
Ckan API client Documentation

Release 0.1a

Samuele Santi

Jun 16, 2017

Contents

1	Clients	3
1.1	Low-level	3
1.2	High-level	5
1.3	Synchronization	8
2	Modules	11
2.1	ckan_api_client.exceptions	11
2.2	ckan_api_client.objects	12
2.3	ckan_api_client.utils	16
3	Testing	19
3.1	Fixtures	19
3.2	Testing utilities	19
4	Indices and tables	23
	Python Module Index	25

Contents:

There are currently three clients for the Ckan API, each one providing a different level of abstraction, and thus can be used for different needs:

- *CkanLowlevelClient* – just a wrapper around the API.
- High-level client: provides more abstraction around the CRUD methods.
- Syncing client: provides facilities for “syncing” a collection of objects into Ckan.

Low-level

This is the client providing the lowest level of abstraction.

```
class ckan_api_client.low_level.CkanLowlevelClient (base_url, api_key=None)
    Ckan low-level client.
```

- Handles authentication and response validation
- Handles request body serialization and response body deserialization
- Raises HTTPError exceptions on failed HTTP requests
- Performs some checks on return values from the API

anonymous

Property, returning a copy of this client, without an api_key set

request (method, path, **kwargs)

Wrapper around `requests.request()`.

Extra functionality provided:

- Add `Authorization` header to requests
- If data is an object, serialize it with json and add the `Content-type: application/json` header.

- If the response didn't contain an "ok" code, raises a `HTTPError` exception.

Parameters

- **method** – HTTP method to be used
- **path** – Path, relative to the Ckan root. For example: `/api/3/action/package_list`
- **headers** – HTTP headers to be added to the request
- **data** – Data to be sent in the request body
- **kwargs** – Extra keyword arguments will be passed directly to the `requests.request()` call.

Raises `ckan_api_client.exceptions.HTTPError` – in case the HTTP request returned a non-ok status code

Returns a requests response object

list_datasets()

Return a list of all dataset ids

iter_datasets()

Generator yielding dataset objects, iterating over the whole database.

get_dataset(dataset_id)

Get a dataset, using API v2

Parameters **dataset_id** – ID of the requested dataset

Returns a dict containing the data as returned from the API

Return type dict

post_dataset(dataset)

POST a dataset, using API v2 (usually for creation)

Parameters **dataset** (*dict*) – a dict containing data to be sent to Ckan. Should not already contain an id

Returns a dict containing the data as returned from the API

Return type dict

put_dataset(dataset)

PUT a dataset, using API v2 (usually for update)

Parameters **dataset** (*dict*) – a dict containing data to be sent to Ckan. Must contain an id, that will be used to build the URL

Returns a dict containing the updated dataset as returned from the API

Return type dict

delete_dataset(dataset_id, ignore_404=True)

DELETE a dataset, using API v2

Parameters

- **dataset_id** – id of the dataset to be deleted
- **ignore_404** (*bool*) – if `True` (the default), will simply ignore http 404 errors from the API


```

list_groups ()
iter_groups ()
get_group (group_id)
post_group (group)
put_group (group)
delete_group (group_id, ignore_404=True)
list_organizations ()
iter_organizations ()
get_organization (id)
post_organization (organization)
put_organization (organization)
    Warning! with api v3 we need to use POST!
delete_organization (id, ignore_404=True)
list_licenses ()

```

High-level

class `ckan_api_client.high_level.CkanHighlevelClient` (*base_url*, *api_key=None*)
 High-level client, handling CRUD of objects.

This class only returns / handles CkanObjects, to make sure we are handling consistent data (they have validators in place)

```
list_datasets ()
```

Returns a list of dataset ids

```
iter_datasets ()
```

Generator, iterating over all the datasets in ckan

```
get_dataset (id, allow_deleted=False)
```

Get a specific dataset, by id

Note: Since the Ckan API use both ids and names as keys, both `get_dataset ()` and `get_dataset_by_name ()` will perform the exact same request in the background.

