Data Stewardship Wizard SDK

Jakub Drahoš

May 07, 2021

CONTENTS:

This projects aims at providing unified and easy-to-use Python library for communicating with the Data Stewardship Wizard API. For more info about the DSW project itself, see official webpage or the API documentation.

CHAPTER

ONE

INSTALLATION

You can install this library via PyPI:

```
pip install dsw-sdk
```

CHAPTER

TWO

QUICKSTART

The only mandatory step need in order to get going is to initialize the whole SDK and tell it, where is the DSW API located and how to connect to it:

```
dsw_sdk = DataStewardshipWizardSDK(
   api_url='http://localhost:3000',
   email='albert.einstein@example.com',
   password='password',
)
```

Now you are ready to go.

Note: Note that this is only *illustrative example* and we encourage you **not** to store secrets like passwords in the source code. There are better mechanisms (env variables) introduced later in this documentation.

CHAPTER

THREE

BASIC USAGE

Most actions should be done via the high-level interfaces provided on an instance of the *DataStewardshipWizardSDK* class. These interfaces operate with subclasses of *Model* class (e.g. *User*) – these are the DSW data entities. Basically they are just data classes with bunch of attributes and methods for saving the entity (*save()*) on the server and deleting it (*delete()*).

```
user = dsw_sdk.users.create_user(
   first_name='John',
   last_name='Doe',
   email='john.doe@example.com',
)
user.password = os.getenv('SECRET_PASSWORD')
user.save()

...

user.delete()
```

For more advanced usage, see the next sections.

3.1 Advanced usage

3.1.1 Configuration

There are 3 ways to configure the SDK, taking precedence in the following order:

- arguments passed to the *DataStewardshipWizardSDK* class when creating it's instance
- each config value can be set as an environment variable prefixed with `DSW_SDK_` (e.g. `DSW_SDK_EMAIL` or `DSW_SDK_API_URL` is the same as passing `email` or `api_url` to the SDK constructor)
- config loaded from a YAML file (it's path is passed as `conf_file` keyword argument to `__init__()` method)

You can combine these to achieve whatever configuration you need. Consider following scenario: You are administrator of multiple DSW instances and you need to manage them in an effective way. You write a script using this SDK that leverage all of the above mentioned configuration possibilities:

- Storing the general configuration that is common for all instances in the `dsw_conf.yml` file.
- Password for each admin user is saved in `DSW_SDK_PASSWORD` environment variable. That way you can include all of the source codes and even the configuration files in your version control system without compromising some secrets.

- Each instance can have it's specific configuration passed when initializing the library (e.g. some values that are computed only at runtime).

Example of a file config

Listing 1: dsw_conf.yml

```yaml
dsw_sdk:  # This section is mandatory
    enable_ssl: false
    headers:
        'User-Agent': 'DSW SDK'
    default_timeout:
        - 6
        - 120
```

Taking from the example introduced in the quickstart section:

```
# Make sure that we have the password set in an env variable
assert os.getenv('DSW_SDK_PASSWORD')

dsw_sdk = DataStewardshipWizardSDK(
   api_url='http://localhost:3000',
   email='albert.einstein@example.com',
   file_conf='dsw_conf.yml',
)
```

Dependency injection

You can also pass pre-configured instances of HTTP client, session used with HTTP client and logger to the SDK constructor in order to achieve even more customizations.

```
from dsw_sdk.http_client.interface import HttpClient, HttpResponse
from somewhere.inside.your.code import already_setup_logger


class CustomHttpClient(HttpClient):
    # Implementing all abstract methods of
    # the `HttpClient` interface
    def get(self, path: str, **kwargs) -> HttpResponse:
        ...


# Initializing the HTTP client in your own way
http_client = CustomHttpClient(...)

dsw_sdk = DataStewardshipWizardSDK(
    http_client=http_client,
    logger=already_setup_logger,
)
```

In the case of HTTP client, you can also pass only the class of your custom client. It will then get instantiated with all the other config values as it normally would. This is useful if you don't want to perform the initialization yourself or in cases, when you want to override just one aspect of the default HTTP client shipped with this library.

```
from dsw_sdk.http_client.requests_impl.http_client import SessionHttpClient
```

(continues on next page)

(continued from previous page)

```
class CustomHttpClient(SessionHttpClient):
    def after_request(self, response):
        ...  # Some custom logic here


dsw_sdk = DataStewardshipWizardSDK(
    api_url='http://localhost:3000',
    email='albert.einstein@example.com',
    http_client=CustomHttpClient,
)
```

For a complete list of all possible configuration values, see *configuration values*.

3.1.2 Low-level API

In case the SDK does not yet support a functionality of the DSW API that you would like to use, you can use the low-level interface. Basically it only provides a way to communicate with the DSW API, so you don't have to implement your own HTTP client.

Configuration described in the *Configuration* section still applies.

The interface is available via `api` attribute defined on the *`DataStewardshipWizardSDK`* class.

```
dsw_sdk = DataStewardshipWizardSDK(...)
dsw_sdk.api.get_document_download('some-doc-uuid-1')
```

It provides a function for every combination of endpoint and HTTP method. So for example `GET` method on endpoint `/documents/{docUuid}/download` is equivalent to a *`get_document_download()`* method.

Listing 2: Example of low-level API implementation

```
def post_documents(self, body: Dict[str, Any], **kwargs) -> HttpResponse:
    body = self._camelize_dict_keys(body)
    return self._http_client.post(f'/documents', body=body, **kwargs)

def delete_document(self, doc_uuuid: str, **kwargs) -> HttpResponse:
    return self._http_client.delete(f'/documents/{doc_uuuid}', **kwargs)

def get_document_download(self, doc_uuid: str, **kwargs) -> HttpResponse:
    return self._http_client.get(f'/documents/{doc_uuid}/download', **kwargs)
```

Each method on the interface has same path parameters as the endpoint itself. If endpoint expects query parameters or a body inside the request, the method also takes `query_params` or `body` arguments respectively.

```
# Get 10 documents with 'test' query
docs = dsw_sdk.api.get_documents(query_params={'q': 'test', 'size': 10})

# Create a user
user = dsw_sdk.api.post_users(body={'firstName': 'John', ...})
```

Each method also accepts arbitrary keyword arguments that are passed to the underlying implementation of the HTTP client. Therefore you can customize each request as you will.

```
# Never timeout - waiting indefinitely (argument
# `timeout` is passed to the Requests `get` method)
dsw_sdk.api.get_documents(timeout=None)
```

Also note, that each method's name is slightly modified to reflect whether it's dealing with one or multiple entities. E.g. `get_documents` for retrieving multiple documents vs. `get_document_download` for downloading one document (although both methods use the `documents` endpoint).

Both `query_params` and `body` arguments keys are converted to camelCase, so you can define these in the snake_case.

```
# Both of these are equal
dsw_sdk.api.post_users(body={'firstName': 'John', ...})
dsw_sdk.api.post_users(body={'first_name': 'John', ...})
```

3.1.3 High-level API

High-level interface provides object-oriented way to deal with the DSW data entities. There are currently 6 of these, accessible on instances of the *DataStewardshipWizardSDK*:

- `app_config`
- `documents`
- `packages`
- `questionnaires`
- `templates`
- `users`

See API reference of respective classes for more info and examples on usage:

- *App config API*
- *Document API*
- *Package API*
- *Questionnaire API*
- *Template API*
- *User API*

Models

There is a *Model* class for each entity (except package) which can be also used directly. This is useful if you already have the data of the entity, but not the model, so have to instantiate it yourself:

```
# Somehow you got all the data of a user
user_data = ...
# You must pass also the `DataStewardshipWizardSDK` instance to the model;
# argument `__update_attrs` is used to instantiate the model and to put it
# in the right state
user = User(dsw_sdk, __update_attrs=user_data)
# You can also use the `_update_attrs` method, it's the same
user = User(dsw_sdk)
user._update_attrs(user_data)
```

3.2 Reference

3.2.1 API

Low-level API

class api.**LowLevelAPI**(*http_client*)

Low-level API mirroring 1:1 the Data Stewardship Wizard API. It contains one method for each combination of HTTP method and API endpoint.

If the endpoint accepts query parameters or body, the method accept these as well. Keys in both query params and body are converted to *camelCase*, so you can pass them in *snake_case* if you want.

Note that this class is *generated* by a script, not written by hand.

