
Tamr Unify Python Client Documentation

Release 0.1

Tamr

Aug 09, 2019

Contents

1 Client	3
2 Datasets	5
3 Projects	7
4 Machine Learning Models	13
5 Operations	15
6 Authentication	17
Index	19

Version: 0.1


```
class tamr_unify_client.Client (auth, host='localhost', protocol='http', port=9100,  
                               base_path='api/versioned/v1')
```

Python Client for Unify API. Each client is specific to a specific origin (protocol, host, port).

Parameters

- **auth** (`requests.auth.AuthBase`) – Unify-compatible Authentication provider. **Recommended:** use one of the classes described in [Authentication](#)
- **host** (`str`) – Host address of remote Unify instance (e.g. `10.0.10.0`). Default: `'localhost'`
- **protocol** (`str`) – Either `'http'` or `'https'`. Default: `'http'`
- **port** (`int`) – Unify instance main port. Default: `9100`
- **base_path** (`str`) – Base API path. Requests made by this client will be relative to this path. Default: `"api/versioned/v1"`

Usage:

```
>>> import tamr_unify_client as api  
>>> from tamr_unify_client.auth import UsernamePasswordAuth  
>>> auth = UsernamePasswordAuth('my username', 'my password')  
>>> local = api.Client(auth) # on http://localhost:9100  
>>> remote = api.Client(auth, protocol='https', host='10.0.10.0') # on https://  
↪/10.0.10.0:9100
```

origin

HTTP origin i.e. `<protocol>://<host>[:<port>]`. For additional information, see [MDN web docs](#).

Type `str`

request (*method, endpoint, **kwargs*)

Sends an authenticated request to the server. The URL for the request will be `"<origin>/<base_path>/<endpoint>"`.

Parameters

- **method** (*str*) – The HTTP method for the request to be sent.
- **endpoint** (*str*) – API endpoint to call (relative to the Base API path for this client).

get (*endpoint*, ***kwargs*)

Calls *request()* with the "GET" method.

post (*endpoint*, ***kwargs*)

Calls *request()* with the "POST" method.

put (*endpoint*, ***kwargs*)

Calls *request()* with the "PUT" method.

delete (*endpoint*, ***kwargs*)

Calls *request()* with the "DELETE" method.

projects

Collection of all projects on this Unify instance.

Returns Collection of all projects.

Return type ProjectCollection

datasets

Collection of all datasets on this Unify instance.

Returns Collection of all datasets.

Return type DatasetCollection

class `tamr_unify_client.models.dataset.collection.DatasetCollection` (*client*,
api_path='datasets')

Collection of *Dataset* s.

Parameters

- **client** (*Client*) – Client for API call delegation.
- **api_path** (*str*) – API path used to access this collection. E.g. "projects/1/inputDatasets". Default: "datasets".

by_resource_id (*resource_id*)

Retrieve a dataset by resource ID.

Parameters **resource_id** (*str*) – The resource ID. E.g. "1"

Returns The specified dataset.

Return type *Dataset*

by_relative_id (*relative_id*)

Retrieve a dataset by relative ID.

Parameters **relative_id** (*str*) – The resource ID. E.g. "datasets/1"

Returns The specified dataset.

Return type *Dataset*

stream ()

Stream datasets in this collection. Implicitly called when iterating over this collection.

Returns Stream of datasets.

Return type Python generator yielding *Dataset*

Usage:

```
>>> for dataset in collection.stream(): # explicit
>>>     do_stuff(dataset)
>>> for dataset in collection: # implicit
>>>     do_stuff(dataset)
```

by_name (*dataset_name*)

Lookup a specific dataset in this collection by exact-match on name.

Parameters **dataset_name** (*str*) – Name of the desired dataset.

Returns Dataset with matching name in this collection.