The difference is only in the high-level handling: the function will check whether the expected id has the correct value, and raise an `HTTPError(404, ..)` otherwise..

Parameters

- **id** (*str*) – the dataset id
- **allow_deleted** – Whether to return even logically deleted objects. If set to `False` (the default) will raise a `HTTPError(404, ..)` if `state != 'active'`

Return type `CkanDataset`

get_dataset_by_name (*name*, *allow_deleted=False*)

Get a specific dataset, by name

Note: See note on *get_dataset ()*

Parameters

- **name** (*str*) – the dataset name
- **allow_deleted** – Whether to return even logically deleted objects. If set to `False` (the default) will raise a `HTTPError(404, ..)` if `state != 'active'`

Return type *CkanDataset*

save_dataset (*dataset*)

If the dataset already has an id, call *update_dataset ()*, otherwise, call *create_dataset ()*.

Returns as returned by the called function.

Return type *CkanDataset*

create_dataset (*dataset*)

Create a dataset

Return type *CkanDataset*

update_dataset (*dataset*)

Update a dataset

Return type *CkanDataset*

delete_dataset (*id*)

Delete a dataset, by id

wipe_dataset (*id*)

Actually delete a dataset, by renaming it first

list_organizations ()

list_organization_names ()

iter_organizations ()

get_organization (*id*, *allow_deleted=False*)

Get organization, by id.

Note: See note on *get_dataset ()*

Parameters

- **id** (*str*) – the organization id
- **allow_deleted** – Whether to return even logically deleted objects. If set to `False` (the default) will raise a `HTTPError(404, ..)` if `state != 'active'`

Return type *CkanOrganization*

get_organization_by_name (*name*, *allow_deleted=False*)

Get organization by name.

Note: See note on `get_dataset()`

Parameters

- **name** (*str*) – the organization name
- **allow_deleted** – Whether to return even logically deleted objects. If set to `False` (the default) will raise a `HTTPError(404, ..)` if `state != 'active'`

Return type `CkanOrganization`

save_organization (*organization*)

create_organization (*organization*)

Create an organization

Return type `CkanOrganization`

update_organization (*organization*)

Return type `CkanOrganization`

delete_organization (*id*)

list_groups ()

list_group_names ()

iter_groups ()

get_group (*id*, *allow_deleted=False*)

Get group, by id.

Note: See note on `get_dataset()`

Parameters

- **id** (*str*) – the group id
- **allow_deleted** – Whether to return even logically deleted objects. If set to `False` (the default) will raise a `HTTPError(404, ..)` if `state != 'active'`

Return type `CkanGroup`

get_group_by_name (*name*, *allow_deleted=False*)

Get group by name.

Note: See note on `get_dataset()`

Parameters

- **name** (*str*) – the group name
- **allow_deleted** – Whether to return even logically deleted objects. If set to `False` (the default) will raise a `HTTPError(404, ..)` if `state != 'active'`

Return type `CkanGroup`

`save_group` (*group*)

`create_group` (*group*)

Return type *CkanGroup*

`update_group` (*group*)

Return type *CkanGroup*

`delete_group` (*id*)

Synchronization

class `ckan_api_client.syncing.SynchronizationClient` (*base_url*, *api_key=None*, ***kw*)

Synchronization client, providing functionality for importing collections of datasets into a Ckan instance.

Synchronization acts as follows:

- Ensure all the required organizations/groups are there; create a map between “source” ids and Ckan ids. Optionally update existing organizations/groups with new details.
- Find all the Ckan datasets matching the `source_name`
- Determine which datasets...
 - ...need to be created
 - ...need to be updated
 - ...need to be deleted
- First, delete datasets to be deleted in order to free up names
- Then, create datasets that need to be created
- Lastly, update datasets using the configured merge strategy (see constructor arguments).