Parameters **http_client** (HttpClient) – Some instance of the *HttpClient* interface.

post_action_keys(*body*, ***kwargs*)

body: type: None email: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_auth(*id*, *query_params=None*, ***kwargs*)

query_params: clientUrl [optional]: string

Parameters

- **id** (*str*) –
- **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_auth_callback(*id*, *query_params=None*, ***kwargs*)

query_params: clientUrl [optional]: string error [optional]: string code [optional]: string

Parameters

- **id** (*str*) –
- **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_book_reference(*br_short_uuid*, ***kwargs*)

Parameters **br_short_uuid** (*str*) –

Return type *interface.HttpResponse*

get_branches(*query_params=None*, ***kwargs*)

query_params: q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_branches(*body*, ***kwargs*)

body: name: string kmId: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_branch(*b_uuid*, ***kwargs*)

Parameters **b_uuid** (*str*) –

Return type *interface.HttpResponse*

put_branch(*b_uuid*, *body*, ***kwargs*)

body: name: string kmId: string events: array

Parameters

- **b_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_branch(*b_uuid*, ***kwargs*)

Parameters **b_uuid** (*str*) –

Return type *interface.HttpResponse*

delete_caches(***kwargs*)

Return type *interface.HttpResponse*

post_caches_knowledge_model(*body*, ***kwargs*)

body: events: array tagUuids: array

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_configs_app(***kwargs*)

Return type *interface.HttpResponse*

put_configs_app(*body*, ***kwargs*)

body: organization: None authentication: None privacyAndSupport: None dashboard: None lookAndFeel: None registry: None questionnaire: None template: None submission: None

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_configs_bootstrap(***kwargs*)

Return type *interface.HttpResponse*

get_documents(*query_params=None*, ***kwargs*)

query_params: questionnaireUuid [optional]: string q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_documents(*body*, ***kwargs*)

body: name: string questionnaireUuid: None templateId: string formatUuid: None

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_documents_housekeeping(***kwargs*)

Return type *interface.HttpResponse*

delete_document(*doc_uuuid*, ***kwargs*)

Parameters **doc_uuuid** (*str*) –

Return type *interface.HttpResponse*

get_document_download(*doc_uuid*, ***kwargs*)

Parameters **doc_uuid** (*str*) –

Return type *interface.HttpResponse*

get_document_available_submission_services(*doc_uuuid*, ***kwargs*)

Parameters **doc_uuuid** (*str*) –

Return type *interface.HttpResponse*

get_feedbacks(*query_params=None*, ***kwargs*)

query_params: packageId [optional]: string questionUuid [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_feedbacks(*body*, ***kwargs*)

body: questionUuid: None packageId: string title: string content: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_feedbacks_synchronization(***kwargs*)

Return type *interface.HttpResponse*

get_feedback(*f_uuid*, ***kwargs*)

Parameters **f_uuid** (*str*) –

Return type *interface.HttpResponse*

get_(***kwargs*)

Return type *interface.HttpResponse*

post_knowledge_models_preview(*body*, ***kwargs*)

body: events: array tagUuids: array

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_levels (***kwargs*)

Return type *interface.HttpResponse*

get_metrics (***kwargs*)

Return type *interface.HttpResponse*

get_branch_migrations_current (*b_uuid*, ***kwargs*)

Parameters **b_uuid** (*str*) –

Return type *interface.HttpResponse*

post_branch_migrations_current (*b_uuid*, *body*, ***kwargs*)

body: targetPackageId: string targetTagUuids: array

Parameters

- **b_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_branch_migrations_current (*b_uuid*, ***kwargs*)

Parameters **b_uuid** (*str*) –

Return type *interface.HttpResponse*

post_branch_migrations_current_conflict (*b_uuid*, *body*, ***kwargs*)

body: originalEventUuid: None action: None

Parameters

- **b_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

post_questionnaire_migrations (*qtn_uuid*, *body*, ***kwargs*)

body: targetPackageId: string targetTagUuids: array

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_questionnaire_migrations_current (*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

put_questionnaire_migrations_current (*qtn_uuid*, *body*, ***kwargs*)

body: resolvedQuestionUuids: array

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_questionnaire_migrations_current(*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

post_questionnaire_migrations_current_completion(*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

get_packages(*query_params=None*, ***kwargs*)

query_params: organizationId [optional]: string kmId [optional]: string q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_packages(*body*, ***kwargs*)

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_packages(*query_params=None*, ***kwargs*)

query_params: organizationId [optional]: string kmId [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_packages_suggestions(*query_params=None*, ***kwargs*)

query_params: q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_package(*pkg_id*, ***kwargs*)

Parameters **pkg_id** (*str*) –

Return type *interface.HttpResponse*

delete_package(*pkg_id*, ***kwargs*)

Parameters **pkg_id** (*str*) –

Return type *interface.HttpResponse*

post_packages_bundle(***kwargs*)

Return type *interface.HttpResponse*

get_package_bundle(*pkg_id*, ***kwargs*)

Parameters **pkg_id** (*str*) –

Return type *interface.HttpResponse*

post_package_pull(*pkg_id*, ***kwargs*)

Parameters **pkg_id** (*str*) –

Return type *interface.HttpResponse*

get_questionnaires(*query_params=None*, ***kwargs*)

query_params: q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_questionnaires(*query_params=None*, ***kwargs*)

query_params: cloneUuid [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_questionnaire(*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

put_questionnaire(*qtn_uuid*, *body*, ***kwargs*)

body: name: string visibility: None sharing: None permissions: array

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_questionnaire(*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

put_questionnaire_content(*qtn_uuid*, *body*, ***kwargs*)

body: events: array

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_questionnaire_report(*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

get_questionnaire_documents(*qtn_uuid*, *query_params=None*, ***kwargs*)

query_params: q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters

- **qtn_uuid** (*str*) –
- **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_questionnaire_documents_preview(*qtn_uuid*, *query_params=None*, ***kwargs*)

query_params: Authorization [optional]: string

Parameters

- **qtn_uuid** (*str*) –
- **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_questionnaire_versions(*qtn_uuid*, ***kwargs*)

Parameters **qtn_uuid** (*str*) –

Return type *interface.HttpResponse*

post_questionnaire_versions(*qtn_uuid*, *body*, ***kwargs*)

body: name: string eventUuid: None

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

put_questionnaire_version(*qtn_uuid*, *v_uuid*, *body*, ***kwargs*)

body: name: string eventUuid: None

Parameters

- **qtn_uuid** (*str*) –
- **v_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_questionnaire_version(*qtn_uuid*, *v_uuid*, ***kwargs*)

Parameters

- **qtn_uuid** (*str*) –
- **v_uuid** (*str*) –

Return type *interface.HttpResponse*

post_questionnaire_revert (*qtn_uuid*, *body*, ***kwargs*)

body: eventUuid: None

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

post_questionnaire_revert_preview (*qtn_uuid*, *body*, ***kwargs*)

body: eventUuid: None

Parameters

- **qtn_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

post_registry_signup (*body*, ***kwargs*)

body: email: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

post_registry_confirmation (*body*, ***kwargs*)

body: organizationId: string hash: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

post_submissions (*body*, ***kwargs*)

body: serviceId: string docUuid: None

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_template_assets (*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

post_template_assets (*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

get_template_asset(*template_id*, *asset_uuid*, ***kwargs*)

Parameters

- **template_id** (*str*) –
- **asset_uuid** (*str*) –

Return type *interface.HttpResponse*

delete_template_asset(*template_id*, *asset_uuid*, ***kwargs*)

Parameters

- **template_id** (*str*) –
- **asset_uuid** (*str*) –

Return type *interface.HttpResponse*

get_template_asset_content(*template_id*, *asset_uuid*, ***kwargs*)

Parameters

- **template_id** (*str*) –
- **asset_uuid** (*str*) –

Return type *interface.HttpResponse*

get_template_files(*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

post_template_files(*template_id*, *body*, ***kwargs*)

body: fileName: string content: string

Parameters

- **template_id** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_template_file(*template_id*, *file_uuid*, ***kwargs*)

Parameters

- **template_id** (*str*) –
- **file_uuid** (*str*) –

Return type *interface.HttpResponse*

put_template_file(*template_id*, *file_uuid*, *body*, ***kwargs*)

body: fileName: string content: string

Parameters

- **template_id** (*str*) –
- **file_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_template_file(*template_id*, *file_uuid*, ***kwargs*)

Parameters

- **template_id** (*str*) –
- **file_uuid** (*str*) –

Return type *interface.HttpResponse*

get_templates(*query_params=None*, ***kwargs*)

query_params: organizationId [optional]: string templateId [optional]: string q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_templates(*body*, ***kwargs*)

body: name: string organizationId: string templateId: string version: string metamodelVersion: integer description: string readme: string license: string allowedPackages: array formats: array