Return type *Dataset*

Raises **KeyError** – If no dataset with specified name was found.

class tamr_unify_client.models.dataset.resource.**Dataset** (*client, data, alias=None*)

A Unify dataset.

name

Type *str*

description

Type *str*

version

Type *str*

tags

Type *list[str]*

update_records (*records*)

Send a batch of record creations/updates/deletions to this dataset.

Parameters **records** (*list[dict]*) – Each record should be formatted as specified in the [Public Docs for Dataset updates](#).

refresh (***options*)

Brings dataset up-to-date if needed, taking whatever actions are required.

Parameters ****options** – Options passed to underlying *Operation* . See [apply_options\(\)](#) .

relative_id

Type *str*

resource_id

Type *str*

class `tamr_unify_client.models.project.collection.ProjectCollection` (*client*,
api_path='projects')

Collection of *Project* s.

Parameters

- **client** (*Client*) – Client for API call delegation.
- **api_path** (*str*) – API path used to access this collection. Default: "projects".

by_resource_id (*resource_id*)

Retrieve a project by resource ID.

Parameters **resource_id** (*str*) – The resource ID. E.g. "1"

Returns The specified project.

Return type *Project*

by_relative_id (*relative_id*)

Retrieve a project by relative ID.

Parameters **relative_id** (*str*) – The resource ID. E.g. "projects/1"

Returns The specified project.

Return type *Project*

stream ()

Stream projects in this collection. Implicitly called when iterating over this collection.

Returns Stream of projects.

Return type Python generator yielding *Project*

Usage:

```
>>> for project in collection.stream(): # explicit
>>>     do_stuff(project)
>>> for project in collection: # implicit
>>>     do_stuff(project)
```

class tamr_unify_client.models.project.resource.**Project** (*client, data, alias=None*)
A Unify project.

name

Type str

description

Type str

type

One of: "SCHEMA_MAPPING" "SCHEMA_MAPPING_RECOMMENDATIONS"
"CATEGORIZATION" "DEDUP"

Type str

unified_dataset ()

Unified dataset for this project.

Returns Unified dataset for this project.

Return type *Dataset*

as_categorization ()

Convert this project to a *CategorizationProject*

Returns This project.

Return type *CategorizationProject*

Raises **TypeError** – If the *type* of this project is not "CATEGORIZATION"

as_mastering ()

Convert this project to a *MasteringProject*

Returns This project.

Return type *MasteringProject*

Raises **TypeError** – If the *type* of this project is not "DEDUP"

relative_id

Type str

resource_id

Type str

class tamr_unify_client.models.project.categorization.**CategorizationProject** (*client, data, alias=None*)
A Categorization project in Unify.

model ()

Machine learning model for this Categorization project. Learns from verified labels and predicts categorization labels for unlabeled records.

Returns The machine learning model for categorization.

Return type *MachineLearningModel*

as_categorization ()

Convert this project to a *CategorizationProject*

Returns This project.

Return type *CategorizationProject*

Raises **TypeError** – If the *type* of this project is not "CATEGORIZATION"

as_mastering ()

Convert this project to a *MasteringProject*

Returns This project.

Return type *MasteringProject*

Raises **TypeError** – If the *type* of this project is not "DEDUP"

description

Type *str*

name

Type *str*

relative_id

Type *str*

resource_id

Type *str*

type

One of: "SCHEMA_MAPPING" "SCHEMA_MAPPING_RECOMMENDATIONS"
"CATEGORIZATION" "DEDUP"

Type *str*

unified_dataset ()

Unified dataset for this project.

Returns Unified dataset for this project.

Return type *Dataset*

class `tamr_unify_client.models.project.mastering.MasteringProject` (*client, data, alias=None*)

A Mastering project in Unify.

pairs ()

Record pairs generated by Unify’s binning model. Pairs are displayed on the “Pairs” page in the Unify UI.

Call `refresh ()` from this dataset to regenerate pairs according to the latest binning model.

Returns The record pairs represented as a dataset.