`__init__` (*base_url*, *api_key=None*, ***kw*)

Parameters

- **base_url** – Base URL of the Ckan instance, passed to high-level client
- **api_key** – API key to be used, passed to high-level client
- **organization_merge_strategy** – One of:
 - ‘create’ (default) if the organization doesn’t exist, create it. Otherwise, leave it alone.
 - ‘update’ if the organization doesn’t exist, create it. Otherwise, update with new values.
- **group_merge_strategy** – One of:
 - ‘create’ (default) if the group doesn’t exist, create it. Otherwise, leave it alone.
 - ‘update’ if the group doesn’t exist, create it. Otherwise, update with new values.
- **dataset_preserve_names** – if `True` (the default) will preserve old names of existing datasets
- **dataset_preserve_organization** – if `True` (the default) will preserve old organizations of existing datasets.
- **dataset_group_merge_strategy** –
 - ‘add’ add groups, keep old ones (default)

- 'replace' replace all existing groups
- 'preserve' leave groups alone

sync (*source_name*, *data*)

Synchronize data from a source into Ckan.

- datasets are matched by `_harvest_source`
- groups and organizations are matched by name

Parameters

- **source_name** – String identifying the source of the data. Used to build ids that will be used in further synchronizations.
- **data** – Data to be synchronized. Should be a dict (or dict-like) with top level keys corresponding to the object type, mapping to dictionaries of `{'id': <object>}`.

ckan_api_client.exceptions

Exceptions used all over the place

exception `ckan_api_client.exceptions.HTTPError` (*status_code, message, original=None*)

Bases: `exceptions.Exception`

Exception representing an HTTP response error.

status_code

HTTP status code

message

Informative error message, if available

exception `ckan_api_client.exceptions.BadApiError`

Bases: `exceptions.Exception`

Exception used to mark bad behavior from the API

exception `ckan_api_client.exceptions.BadApiWarning`

Bases: `exceptions.UserWarning`

Warning to mark bad behavior from the API

exception `ckan_api_client.exceptions.OperationFailure`

Bases: `exceptions.Exception`

Something went wrong // failed expectations somewhere..

ckan_api_client.objects

Base objects

Classes to represent / validate Ckan objects.

```
class ckan_api_client.objects.base.BaseField (default=<object object>, is_key=<object object>, required=False)
```

Bases: object

Pseudo-descriptor, accepting field names along with instance, to allow better retrieving data for the instance itself.

Warning: Beware that fields shouldn't carry state of their own, a part from the one used for generic field configuration, as they are shared between instances.

default = None

is_key = False

get (*instance, name*)

Get the value for the field from the main instance, by looking at the first found in:

- the updated value
- the initial value
- the default value

get_default ()

validate (*instance, name, value*)

The validate method should be the (updated) value to be used as the field value, or raise an exception in case it is not acceptable at all.

set_initial (*instance, name, value*)

Set the initial value for a field

set (*instance, name, value*)

Set the modified value for a field

delete (*instance, name*)

Delete the modified value for a field (logically restores the original one)

serialize (*instance, name*)

Returns the "serialized" (json-encodable) version of the object.

is_modified (*instance, name*)

Check whether this field has been modified on the main instance.

is_equivalent (*instance, name, other, ignore_key=True*)

```
class ckan_api_client.objects.base.BaseObject (values=None)
```

Bases: object

Base for the other objects, dispatching get/set/deletes to BaseField instances, if available.

classmethod from_dict (*data*)

set_initial (*values*)

Set initial values for all fields

to_dict ()

serialize ()

Create a serializable representation of the object.

iter_fields ()

Iterate over fields in this objects, yielding (name, field) pairs.

is_equivalent (*other*, *ignore_key=True*)

Equivalency check between objects. Will make sure that values in all the non-key fields match.

Parameters

- **other** – other object to compare
- **ignore_key** – if set to True (the default), it will ignore “key” fields during comparison

is_modified ()

The object is modified if any of its fields reports itself as modified.