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_templates(*query_params=None*, ***kwargs*)

query_params: organizationId [optional]: string templateId [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_templates_all(*query_params=None*, ***kwargs*)

query_params: organizationId [optional]: string templateId [optional]: string pkgId [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_templates_suggestions(*query_params=None*, ***kwargs*)

query_params: pkgId [optional]: string q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_template(*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

put_template(*template_id*, *body*, ***kwargs*)

body: name: string organizationId: string templateId: string version: string metamodelVersion: integer description: string readme: string license: string allowedPackages: array formats: array

Parameters

- **template_id** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_template(*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

post_templates_bundle(***kwargs*)

Return type *interface.HttpResponse*

get_template_bundle(*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

post_template_pull(*template_id*, ***kwargs*)

Parameters **template_id** (*str*) –

Return type *interface.HttpResponse*

post_tokens(*body*, ***kwargs*)

body: email: string password: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

post_typehints(*body*, ***kwargs*)

body: events: array questionUuid: None q: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_users(*query_params=None*, ***kwargs*)

query_params: q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

post_users(*body*, ***kwargs*)

body: firstName: string lastName: string email: string password: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_users_suggestions(*query_params=None*, ***kwargs*)

query_params: q [optional]: string page [optional]: integer size [optional]: integer sort [optional]: string

Parameters **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

get_users_current(***kwargs*)

Return type *interface.HttpResponse*

put_users_current(*body*, ***kwargs*)

body: firstName: string lastName: string email: string submissionProps: array

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

put_users_current_password(*body*, ***kwargs*)

body: password: string

Parameters **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

get_user(*u_uuid*, ***kwargs*)

Parameters **u_uuid** (*str*) –

Return type *interface.HttpResponse*

put_user(*u_uuid*, *body*, ***kwargs*)

body: firstName: string lastName: string email: string role: string active: boolean

Parameters

- **u_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

delete_user(*u_uuid*, ***kwargs*)

Parameters **u_uuid** (*str*) –

Return type *interface.HttpResponse*

put_user_password(*u_uuid*, *body*, *query_params=None*, ***kwargs*)

body: password: string

query_params: hash [optional]: string

Parameters

- **u_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –
- **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

put_user_state(*u_uuid*, *body*, *query_params=None*, ***kwargs*)

body: active: boolean

query_params: hash [optional]: string

Parameters

- **u_uuid** (*str*) –
- **body** (*Dict[str, Any]*) –
- **query_params** (*Optional[Dict[str, Any]]*) –

Return type *interface.HttpResponse*

put_branch_version(*b_uuid*, *version*, *body*, ***kwargs*)

body: description: string readme: string license: string

Parameters

- **b_uuid** (*str*) –
- **version** (*str*) –
- **body** (*Dict[str, Any]*) –

Return type *interface.HttpResponse*

App config API

App config high-level API.

class app_config_api.**AppConfigAPI**(*sdk*)

API for the App config "entity".

There is just one GET and one PUT method on the DSW API, so we can only return the app config values, updated it and then save it on the remote server once again.

model_class

alias of dsw_sdk.high_level_api.models.app_config.AppConfig

get_config()

Get app config from the server.

Returns object containing DSW app config

Return type dsw_sdk.high_level_api.models.app_config.AppConfig

Document API

Documents high-level API.

class document_api.**DocumentAPI**(*sdk*)

API for the Document entities.

For now, there are just two methods for getting one or many documents (as others were not yet required, but might be implemented in the future).

Example usage:

```
api = DocumentAPI(...)

# Get one document by UUID
doc = api.get_document('some-uuid-1234')

# Get all documents for a given questionnaire
docs = api.get_documents(questionnaire_uuid='q-uuid-5')

# Get page number 1 (each page having 10 documents) of documents
# containing the "foo" string, sorted by the UUID attribute in the
# ascending order
docs = api.get_documents(q='foo', page=1, size=10, sort='uuid,asc')
```

model_class

alias of dsw_sdk.high_level_api.models.document.Document

get_document(*uuid*)

Retrieves one document, identified by it's UUID.

Parameters **uuid** (*str*) – universally unique identifier of the document

Returns object representing a document

Return type dsw_sdk.high_level_api.models.document.Document

get_documents(*questionnaire_uuid=None*, ***query_params*)

Retrieves list of documents. Possibly identified by the questionnaire to which they belong to.

Parameters

- **questionnaire_uuid** (*Optional[str]*) – UUID of the questionnaire whose documents we want to fetch
- **query_params** – optional query params: q, size, page and sort

Returns list of objects, each representing a document

Return type List[dsw_sdk.high_level_api.models.document.Document]

Package API

Packages high-level API.

class package_api.**PackageAPI**(*sdk*)

API for the Package entities.

This class provides a method for pulling new versions of packages (knowledge models). Other methods might be added in the future.

Example usage:

```
api = PackageAPI(...)

# Pull specified packages from the Data Stewardship Registry
api.pull_packages(['dsw:package1:2.2.0', 'dsw:package2:1.0.0'])
```

pull_packages(*ids*)

Pulls given packages from the Data Stewardship Registry, so they become available in the DSW instance.

Parameters **ids** (*List[str]*) – IDs of the packages you want to pull form the registry

update_packages(*ids=None*)

Parameters **ids** (*Optional[List[str]]*) –

model_class: Type[dsw_sdk.high_level_api.models.model.Model]

Questionnaire API

Questionnaires high-level API.

class questionnaire_api.**QuestionnaireAPI**(*sdk*)

API for the Questionnaire entities.

For now, there are just two methods for getting one or many questionnaires (as others were not yet required, but might be implemented in the future).

Example usage:

```
api = QuestionnaireAPI(...)

# Get one questionnaire by UUID
q = api.get_questionnaire('some-uuid-1234')

# Get page number 1 (each page having 10 questionnaires) of
# questionnaires containing the "foo" string, sorted by the UUID
# attribute in the ascending order
qs = api.get_questionnaires(q='foo', page=1, size=10, sort='uuid,asc')
```

model_class

alias of dsw_sdk.high_level_api.models.questionnaire.Questionnaire

get_questionnaire(*uuid*)

Retrieves one questionnaire, identified by it's UUID. Also loading all of it's related documents.

Parameters **uuid** (*str*) – universally unique identifier of the questionnaire

Returns object representing a questionnaire

Return type dsw_sdk.high_level_api.models.questionnaire.Questionnaire

get_questionnaires(***query_params*)
Retrieves list of questionnaires. Also loading all related documents.

Parameters **query_params** – optional query params `q`, `size`, `page` and `sort`

Returns list of objects, each representing a questionnaire

Return type List[dsw_sdk.high_level_api.models.questionnaire.Questionnaire]

Template API

Templates high-level API.

class template_api.**TemplateAPI**(*sdk*)
API for the Template entities.

Supports all of the CRUD operations and pulling/updating templates.

Example usage:

```
api = TemplateAPI(...)

# Get one template by ID
template = api.get_template('dsw:template:1.0.0')

# Get all templates for a given organization
templates = api.get_templates(organization_id='dsw')

# Get page number 1 (each page having 10 templates) of templates
# containing the "foo" string, sorted by the UUID attribute in the
# ascending order
templates = api.get_templates(q='foo', page=1, size=10, start='id,asc')

# Create template
template = api.create_template(template_id='new_temp', ...)

# Delete template
api.delete_template(template.id)

# Delete all templates for a given organization
api.delete_templates(organization_id='dsw')

# Delete list of templates
api.delete_templates(ids=['dsw:template:1.0.0', 'dsw:temp:2.3.1'])

# Pull specified templates from the Data Stewardship Registry
api.pull_templates(['dsw:template:1.0.0', 'dsw:temp:2.3.1'])

# Update specified templates (so that they have latest version
# available)
api.update_templates(['dsw:template:1.0.0', 'dsw:temp:2.3.1'])

# Update all templates to the latest version available
api.update_templates()
```

model_class
alias of `dsw_sdk.high_level_api.models.templates.template.Template`

get_template(*id_*)
Retrieves one template, identified by it's ID. Also loading all of it's related files and assets.

Parameters

- **id** – template identifier
- **id_** (*str*) –

Returns object representing a template

Return type dsw_sdk.high_level_api.models.templates.template.Template

get_templates(*organization_id=None*, *template_id=None*, ***query_params*)
Retrieves list of templates. Also loading all related files and assets.

Possibly identified by the organization to which they belong to. It's also possible to search based on the `template_id` (note that this is different from template's `id` attribute).

Parameters

- **organization_id** (*Optional[str]*) – ID of the organization whose templates we want to fetch
- **template_id** (*Optional[str]*) – `template_id` attribute of a template (different from `id` attribute)
- **query_params** – optional query params: `q`, `size`, `page` and `sort`

Returns list of objects, each representing a template

Return type List[dsw_sdk.high_level_api.models.templates.template.Template]

create_template(***kwargs*)
Creates a template with given data on the DSW server.