Return type *Dataset*

pair_matching_model()

Machine learning model for pair-matching for this Mastering project. Learns from verified labels and predicts categorization labels for unlabeled pairs.

Calling *predict()* from this dataset will produce new (unpublished) clusters. These clusters are displayed on the “Clusters” page in the Unify UI.

Returns The machine learning model for pair-matching.

Return type *MachineLearningModel*

high_impact_pairs()

High-impact pairs as a dataset. Unify labels pairs as “high-impact” if labeling these pairs would help it learn most quickly (i.e. “Active learning”).

High-impact pairs are displayed with a lightning bolt icon on the “Pairs” page in the Unify UI.

Call *refresh()* from this dataset to produce new high-impact pairs according to the latest pair-matching model.

Returns The high-impact pairs represented as a dataset.

Return type *Dataset*

published_clusters()

Published record clusters generated by Unify’s pair-matching model.

Call *refresh()* from this dataset to republish clusters according to the latest clustering.

Returns The published clusters represented as a dataset.

Return type *Dataset*

as_categorization()

Convert this project to a *CategorizationProject*

Returns This project.

Return type *CategorizationProject*

Raises **TypeError** – If the *type* of this project is not "CATEGORIZATION"

as_mastering()

Convert this project to a *MasteringProject*

Returns This project.

Return type *MasteringProject*

Raises **TypeError** – If the *type* of this project is not "DEDUP"

description

Type *str*

name

Type *str*

relative_id

Type *str*

resource_id

Type *str*

type

One of: "SCHEMA_MAPPING" "SCHEMA_MAPPING_RECOMMENDATIONS"
"CATEGORIZATION" "DEDUP"

Type *str*

unified_dataset()

Unified dataset for this project.

Returns Unified dataset for this project.

Return type *Dataset*

Machine Learning Models

class `tamr_unify_client.models.machine_learning_model.MachineLearningModel` (*client*,
data,
alias=None)

A Unify Machine Learning model.

train (***options*)
Learn from verified labels.

Parameters ***options* – Options passed to underlying *Operation* . See *apply_options()* .

predict (***options*)
Suggest labels for unverified records.

Parameters ***options* – Options passed to underlying *Operation* . See *apply_options()* .

relative_id
Type `str`

resource_id
Type `str`

class `tamr_unify_client.models.operation.Operation` (*client, data, alias=None*)

A long-running operation performed by Unify. Operations appear on the “Jobs” page of the Unify UI.

By design, client-side operations represent server-side operations *at a particular point in time* (namely, when the operation was fetched from the server). In other words: Operations *will not* pick up on server-side changes automatically. To get an up-to-date representation, refetch the operation e.g. `op = op.poll()`.

apply_options (*asynchronous=False, **options*)

Applies operation options to this operation.

NOTE: This function **should not** be called directly. Rather, options should be passed in through a higher-level function e.g. `refresh()`.

Synchronous mode: Automatically waits for operation to resolve before returning the operation.

asynchronous mode: Immediately return the 'PENDING' operation. It is up to the user to coordinate this operation with their code via `wait()` and/or `poll()`.

Parameters

- **asynchronous** (*bool*) – Whether or not to run in asynchronous mode. Default: `False`.
- ****options** – When running in synchronous mode, these options are passed to the underlying `wait()` call.

Returns Operation with options applied.

Return type `Operation`

type

Type `str`

description

Type `str`

state

Server-side state of this operation.

Operation state can be unresolved (i.e. `state` is one of: 'PENDING', 'RUNNING'), or resolved (i.e. `state` is one of: 'CANCELED', 'SUCCEEDED', 'FAILED'). Unless opting into asynchronous mode, all exposed operations should be resolved.

Note: you only need to manually pick up server-side changes when opting into asynchronous mode when kicking off this operation.

