_cloudagent.messaging.jsonld.routes](#) module), 97
[verify_credential\(\)](#) ([aries_cloudagent.messaging.jsonld.credential](#) module), 96
[verify_diddoc\(\)](#) ([aries_cloudagent.protocols.didexchange.v1_0.manager.PresentProofV1_0Manager](#) module), 185
[verify_jws_header\(\)](#) ([aries_cloudagent.messaging.jsonld.credential](#) module), 96
[verify_message\(\)](#) ([aries_cloudagent.wallet.base.BaseWallet](#) module), 360
[verify_message\(\)](#) ([aries_cloudagent.wallet.in_memory.InMemoryWallet](#) module), 368
[verify_message\(\)](#) ([aries_cloudagent.wallet.indy.IndySdkWallet](#) module), 371
[verify_presentation\(\)](#) ([aries_cloudagent.indy.verifier.IndySdkVerifier](#) module), 57
[verify_presentation\(\)](#) ([aries_cloudagent.indy.verifier.IndyVerifier](#) module), 63
[verify_presentation\(\)](#) ([aries_cloudagent.protocols.present_proof.v1_0.manager.PresentProofV1_0Manager](#) module), 277
[routes.CredentialDefinitionSchema](#) ([aries_cloudagent.messaging.agent_message.AgentMessage](#) module), 108
[verify_signed_field\(\)](#) ([aries_cloudagent.messaging.agent_message.AgentMessage](#) module), 108
[aries_cloudagent.protocols.issue_credential.v2_0.messages.issue_cred_v2_0.messages.V20CredAttrSpecSchema](#) ([aries_cloudagent.wallet.crypto](#) module), 364
[aries_cloudagent.messaging.jsonld.routes](#) ([aries_cloudagent.messaging.jsonld.routes](#) module), 97
[aries_cloudagent.ledger.routes.RegisterLedgerNymQueryQueryStringSchema](#) ([aries_cloudagent.messaging.jsonld.routes](#) module), 97
[aries_cloudagent.messaging.jsonld.routes](#) ([aries_cloudagent.messaging.jsonld.routes](#) module), 97
[aries_cloudagent.wallet.base.DIDInfo](#) ([aries_cloudagent.wallet.base.DIDInfo](#) attribute), 360
[aries_cloudagent.protocols.present_proof.v1_0.models.present_proof_v1_0.models.VerificationExchangeV1_0.VerificationExchangeSchema](#) ([aries_cloudagent.wallet.base.DIDInfo](#) attribute), 361
[aries_cloudagent.wallet.routes.DIDListQueryQueryStringSchema](#) ([aries_cloudagent.wallet.routes.DIDListQueryQueryStringSchema](#) attribute), 372
[aries_cloudagent.wallet.routes.DIDSchema](#) ([aries_cloudagent.wallet.routes.DIDSchema](#) attribute), 372

tribute), 372

version (aries_cloudagent.ledger.routes.AMLRecordSchema attribute), 71

version (aries_cloudagent.ledger.routes.TAAAcceptSchema attribute), 71

version (aries_cloudagent.ledger.routes.TAARecordSchema attribute), 72

version (aries_cloudagent.messaging.schemas.routes.Schema attribute), 105

version (aries_cloudagent.protocols.present_proof.v1_0.models.route_record attribute), 279

wallet_get () (in module aries_cloudagent.multitenant.admin.routes), 120

wallet_get_did_endpoint () (in module aries_cloudagent.wallet.routes), 373

wallet_get_public_did () (in module aries_cloudagent.wallet.routes), 373

wallet_id (aries_cloudagent.multitenant.admin.routes.WalletIdMatchIn attribute), 119

wallet_id (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute), 357

wallet_id (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute), 357

wallet_key (aries_cloudagent.multitenant.admin.routes.CreateWalletRe attribute), 118

wallet_key (aries_cloudagent.multitenant.admin.routes.CreateWalletTo attribute), 118

wallet_key (aries_cloudagent.multitenant.admin.routes.RemoveWalletR attribute), 119

wallet_key (aries_cloudagent.wallet.models.wallet_record.WalletReco attribute), 357

wallet_name (aries_cloudagent.multitenant.admin.routes.CreateWalletK attribute), 118

wallet_name (aries_cloudagent.multitenant.admin.routes.WalletListQue attribute), 119

wallet_name (aries_cloudagent.wallet.models.wallet_record.WalletReco attribute), 357

wallet_only (aries_cloudagent.wallet.did_posture.DIDPosture attribute), 364

wallet_remove () (in module aries_cloudagent.multitenant.admin.routes), 120

wallet_rotate_did_keypair () (in module aries_cloudagent.wallet.routes), 373

wallet_set_did_endpoint () (in module aries_cloudagent.wallet.routes), 373

wallet_set_public_did () (in module aries_cloudagent.wallet.routes), 373

wallet_type (aries_cloudagent.multitenant.admin.routes.CreateWalletR attribute), 118

wallet_type (aries_cloudagent.wallet.models.wallet_record.WalletReco attribute), 357

wallet_update () (in module aries_cloudagent.multitenant.admin.routes), 120

wallet_webhook_urls (aries_cloudagent.multitenant.admin.routes.CreateWalletRequest attribute), 118

wallet_webhook_urls (aries_cloudagent.multitenant.admin.routes.UpdateWalletRequest attribute), 119

wallet_webhook_urls (aries_cloudagent.wallet.models.wallet_record.WalletRecord attribute), 357

wallet_webhook_urls

Y

`yes` (*aries_cloudagent.protocols.present_proof.v1_0.util.predicate.Relation*
attribute), [275](#)