Parameters **kwargs** – all the data needed for the template creation

Returns object representing the new template

Return type dsw_sdk.high_level_api.models.templates.template.Template

delete_template(*id_*)
Deletes a template from the DSW server.

Parameters **id_** (*str*) – ID of the template to delete

delete_templates(*organization_id=None*, *ids=None*)
Deletes multiple templates on the DSW server, identified either by organization or ID.

Parameters

- **organization_id** (*Optional[str]*) – ID of the organization whose templates we want to delete
- **ids** (*Optional[List[str]]*) – IDs of templates to delete

pull_templates(*ids*)
Pulls given templates from the Data Stewardship Registry, so they become available in the DSW instance.

Parameters **ids** (*List[str]*) – IDs of the templates you want to pull form the registry

update_templates(*ids=None*)
Updates specified templates, pulling their latest available version. If no IDs are given, updates *all* templates on the DSW instance.

Parameters **ids** (*Optional[List[str]]*) – optional list of template IDs to update

User API

Users high-level API.

class user_api.**UserAPI**(*sdk*)

API for the User entities.

Supports all of the CRUD operations.

Example usage:

```
api = UserAPI(...)

# Get one user by UUID
user = api.get_user('some-user-uuid')

# Get all users
users = api.get_users()

# Get page number 1 (each page having 10 users) of users
# containing the "foo" string, sorted by the UUID attribute in the
# ascending order
users = api.get_users(q='foo', page=1, size=10, sort='uuid,asc')

# Create user
user = api.create_user(first_name='John', ...)

# Delete user
api.delete_user(user.uuid)

# Delete list of users
api.delete_users(['some-user-uuid', 'another-user-123'])
```

model_class

alias of dsw_sdk.high_level_api.models.user.User

get_user(*uuid*)

Retrieves one user, identified by it's UUID. Also loading all his/her related questionnaires and documents.

Parameters **uuid** (*str*) – user identifier

Returns object representing a user

Return type dsw_sdk.high_level_api.models.user.User

get_users(***query_params*)

Retrieves list of users. Also loading all related questionnaires and documents.

Parameters **query_params** – optional query params: q, size, page and sort

Returns list of objects, each representing a user

Return type List[dsw_sdk.high_level_api.models.user.User]

create_user(***kwargs*)

Creates a user with given data on the DSW server.

Parameters **kwargs** – all the data needed for the user creation

Returns object representing the new user

Return type dsw_sdk.high_level_api.models.user.User

delete_user(*uuid*)
Deletes a user from the DSW server.

Parameters **uuid** (*str*) – UUID of the user to delete

delete_users(*uuids*)
Deletes multiple users on the DSW server, identified by UUIDs.

Parameters **uuids** (*List[str]*) – UUIDs of users to delete

3.2.2 Common

Attributes

Core module for working with attributes. Defines two important classes *AttributesMixin* (attributes container) and *Attribute*. Also defining all different kinds of specific attributes.

When designing this library, there was a need to implement some kind of mechanism, that would allow following:

- Easy definition of data entities and their attributes, including their type, default value, range and whether they are immutable or read-only. This should be done in as declarative and concise form as possible.
- Validation and possible conversion of values when assigning to these attributes.
- Possibility to initialize the whole entity either in one step (passing all the values to the `__init__` method) or gradually by assigning one attribute at a time.
- There must be a way to initialize the entity without any validation.

Because of these requirements, I decided to implement the attribute as a descriptor. It allows for concise and clean definition and we can specify any custom logic we want when assigning a value, while keeping simple dot notation (`obj.attribute = 1`).

AttributesMixin is just a container for these descriptors, containing all the methods to ease the usage of it's subclasses.

exception attributes.**AttributeNotSetError**(*name*)
Raised when accessing an attribute that was not yet set.

Parameters **name** (*str*) –

msg = 'You must set the `{}` parameter'

exception attributes.**AttributesNotSetError**(*params*)
Raised if there are some attributes that were not yet set when validating an *AttributesMixin* instance.

Parameters **params** (*List[str]*) –

msg = 'You must set following parameters: {}'

exception attributes.**ReadOnlyAccessError**(*name*)
Raised when assigning value to a read-only attribute.

Parameters **name** (*str*) –

msg = 'Attribute `{}` is read only'

exception attributes.**ModifyImmutableError**(*name*)
Raised when assigning value to an immutable attribute that already has some value set.

Parameters **name** (*str*) –

msg = 'Attribute `{}` is immutable - once set, it cannot be changed.'

exception attributes.**InvalidValueError**(*value*, *name*, *type_*)
Raised if validation failed when assigning to an attribute.

Parameters

- **value** (*Any*) –
- **name** (*str*) –
- **type_** (Type) –

msg = 'Invalid value `{}` for attribute `{}` of type `{}`'

exception attributes.**NotInChoicesError**(*value*, *name*, *choices*)
Raised if value assigned to an attribute is not in specified range of values.

Parameters

- **value** (*Any*) –
- **name** (*str*) –
- **choices** (*Sequence[Any]*) –

msg = 'Invalid value `{}` for attribute `{}`, expected one of `{}`'

class attributes.**AttributesMixin**(***kwargs*)
Container for *Attribute* instances. This class should not be used directly, instead it should be subclassed and have some attributes defined.

Note that if you try to retrieve an attribute that was not yet set and it does not have any default value, an exception will be raised.

Example usage:

```
>>> class Foo(AttributesMixin):
...     some_attr = BoolAttribute(immutable=True)
...     bar = IntegerAttribute(choices=(1, 2))

# Passing attribute values when creating object
>>> foo = Foo(some_attr=False)
>>> foo.some_attr
False

>>> foo.some_attr = True
Traceback (most recent call last):
...
dsw_sdk.common.attributes.ModifyImmutableError: ...

>>> foo.bar
Traceback (most recent call last):
...
dsw_sdk.common.attributes.AttributeNotSetError: ...

>>> foo.bar = 1
>>> foo.bar = 'two'
Traceback (most recent call last):
...
dsw_sdk.common.attributes.InvalidValueError: ...

>>> foo.bar = 3
Traceback (most recent call last):
...
```

(continues on next page)

(continued from previous page)

```
dsw_sdk.common.attributes.NotInChoicesError: ...

# Use `_update_attrs` to skip validations
of `read_only` and `immutable` flags
>>> foo._update_attrs(some_attr=True)
>>> foo.some_attr
True
```

Parameters **kwargs** – All the attributes you want to set on this instance. Note that the attribute must be defined on the class, otherwise it won't get assigned. Attributes passed as a dict in `_update_attrs` keyword argument skip validation of *read_only* and *immutable* flags.

property attr_names
List of all attributes names defined on this class and all of it's superclasses.

Returns names of all attributes

attrs()
Collects all attribute values set on this instance, including default values of attributes and returns them as `dict` (keys are the attributes names).

Returns `dict` with attribute names and corresponding values

Return type Dict[str, Any]

validate()
Validates if all attributes that are needed (i.e. they don't have the `nullable` flag set) are set. If not, raises an exception.

Raises *AttributesNotSetError* if there are some attributes needed to be set

to_json()
Converts the whole instance (it's attributes) to JSON representation. Useful for serializing the instance's state.

Returns dict representation of all the attributes and their values

Return type Dict[str, Any]

class attributes.**Attribute**(*type_*, ***kwargs*)
Class representing one attribute on the *AttributesMixin* classes.

It's defined as a data descriptor, responsible for validating values when the assignment takes place.

For example usage, see *AttributesMixin*, as these two can't be really used separately.

Parameters **type_** (`Type`) – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

to_json(*value*)
Returns JSON representation of given `value` for the `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns JSON representation of the `value`

Return type Any

value_repr(*value*)

Returns string representation of given `value` for the `self._type` data type.

Parameters **value** (*Any*) – value to be represented as a string

Returns string representing `value`

Return type str

class attributes.**StringAttribute**(***kwargs*)

Parameters **type_** – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**IntegerAttribute**(***kwargs*)

Parameters **type_** – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**FloatAttribute**(***kwargs*)

Parameters **type_** – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**BoolAttribute**(***kwargs*)

Parameters **type_** – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**ListAttribute**(*type_*, ***kwargs*)

Parameters **type_** (`Type`) – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**DictAttribute**(*key_type*, *value_type*, ***kwargs*)

Parameters

- **type_** – type of the attribute
- **key_type** (`Type`) –
- **value_type** (`Type`) –

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**DateTimeAttribute**(***kwargs*)

Parameters **type_** – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**ObjectAttribute**(*type_*, ***kwargs*)

Parameters **type_** (*TypingType[AttributesMixin]*) – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class attributes.**Alias**(*aliased_attr*)

Parameters **aliased_attr** (*str*) –

Snapshot

Module containing all classes and functions responsible for dealing with snapshots.

class snapshot.**Snapshot**(*json_repr*)

Snapshot of an object's state in a particular point of time.