Usage:

```

>>> op.status # operation is currently 'PENDING'
'PENDING'
>>> op.wait() # continually polls until operation resolves
>>> op.status # incorrect usage; operation object status never changes.
'PENDING'
>>> op = op.poll() # correct usage; use value returned by Operation.poll_
↳or Operation.wait
>>> op.status
'SUCCEEDED'
```

poll()

Poll this operation for server-side updates.

Does not update the calling `Operation` object. Instead, returns a new `Operation`.

Returns Updated representation of this operation.

Return type `Operation`

wait(poll_interval_seconds=3, timeout_seconds=None)

Continuously polls for this operation's server-side state.

Parameters

- **poll_interval_seconds** (*int*) – Time interval (in seconds) between subsequent polls.
- **timeout_seconds** (*int*) – Time (in seconds) to wait for operation to resolve.

Raises `TimeoutError` – If operation takes longer than `timeout_seconds` to resolve.

Returns Resolved operation.

Return type `Operation`

succeeded()

Convenience method for checking if operation was successful.

Returns True if operation's state is 'SUCCEEDED', False otherwise.

Return type `bool`

relative_id

Type `str`

resource_id

Type `str`

class tamr_unify_client.auth.UsernamePasswordAuth(*username, password*)

Provides username/password authentication for Unify. Specifically, sets the *Authorization* HTTP header with Unify's custom *BasicCreds* format.

Parameters

- **username** (*str*) –
- **password** (*str*) –

Usage:

```
>>> from tamr_unify_client.auth import UsernamePasswordAuth
>>> auth = UsernamePasswordAuth('my username', 'my password')
>>> import tamr_unify_client as api
>>> unify = api.Client(auth)
```


A

apply_options() (*tamr_unify_client.models.operation.Operation* *method*), 15
 as_categorization() (*tamr_unify_client.models.project.categorization.CategorizationProject* *method*), 9
 as_categorization() (*tamr_unify_client.models.project.mastering.MasteringProject* *method*), 10
 as_categorization() (*tamr_unify_client.models.project.resource.Project* *method*), 8
 as_mastering() (*tamr_unify_client.models.project.categorization.CategorizationProject* *method*), 9
 as_mastering() (*tamr_unify_client.models.project.mastering.MasteringProject* *method*), 10
 as_mastering() (*tamr_unify_client.models.project.resource.Project* *method*), 8
 DatasetCollection (class in *tamr_unify_client.models.dataset.collection*), 5
 datasets (*tamr_unify_client.Client* attribute), 4
 delete() (*tamr_unify_client.Client* *method*), 4
 description (*tamr_unify_client.models.dataset.resource.Dataset* attribute), 6
 description (*tamr_unify_client.models.operation.Operation* attribute), 15
 description (*tamr_unify_client.models.project.categorization.CategorizationProject* attribute), 9
 description (*tamr_unify_client.models.project.mastering.MasteringProject* attribute), 10
 description (*tamr_unify_client.models.project.resource.Project* attribute), 8
 get_Project() (*tamr_unify_client.Client* *method*), 4

B

by_name() (*tamr_unify_client.models.dataset.collection.DatasetCollection* *method*), 6
 by_relative_id() (*tamr_unify_client.models.dataset.collection.DatasetCollection* *method*), 5
 by_relative_id() (*tamr_unify_client.models.project.collection.ProjectCollection* *method*), 7
 by_resource_id() (*tamr_unify_client.models.dataset.collection.DatasetCollection* *method*), 5
 by_resource_id() (*tamr_unify_client.models.project.collection.ProjectCollection* *method*), 7

C

CategorizationProject (class in *tamr_unify_client.models.project.categorization*), 8
 Client (class in *tamr_unify_client*), 3
 Dataset (class in *tamr_unify_client.models.dataset.resource*), 6
 DatasetCollection (class in *tamr_unify_client.models.dataset.collection*), 5
 MasteringProject (class in *tamr_unify_client.models.project.mastering*), 9
 model() (*tamr_unify_client.models.project.categorization.CategorizationProject* *method*), 8

D

name (*tamr_unify_client.models.dataset.resource.Dataset* attribute), 6
 name (*tamr_unify_client.models.project.categorization.CategorizationProject* attribute), 9
 name (*tamr_unify_client.models.project.mastering.MasteringProject* attribute), 10

name (*tamr_unify_client.models.project.resource.Project attribute*), 8

O

Operation (class in *tamr_unify_client.models.operation*), 15
 origin (*tamr_unify_client.Client attribute*), 3

P

pair_matching_model () (*tamr_unify_client.models.project.mastering.MasteringProject method*), 10
 pairs () (*tamr_unify_client.models.project.mastering.MasteringProject method*), 9
 poll () (*tamr_unify_client.models.operation.Operation method*), 16

post () (*tamr_unify_client.Client method*), 4
 predict () (*tamr_unify_client.models.machine_learning_model.MachineLearningModel method*), 13

Project (class in *tamr_unify_client.models.project.resource*), 8

ProjectCollection (class in *tamr_unify_client.models.project.collection*), 7

projects (*tamr_unify_client.Client attribute*), 4

published_clusters () (*tamr_unify_client.models.project.mastering.MasteringProject method*), 10

put () (*tamr_unify_client.Client method*), 4

R

refresh () (*tamr_unify_client.models.dataset.resource.Dataset method*), 6

relative_id (*tamr_unify_client.models.dataset.resource.Dataset attribute*), 6

relative_id (*tamr_unify_client.models.machine_learning_model.MachineLearningModel attribute*), 13

relative_id (*tamr_unify_client.models.operation.Operation attribute*), 16

relative_id (*tamr_unify_client.models.project.categorization.CategorizationProject attribute*), 9

relative_id (*tamr_unify_client.models.project.mastering.MasteringProject attribute*), 10

relative_id (*tamr_unify_client.models.project.resource.Project attribute*), 8

request () (*tamr_unify_client.Client method*), 3

resource_id (*tamr_unify_client.models.dataset.resource.Dataset attribute*), 6

resource_id (*tamr_unify_client.models.machine_learning_model.MachineLearningModel attribute*), 13

resource_id (*tamr_unify_client.models.operation.Operation attribute*), 16

resource_id (*tamr_unify_client.models.project.categorization.CategorizationProject attribute*), 9

resource_id (*tamr_unify_client.models.project.mastering.MasteringProject attribute*), 10

resource_id (*tamr_unify_client.models.project.resource.Project attribute*), 8

S

state (*tamr_unify_client.models.operation.Operation attribute*), 15

stream () (*tamr_unify_client.models.dataset.collection.DatasetCollection method*), 5

string_pairs () (*tamr_unify_client.models.project.collection.ProjectCollection method*), 7

successful () (*tamr_unify_client.models.operation.Operation method*), 16

T

tags (*tamr_unify_client.models.dataset.resource.Dataset attribute*), 15

train () (*tamr_unify_client.models.machine_learning_model.MachineLearningModel method*), 13

type (*tamr_unify_client.models.operation.Operation attribute*), 15

type (*tamr_unify_client.models.project.categorization.CategorizationProject attribute*), 9

type (*tamr_unify_client.models.project.mastering.MasteringProject attribute*), 10

type (*tamr_unify_client.models.project.resource.Project attribute*), 8

U

unified_dataset () (*tamr_unify_client.models.project.categorization.CategorizationProject method*), 9

unified_dataset () (*tamr_unify_client.models.project.mastering.MasteringProject method*), 11

unified_dataset () (*tamr_unify_client.models.project.resource.Project method*), 8

update_records () (*tamr_unify_client.models.dataset.resource.Dataset method*), 16

UsernamePasswordAuth (class in *tamr_unify_client.auth*), 17

V

version (*tamr_unify_client.models.dataset.resource.Dataset attribute*), 6

W

wait () (*tamr_unify_client.models.operation.Operation method*), 16