For now, it's basically just a dict.

Parameters **json_repr** (*Dict[str, Any]*) –

items()

Mimics the built-in `dict.items()` method.

Returns Exactly the same result as a `dict.items()` would.

Return type ItemsView[str, Any]

class snapshot.**SnapshotDiff**

Class representing the differences between two snapshots (instances of *Snapshot* class).

Contains 3 categories:

- what was modified (same keys, different values)
- what was added (new keys)
- what was deleted (old keys not present anymore)

snapshot.**make_snapshot**(*obj*)

Creates a snapshot from `AttributesMixin` instance.

Parameters **obj** (*dsw_sdk.common.attributes.AttributesMixin*) – instance of class `AttributesMixin`

Returns snapshot of `obj` state

Return type *snapshot.Snapshot*

snapshot.**snapshots_diff**(*old*, *new*)

Compares two snapshots (assuming the first one is older and the second one is newer), returning attributes in which they differ.

Parameters

- **old** (snapshot.Snapshot) – former state of the object

- **new** (snapshot.Snapshot) – newer (current) state of the object

Returns attributes that was added, changed or deleted

Return type *snapshot.SnapshotDiff*

Types

Module containing all the data types used with Attribute classes.

class common.types.**Type**

The most general type, parent for all specific types.

validate(*value*)

Validates if value is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises ValueError if validation fails

convert(*value*)

Tries to convert value to self._type data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type Any

to_json(*value*)

Converts the value to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*Any*) – value to be converted

Returns JSON representation of the value

Return type Any

value_repr(*value*)

Returns the string representation of the value for the self._type data type.

Parameters **value** (*Any*) – value to be represented as a string

Returns string representing value for this particular data type

Return type str

class common.types.**AnyType**

validate(*value*)

Validates if value is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises ValueError if validation fails

convert(*value*)

Tries to convert value to self._type data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type Any

class common.types.**NoneType**

class common.types.**BoolType**

class common.types.**StringType**

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type str

value_repr(*value*)
Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*str*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

class common.types.**IntegerType**

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type int

class common.types.**FloatType**

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type float

class common.types.**UnionType**(**of_types*)

Parameters **of_types** (Type) –

validate(*value*)
Validates if `value` is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises `ValueError` if validation fails

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type Any

to_json(*value*)
Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*Any*) – value to be converted

Returns JSON representation of the `value`

Return type Any

value_repr(*value*)
Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*Any*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

class common.types.**TupleType**(**of_types*)

Parameters **of_types** (Type) –

validate(*value*)
Validates if `value` is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises `ValueError` if validation fails

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type tuple

to_json(*value*)
Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*tuple*) – value to be converted

Returns JSON representation of the `value`

Return type list

value_repr(*value*)
Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*tuple*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

class common.types.**ListType**(*of_type=Any*)

Parameters **of_type** (Type) –

validate(*value*)
Validates if `value` is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises `ValueError` if validation fails

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type list

to_json(*value*)
Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*list*) – value to be converted

Returns JSON representation of the `value`

Return type list

value_repr(*value*)
Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*list*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

class common.types.**DictType**(*keys*, *values*)

Parameters

- **keys** (Type) –
- **values** (Type) –

validate(*value*)
Validates if `value` is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises `ValueError` if validation fails

convert(*value*)
Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type Any

to_json(*value*)
Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*dict*) – value to be converted

Returns JSON representation of the `value`

Return type dict

value_repr(*value*)
Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*dict*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

class common.types.**DateTimeType**

to_json(*value*)

Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*datetime.datetime*) – value to be converted

Returns JSON representation of the `value`

Return type str

class common.types.**ObjectType**(*class_*)

Type representing some custom, user-defined class. It is assumed that objects of this class can be fully instantiated with a dict containing all the needed data.

Example:

```
>>> class Foo:
...     def __init__(self, a, b):
...         self.a = a
...         self.b = b
...
...     def __repr__(self):
...         return f'<Foo a={self.a}, b={self.b} />'

>>> type_ = ObjectType(Foo)
>>> type_.convert({'a': 123, 'b': 'bar'})
<Foo a=123, b=bar />
```

Parameters **class_** (*TypingType[T]*) –

convert(*value*)

Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type T

to_json(*value*)

Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*T*) – value to be converted

Returns JSON representation of the `value`

Return type Dict[str, Any]

value_repr(*value*)

Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*T*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

class common.types.**MappingType**(*mapping_key*, *mapping*)

Type representing some kind of mapping. It is used to determine the correct type of the object at runtime, depending on the specified field.

Example:

```
>>> class Foo:
...     def __init__(self, a, type):
...         self.a = a
...         self.type = type
...
...     def __repr__(self):
...         return f'<Foo a={self.a} />'

>>> class Bar:
...     def __init__(self, a, b, c, type):
...         self.abc = f'{a}{b}{c}'
...         self.type = type
...
...     def __repr__(self):
...         return f'<Bar abc={self.abc} />'

>>> type_ = MappingType('type', {
...     'foo': ObjectType(Foo),
...     'bar': ObjectType(Bar),
... })

>>> type_.convert({'type': 'foo', 'a': 123})
<Foo a=123 />

>>> type_.convert({'type': 'bar', 'a': 'a', 'b': 'b', 'c': 42})
<Bar abc=ab42 />

>>> foo = Foo(a=123, type='foo')
>>> type_.validate(foo)

>>> foo = Foo(a=123, type='bar')
>>> type_.validate(foo)
Traceback (most recent call last):
...
ValueError
```

Parameters

- **mapping_key** (*str*) –
- **mapping** (*Dict[str, ObjectType]*) –

convert(*value*)

Tries to convert `value` to `self._type` data type.

Parameters **value** (*Any*) – value to be converted

Returns possibly converted value, but it might also be just the original value

Return type Any

validate(*value*)
Validates if `value` is of correct data type.

Parameters **value** (*Any*) – value to be validated

Raises `ValueError` if validation fails

to_json(*value*)
Converts the `value` to JSON.

Be aware, that the result is not a string, but is instead represented with built-in Python types.

Parameters **value** (*Any*) – value to be converted

Returns JSON representation of the `value`

Return type Dict[str, Any]

value_repr(*value*)
Returns the string representation of the `value` for the `self._type` data type.

Parameters **value** (*Any*) – value to be represented as a string

Returns string representing `value` for this particular data type

Return type str

common.types.**new_type**(*type_*)

Parameters **type_** (common.types.Type) –

Return type Callable[[], *common.types.Type*]

common.types.**NumericType**()

Utils

Utility methods and helpers.

utils.**to_camel_case**(*string*)
Converts a string from snake_case to camelCase.

Parameters **string** (*str*) – string in snake_case notation

Returns *string* converted to camelCase

Return type str

utils.**to_snake_case**(*string*)
Converts a string from camelCase to snake_case.

Parameters **string** (*str*) – string in camelCase notation

Returns *string* converted to snake_case

Return type str

utils.**truncate_long_string**(*string*, *max_len*)
Truncates the given text from the first newline character. If no newline is present in the `string`, truncate text to `max_len` characters. If the text is shorter than `max_len`, does nothing.

Parameters

- **string** (*str*) – text to truncate
- **max_len** (*int*) – maximal length of the truncated string

Returns truncated string, enquoted in double quotes, with the `(truncated)` string appended at the end

Return type str

3.2.3 Config

Module defining configuration of the whole library and implementing the means of collecting the user config.

Configuration for each component is defined in separate class.

class config.**ComponentConfig**(***kwargs*)

Base class for each component config.

Implements all abstract methods of the `collections.abc.Mapping` abstract class, so it can be used in following manner:

```
>>> class Conf(ComponentConfig):
...     some_value = StringAttribute()

>>> conf = Conf()
>>> conf.some_value
Traceback (most recent call last):
...
dsw_sdk.common.attributes.AttributeNotSetError: ...
>>> conf.some_value = 'foo'

# Check if the value is set
>>> 'some_value' in conf
True

# Get the item
>>> conf.some_value
'foo'
>>> conf['some_value']
'foo'

# Get number of configured values
>>> len(conf)
1

# Iterate over the config as dict
>>> for value in conf.values(): pass
>>> for key in conf.keys(): pass
>>> for key, value in conf.items(): pass

# Deconstruct the config to pass it as kwargs
>>> def foo(**kwargs):
...     return kwargs['some_value']

>>> foo(**conf)
'foo'
```

Parameters **kwargs** – All the attributes you want to set on this instance. Note that the attribute must be defined on the class, otherwise it won't get assigned. Attributes passed as a dict in `_update_attrs` keyword argument skip validation of *read_only* and *immutable* flags.

class config.**HttpClientConfig**(***kwargs*)
Config for the HTTP client.

Parameters **kwargs** – All the attributes you want to set on this instance. Note that the attribute must be defined on the class, otherwise it won't get assigned. Attributes passed as a dict in `_update_attrs` keyword argument skip validation of *read_only* and *immutable* flags.

api_url: str

email: str

password: str

enable_ssl: bool

auth_endpoint: str

headers: ***Headers***
Class representing one attribute on the `AttributesMixin` classes.

It's defined as a data descriptor, responsible for validating values when the assignment takes place.

For example usage, see `AttributesMixin`, as these two can't be really used separately.

default_timeout: ***Timeout***
Class representing one attribute on the `AttributesMixin` classes.

It's defined as a data descriptor, responsible for validating values when the assignment takes place.

For example usage, see `AttributesMixin`, as these two can't be really used separately.

class config.**LoggerConfig**(***kwargs*)
Config for the Logger.

Parameters **kwargs** – All the attributes you want to set on this instance. Note that the attribute must be defined on the class, otherwise it won't get assigned. Attributes passed as a dict in `_update_attrs` keyword argument skip validation of *read_only* and *immutable* flags.

logger_name: str

logger_level: str
Class representing one attribute on the `AttributesMixin` classes.

It's defined as a data descriptor, responsible for validating values when the assignment takes place.

For example usage, see `AttributesMixin`, as these two can't be really used separately.

logger_format: str

class config.**Config**(***obj_config*)
This class serves 2 purposes.

It contains all the other "partial" configs. E.g. config objects for HTTP client or logger.

It also collects user-defined configuration from env variables (prefixed by `DSW_SDK`) and from file (YAML containing `dsw_sdk` section) passed in `conf_file` keyword argument. Then it merges these with all other values passed as keyword arguments, in following order (first takes precedence over the others):

1) keyword arguments
2) environment variables
3) file config

Example file config:

```
dsw_sdk:
  enable_ssl: false
  auth_endpoint: '/auth'
  headers:
    'X-CUSTOM-HEADER': 'Custom value'
  default_timeout:
    - 6
    - 120
```

Parameters **obj_config** – arbitrary config values passed as keyword arguments; if path is passed in `conf_file`, it tries to load the config values from a file

`config.`**Timeout**

The central part of internal API.

This represents a generic version of type 'origin' with type arguments 'params'. There are two kind of these aliases: user defined and special. The special ones are wrappers around builtin collections and ABCs in collections.abc. These must have 'name' always set. If 'inst' is False, then the alias can't be instantiated, this is used by e.g. typing.List and typing.Dict.

alias of Union[None, int, float, Tuple[Union[int, float], Union[int, float]]]

`config.`**Headers**

The central part of internal API.

This represents a generic version of type 'origin' with type arguments 'params'. There are two kind of these aliases: user defined and special. The special ones are wrappers around builtin collections and ABCs in collections.abc. These must have 'name' always set. If 'inst' is False, then the alias can't be instantiated, this is used by e.g. typing.List and typing.Dict.

alias of Dict[str, str]

3.2.4 HTTP client

Interface

General interfaces for HTTP communication – HTTP client, HTTP response and exceptions related to the whole process.

class `interface.`**HttpResponse**

Object representing basic HTTP response that is passed across this SDK. Offers only basic methods, but the original response has to be always accessible via the *`orig_response()`* property.

property path

property orig_response

property text

json(***kwargs*)

Return type Dict[str, Any]

exception `interface.`**HttpError**(*status_code*, *response*)

General HTTP error occurring while communicating with the DSW API. Contains HTTP status code, error message and error response from the server.

Parameters

- **status_code** (*int*) –
- **response** (HttpResponse) –

exception interface.**BadRequestError**(*response*)
HTTP error with status code 400 Bad request. Client sent some invalid request to the server.

Parameters **response** (HttpResponse) –

exception interface.**ForbiddenError**(*response*)
HTTP error with status code 403 Forbidden. Client is not authorized to perform such request.

Parameters **response** (HttpResponse) –

exception interface.**NotFoundError**(*response*)
HTTP error with status code 404 Not found. Client requested a resource that was not found on the server.

Parameters **response** (HttpResponse) –

exception interface.**UnexpectedAuthError**
Something unexpected happened when performing the authentication. There may be something wrong with the DSW API or it could be just some temporary malfunction.

class interface.**HttpClient**(**args*, ***kwargs*)
General interface for any HTTP client responsible for communication with the DSW API.

It does not make any assumptions on how the implementation should work or which libraries it should leverage. So if you want implement some client that is not shipped with this SDK (e.g. for asynchronous communication) or you want to use different libraries for HTTP stuff, feel free to subclass this interface and pass an instance of your client to the SDK when the initialization goes on.

head(*path*, ***kwargs*)
Sends a HEAD HTTP request.

Parameters

- **path** (*str*) – path for the request
- **kwargs** – Optional arguments that the request takes

Raises *HttpError* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *UnexpectedAuthError* on unexpected errors while authenticating

Returns a response from the server contained in the *HttpResponse* object.

Return type *interface.HttpResponse*

options(*path*, ***kwargs*)
Sends a OPTIONS HTTP request.

Parameters

- **path** (*str*) – path for the request
- **kwargs** – Optional arguments that the request takes

Raises *HttpError* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *UnexpectedAuthError* on unexpected errors while authenticating

Returns a response from the server contained in the *HttpResponse* object.

Return type *interface.HttpResponse*

get (*path*, ***kwargs*)
Sends a GET HTTP request.

Parameters

- **path** (*str*) – path for the request
- **kwargs** – Optional arguments that the request takes

Raises *HttpError* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *UnexpectedAuthError* on unexpected errors while authenticating

Returns a response from the server contained in the *HttpResponse* object.

Return type *interface.HttpResponse*

post (*path*, *body=None*, ***kwargs*)
Sends a POST HTTP request.

Parameters

- **path** (*str*) – path for the request
- **body** (*Optional[Dict[str, Any]]*) – body of the request
- **kwargs** – Optional arguments that the request takes

Raises *HttpError* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *UnexpectedAuthError* on unexpected errors while authenticating

Returns a response from the server contained in the *HttpResponse* object.

Return type *interface.HttpResponse*

put (*path*, *body=None*, ***kwargs*)
Sends a PUT HTTP request.

Parameters

- **path** (*str*) – path for the request
- **body** (*Optional[Dict[str, Any]]*) – body of the request
- **kwargs** – Optional arguments that the request takes

Raises *HttpError* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *UnexpectedAuthError* on unexpected errors while authenticating

Returns a response from the server contained in the *HttpResponse* object.

Return type *interface.HttpResponse*

delete (*path*, ***kwargs*)
Sends a DELETE HTTP request.

Parameters

- **path** (*str*) – path for the request
- **kwargs** – Optional arguments that the request takes

Raises *HttpError* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *UnexpectedAuthError* on unexpected errors while authenticating

Returns a response from the server contained in the *HttpResponse* object.

Return type *interface.HttpResponse*

Requests implementation

HTTP client

Module containing a synchronous implementation of the *HttpClient* interface via the popular *requests* library.

class http_client.**RequestsHttpResponse**(*response*)

Requests implementation of the *HttpResponse* interface.

Parameters **response** (*requests.Response*) – original `requests.Response` object

property path

property orig_response

property text

json(***kwargs*)

Return type Dict[str, Any]

class http_client.**SessionHttpClient**(*base_url*, *auth_class*, *logger*, ***kwargs*)

HTTP client for easy communication with the Data Stewardship Wizard API. It's a synchronous implementation of a *HttpClient* interface via the `Requests` package.

Uses `requests.Session` in order to re-use HTTP connections and improve overall performance. Also transparently mediates the authentication process.

It is possible to add custom logic with *before_request()* and *after_request()* hooks.

All requests raise *HttpError*, so you don't have to check every response for valid status code. Just assume it's OK and catch the aforementioned exception somewhere. The *UnexpectedAuthError* is raised if some unexpected error occurs during the authentication process. This means the API responded in an unexpected way and something is wrong.

Keyword arguments

- **default_timeout:** This timeout is used as default value when no timeout is specified within a request. You can pass one numeric value (applied for both the connect and read timeouts), tuple of numeric values to specify different connect and read timeouts or `None` to wait forever. Default: `None`.
- **session (*requests.Session*):** If you want, you can configure your own session (instance of a `requests.Session` class) outside of this client and pass it as keyword argument.
- **headers: (*Dict[str, str]*)** Dict of all the HTTP headers you want to send with each requests.
- **enable_ssl (*bool*):** If you set this flag to `True`, every request gets encrypted and the whole communication will be done over SSL/TLS. Default: `True`.

Parameters

- **base_url** (*str*) – URL of the DSW API
- **auth_class** (*AuthBase*) – class responsible for the authentication process
- **logger** (*logging.Logger*) – logger object

before_request(**args*, ***kwargs*)
This method can be overridden by users to provide custom logic before sending the request.

`args` and `kwargs` are the exact arguments that go into the request. This method must return the same, new or modified `args` and `kwargs`.

Returns current, new or modified positional and keyword arguments for the request

Return type Tuple[Tuple[Any, ..], Dict[str, Any]]

after_request(*response*)
This method can be overridden by users to provide custom logic after receiving the response.

`response` is the original response from the server. This method must return the same, new or modified response.

Parameters **response** (`http_client.RequestsHttpResponse`) – original response from the server

Returns current, new or modified response from the server

Return type *http_client.RequestsHttpResponse*

head(*path*, ***kwargs*)
Sends a HEAD request. Returns *`RequestsHttpResponse`* object.

Parameters

- **path** (*str*) – path for the new `requests.Request` object.
- **kwargs** – Optional arguments that the request takes.

Raises *`HttpError`* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *`UnexpectedAuthError`* on unexpected errors while authenticating

Return type *http_client.RequestsHttpResponse*

options(*path*, ***kwargs*)
Sends an OPTIONS request. Returns *`RequestsHttpResponse`* object.

Parameters

- **path** (*str*) – path for the new `requests.Request` object.
- **kwargs** – Optional arguments that the request takes.

Raises *`HttpError`* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *`UnexpectedAuthError`* on unexpected errors while authenticating

Return type *http_client.RequestsHttpResponse*

get(*path*, ***kwargs*)
Sends a GET request. Returns *`RequestsHttpResponse`* object.

Parameters

- **path** (*str*) – path for the new `requests.Request` object.
- **kwargs** – Optional arguments that request takes.

Raises *`HttpError`* on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises *`UnexpectedAuthError`* on unexpected errors while authenticating

Return type *http_client.RequestsHttpResponse*

post(*path*, *body=None*, ***kwargs*)
Sends a POST request. Returns `RequestsHttpResponse` object.

Parameters

- **path** (*str*) – path for the new `requests.Request` object.
- **body** (*Optional[Dict[str, Any]]*) – (optional) json to send in the body of the `requests.Request`.
- **kwargs** – Optional arguments that the request takes.

Raises `HttpError` on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises `UnexpectedAuthError` on unexpected errors while authenticating

Return type *http_client.RequestsHttpResponse*

put(*path*, *body=None*, ***kwargs*)
Sends a PUT request. Returns `RequestsHttpResponse` object.

Parameters

- **path** (*str*) – path for the new `requests.Request` object.
- **body** (*Optional[Dict[str, Any]]*) – (optional) json to send in the body of the `requests.Request`.
- **kwargs** – Optional arguments that the request takes.

Raises `HttpError` on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises `UnexpectedAuthError` on unexpected errors while authenticating

Return type *http_client.RequestsHttpResponse*

delete(*path*, ***kwargs*)
Sends a DELETE request. Returns `RequestsHttpResponse` object.

Parameters

- **path** (*str*) – path for the new `requests.Request` object.
- **kwargs** – Optional arguments that the request takes.

Raises `HttpError` on HTTP status codes between 400 and 599 while authenticating or doing the request itself

Raises `UnexpectedAuthError` on unexpected errors while authenticating

Return type *http_client.RequestsHttpResponse*

Auth

Module with classes procuring the whole authentication process.

class auth.**TokenRetrievalStrategy**(*auth_url*, *credentials*)
Abstract class for strategy on how to retrieve auth token from the server.

Every subclass has to implement the *get_new_token()* method.

Parameters

- **auth_url** (*str*) – URL of the authentication endpoint
- **credentials** (*Dict[str, str]*) – dict containing keys email and password

get_new_token()
Make a POST request to the authentication URL and obtain new auth token.

Raises requests.HTTPError on HTTP status codes between 400 and 599

Raises *UnexpectedAuthError* on missing token in HTTP response body

Returns authentication token to be passed with every request

Return type str

class auth.**BodyTokenRetrievalStrategy**(*auth_url*, *credentials*)
This strategy knows how to retrieve an authentication token contained in the body of the HTTP response from the server.

Parameters

- **auth_url** (*str*) – URL of the authentication endpoint
- **credentials** (*Dict[str, str]*) – dict containing keys email and password

get_new_token()
Make a POST request to the authentication URL and obtain new auth token.

Raises requests.HTTPError on HTTP status codes between 400 and 599

Raises *UnexpectedAuthError* on missing token in HTTP response body

Returns authentication token to be passed with every request

Return type str

class auth.**BearerAuth**(*token*)
Subclass of requests.AuthBase, defining how to authenticate with every request via the Bearer authentication protocol defined in RFC 6750.

Parameters **token** (*str*) – Authentication token obtained from the auth endpoint

class auth.**JWTBearerAuth**(*token_retrieval_strategy*)
Subclass of requests.AuthBase, defining how to authenticate with every request using JSON Web Tokens defined in RFC 7519.

Parameters **token_retrieval_strategy** (TokenRetrievalStrategy) – means of authentication

static token_expiration(*jwt_token*)
Return expiration datetime of given jwt_token.

Parameters **jwt_token** – string representation of a JWT token (encoded)

Raises *UnexpectedAuthError* if jwt_token is not a valid JWT token

Returns datetime of expiration

Return type datetime.datetime

3.2.5 Models

AppConfig model

class app_config.**AppConfig**(*sdk*, ***kwargs*)

Parameters

- **sdk** – instance of the DataStewardshipWizardSDK class
- **kwargs** – arbitrary attributes that can be set on this entity

authentication

created_at

dashboard

look_and_feel

organization

privacy_and_support

questionnaire

registry

submission

template

updated_at

uuid: str

Document model

class document.**Document**(*sdk*, ***kwargs*)

Parameters

- **sdk** – instance of the DataStewardshipWizardSDK class
- **kwargs** – arbitrary attributes that can be set on this entity

created_at

creator_uuid

format_uuid

name

questionnaire

questionnaire_event_uuid

state

template

uuid: str

Model

Module containing the base Model class along with it's utility classes. Also defining two attributes, specific for the Model subclasses only.

exception model.**AlreadyRemovedError**(*model*)

Exception raised when trying to save (via *Model.save()* method) an already deleted model (the method *Model.delete()* was called).

Parameters **model** (Model) – Instance of a model that has been deleted

msg = 'Model `{}` was already removed and cannot be updated anymore.'

class model.**State**

Base class for all other states of the model creating a state machine (state design pattern).

save(*model*)

According to a current state, does whatever is appropriate for saving the entity. Must be implemented in every subclass.

Parameters **model** (model.Model) – model to perform the "save" operation on

Returns instance of the next state

Return type *model.State*

delete(*model*)

According to a current state, does whatever is appropriate for removing the entity. Must be implemented in every subclass.

Parameters **model** (model.Model) – model to perform the "delete" operation on

Returns instance of the next state

Return type *model.State*

class model.**NewState**

State of an entity that does not yet exist on the server, but an instance of a corresponding model is already created.

save(*model*)

Creates new entity on the server.

Parameters **model** (model.Model) – model to perform the "save" operation on

Returns instance of class:*ExistingState* class

Return type *model.State*

delete(*model*)

Does nothing as the entity doesn't really exist.

Parameters **model** (model.Model) – model to perform the "delete" operation on

Returns instance of class:*DeletedState* class

Return type *model.State*

class model.**ExistingState**

State of an entity that exists on the server.

save(*model*)

Updates the entity on the server.

Parameters **model** (model.Model) – model to perform the "save" operation on

Returns self

Return type *model.State*

delete(*model*)

Removes the entity on the server.

Parameters **model** (model.Model) – model to perform the "delete" operation on

Returns instance of class:*DeletedState* class

Return type *model.State*

class model.**DeletedState**

State of an entity that was on the server, but has been deleted and now exists only an instance of a corresponding model.

save(*model*)

This *always* raises an *AlreadyRemovedError* exception as it is invalid operation. Cannot update an entity that does not exist anymore.

Parameters **model** (model.Model) – model to perform the "save" operation on

Raises *AlreadyRemovedError* always

Return type NoReturn

delete(*model*)

Does nothing as the entity is already deleted on the server.

Parameters **model** (model.Model) – model to perform the "delete" operation on

Returns self

Return type *model.State*

class model.**Model**(*sdk*, ***kwargs*)

This is the base class for all the data stewardship wizard data entities. Defines the one attribute that is common for all entities – UUID.

It tracks it's own state and according to this state it can decide, which operation to do when calling the *save()* and *delete()* methods.

For tracking of changes, snapshots (Snapshot class) are used.

It also modifies a behavior of it's parent, AttributesMixin class – when the attribute is not yet set, it does not raise. Instead None value is returned.

If you, for some reason, want to just create a model for an entity that already exists on the server and you have all the required data, do:

```
# 1
>>> model = Model(None, __update_attrs={'uuid': '123'})
>>> model.uuid
'123'

# 2
>>> model = Model(None)
>>> model._update_attrs(uuid='123')
>>> model.uuid
'123'
```

In either case, the model will be set with correct state.

Parameters

- **sdk** – instance of the `DataStewardshipWizardSDK` class
- **kwargs** – arbitrary attributes that can be set on this entity

uuid: str

save()
If there are some changes to save, persist them on the server. It will either create or update the entity, according to it's state.

Raises *AlreadyRemovedError* if the entity was already deleted

delete()
Deletes the entity. Particular actions depend on entity's internal state.

attrs()
Gets all attributes with non-`None` value.

Returns dict with all entity's attributes, excluding the `None` values

Return type Dict[str, Any]

class model.**ModelAttribute**(*type_*, ***kwargs*)
Attribute containing another *Model* instance.

Parameters **type_** (*TypingType[AttributesMixin]*) – type of the attribute

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

class model.**ListOfModelsAttribute**(*model_class*, ***kwargs*)
Attribute containing a list of another *Model* instances.

Parameters

- **type_** – type of the attribute
- **model_class** (`Type[Model]`) –

Keyword arguments

- **default** (*Any*): default value for the attribute
- **nullable** (*bool*): whether `None` should be a valid value
- **read_only** (*bool*): if set to `True`, assigning to this attribute will raise an exception
- **immutable** (*bool*): if set to `True`, it's possible to assign a value to this attribute only once; any other try will raise an exception
- **choices** (*Sequence[Any]*): sequence defining range of possible values

Questionnaire model

class questionnaire.**Questionnaire**(*sdk*, ***kwargs*)

Parameters

- **sdk** – instance of the DataStewardshipWizardSDK class
- **kwargs** – arbitrary attributes that can be set on this entity

created_at

creator_uuid

events

format

format_uuid

knowledge_model

labels

level

name

package

permissions

replies

selected_tag_uuids

sharing

state

template

template_id

updated_at

versions

visibility

documents
Attribute containing a list of another Model instances.

uuid: str

Template model

class templates.template.**Template**(*sdk*, ***kwargs*)

Parameters

- **sdk** – instance of the DataStewardshipWizardSDK class
- **kwargs** – arbitrary attributes that can be set on this entity

allowed_packages: List[dsw_sdk.high_level_api.dto.template.TemplateAllowedPackage]

created_at: datetime.datetime

description: str

formats: List[dsw_sdk.high_level_api.dto.template.TemplateFormat]

id: str

license: str

metamodel_version: int

name: str

organization: Optional[dsw_sdk.high_level_api.dto.common.OrganizationSimple]

organization_id: str

readme: str

recommended_package_id: Optional[str]

registry_link: Optional[str]

remote_latest_version: Optional[str]

state: str

template_id: str

usable_packages: List[dsw_sdk.high_level_api.dto.common.PackageSimpleDTO]

version: str

versions: List[str]

assets: List[dsw_sdk.high_level_api.models.templates.template_asset.TemplateAsset]
Attribute containing a list of another `Model` instances.

files: List[dsw_sdk.high_level_api.models.templates.template_file.TemplateFile]
Attribute containing a list of another `Model` instances.

User model

class user.**User**(*sdk*, ***kwargs*)

Parameters

- **sdk** – instance of the `DataStewardshipWizardSDK` class
- **kwargs** – arbitrary attributes that can be set on this entity

active: bool

affiliation: str

created_at: datetime.datetime

email: str

first_name: str

groups: List[dsw_sdk.high_level_api.dto.user.GroupMembership]

image_url: str

last_name: str

password: str

permissions: List[str]

role: str

sources: List[str]

updated_at: datetime.datetime

questionnaires
Attribute containing a list of another `Model` instances.

3.2.6 SDK

Module containing the main class of the whole library. This is the class that users will be interacting with. It's the main entrypoint for all other actions.

class sdk.**DataStewardshipWizardSDK**(*session=None*, *http_client=None*, *logger=None*, ***kwargs*)

This class provides simple and concise way to communicate with the Data Stewardship Wizard API. It offers both low-level and object-oriented interfaces.

Low-level API reflects exactly 1:1 the whole SDK API. For each endpoint and each HTTP method there's a function on the low-level interface, which is accessible as the *api* attribute. This interface is intended for use cases which are not covered by the high-level API, offering 100% of the DSW's functionality.

High-level object-oriented interface is available for the most common operations. It's possible that new features will be added in the future. This interface is divided by the entities it is concerned with. Right now there are 6 interfaces of this kind, accessible via attributes:

- *app_config*
- documents
- packages
- questionnaires
- templates
- users

each containing it's own set of functionality (but in most cases it's just CRUD operations).

There some dependency injection parameters you can pass in order to alter the default behavior:

- session – you can pre-configure your own session and pass it as the argument; it will be used with the HTTP client
- http_client – either instance or class implementing the *HttpClient* interface. If you pass only the class, it will get instantiated with all the other config values and auth settings. This way you can override some aspects of the default HTTP client, but don't have to initialize it yourself. But you can also pass the ready-to-use instance and all the HTTP client related config will be ignored.
- logger – if you want, you can specify your own logger.

All configuration values can be set in 3 different ways:

- passing as keyword arguments to the `__init__()` method
- setting values as env variables (prefixed with `DSW_SDK_`)
- via config file (in YAML format); path is passed in the `conf_file` keyword argument

Here is a list of all possible configuration values:

Keyword arguments

- **api_url** (*str*): URL address of the DSW API you want to connect to. It must contain valid url scheme (e.g. *https://*) Mandatory if not passing your own `http_client`.
- **email** (*str*): e-mail address of the user on whose behalf you will be acting. Mandatory if not passing your own `http_client`.
- **password** (*str*): password for logging in. Mandatory if not passing your own `http_client`.
- **enable_ssl** (*bool*): flag defining if the HTTP communication should be encrypted or not. Default: `True`.
- **auth_endpoint** (*str*): endpoint on the DSW API that is responsible for the authorization. Default: `'/tokens'`.
- **headers** (*Dict[str, str]*): Dict of default headers to be sent with every HTTP request. Default: `{}`.
- **default_timeout**: This timeout is used as default value when no timeout is specified within a specific request. You can pass one numeric value (applied for both the connect and read timeouts), tuple of numeric values to specify different connect and read timeouts or `None` to wait forever. Default: `(6.05, 27)`:
- **logger_name** (*str*): Name of the default logger. Default: `'dsw_sdk'`.
- **logger_level** (*Union[int, str]*): Logging level of the default logger. You can use both string levels (e.g. `'INFO'`) or integer levels, ideally constants from the `logging` library (e.g. `logging.INFO`). Default: `logging.WARNING`.
- **logger_format** (*str*): String describing the format of the default logger. For more info, consult the official Python docs about `logging` module. Default: `[%(asctime)s] - %(name)s | %(levelname)s | %(message)s`.

Parameters

- **session** (*requests.Session*) –
- **http_client** (*HTTP_CLIENT*) –
- **logger** (*logging.Logger*) –

Example usage:

```
# Basic configuration
sdk = DataStewardshipWizardSDK(
    api_url='http://localhost:3000',
    email='albert.einstein@example.com',
    password='password',
)

# High-level API usage
temps = sdk.templates.get_templates(size=2, q='my templates')
for temp in temps:
    # Each `temp` is a `Template` instance
    print(template.usable_packages[0].version)
    temp.name = 'Modified template name'
    temp.save()
```

Parameters

- **session** (*requests.Session*) – pre-configured session to be used with the HTTP client instead of the default one

- **http_client** (*HTTP_CLIENT*) – instance or class implementing the *HttpClient* interface
- **logger** (*logging.Logger*) – pre-configured logger to be used instead of the default one

sdk.**HTTP_CLIENT**

The central part of internal API.

This represents a generic version of type 'origin' with type arguments 'params'. There are two kind of these aliases: user defined and special. The special ones are wrappers around builtin collections and ABCs in collections.abc. These must have 'name' always set. If 'inst' is False, then the alias can't be instantiated, this is used by e.g. typing.List and typing.Dict.

alias of Union[dsw_sdk.http_client.interface.HttpClient, Type[dsw_sdk.http_client.interface.HttpClient]]

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V