Base fields

```
class ckan_api_client.objects.fields.StringField (default=<object object>,
                                                is_key=<object object>,
                                                required=False)
```

Bases: `ckan_api_client.objects.base.BaseField`

default = None

validate (*instance*, *name*, *value*)

```
class ckan_api_client.objects.fields.ListField (default=<object object>, is_key=<object
                                                object>, required=False)
```

Bases: `ckan_api_client.objects.fields.MutableFieldMixin`, `ckan_api_client.objects.base.BaseField`

static default ()

validate (*instance*, *name*, *value*)

```
class ckan_api_client.objects.fields.DictField (default=<object object>, is_key=<object
                                                object>, required=False)
```

Bases: `ckan_api_client.objects.fields.MutableFieldMixin`, `ckan_api_client.objects.base.BaseField`

static default ()

validate (*instance*, *name*, *value*)

```
class ckan_api_client.objects.fields.GroupsField (default=<object object>,
                                                  is_key=<object object>,
                                                  required=False)
```

Bases: `ckan_api_client.objects.fields.SetField`

validate (*instance*, *name*, *value*)

```
class ckan_api_client.objects.fields.ExtrasField (default=<object object>,
                                                  is_key=<object object>,
                                                  required=False)
```

Bases: `ckan_api_client.objects.fields.DictField`

validate (*instance*, *name*, *value*)

is_equivalent (*instance*, *name*, *other*, *ignore_key=True*)

Ckan dataset / resource

```
class ckan_api_client.objects.ckan_dataset.ResourcesField (default=<object object>,
                                                         is_key=<object object>,
                                                         required=False)
```

Bases: *ckan_api_client.objects.fields.ListField*

The ResourcesField should behave pretty much as a list field, but will keep track of changes, and make sure all elements are CkanResources.

validate (*instance, name, value*)

serialize (*instance, name*)

is_equivalent (*instance, name, other, ignore_key=True*)

```
class ckan_api_client.objects.ckan_dataset.CkanDataset (values=None)
```

Bases: *ckan_api_client.objects.base.BaseObject*

id = StringField(default=None, is_key=True, required=False)

name = StringField(default=None, is_key=False, required=False)

title = StringField(default=None, is_key=False, required=False)

author = StringField(default='', is_key=False, required=False)

author_email = StringField(default='', is_key=False, required=False)

license_id = StringField(default='', is_key=False, required=False)

maintainer = StringField(default='', is_key=False, required=False)

maintainer_email = StringField(default='', is_key=False, required=False)

notes = StringField(default='', is_key=False, required=False)

owner_org = StringField(default='', is_key=False, required=False)

private = BoolField(default=False, is_key=False, required=False)

state = StringField(default='active', is_key=False, required=False)

type = StringField(default='dataset', is_key=False, required=False)

url = StringField(default='', is_key=False, required=False)

extras = ExtrasField(default=<function <lambda>>, is_key=False, required=False)

groups = GroupsField(default=<function <lambda>>, is_key=False, required=False)

resources = ResourcesField(default=<function <lambda>>, is_key=False, required=False)

tags = SetField(default=<function <lambda>>, is_key=False, required=False)

```
class ckan_api_client.objects.ckan_dataset.CkanResource (values=None)
```

Bases: *ckan_api_client.objects.base.BaseObject*

id = StringField(default=None, is_key=True, required=False)

description = StringField(default='', is_key=False, required=False)

format = StringField(default='', is_key=False, required=False)

mimetype = StringField(default=None, is_key=False, required=False)

mimetype_inner = StringField(default=None, is_key=False, required=False)

name = StringField(default='', is_key=False, required=False)

```

resource_type = StringField(default='', is_key=False, required=False)
size = StringField(default=None, is_key=False, required=False)
url = StringField(default='', is_key=False, required=False)
url_type = StringField(default=None, is_key=False, required=False)

```

Ckan group

```

class ckan_api_client.objects.ckan_group.CkanGroup (values=None)
  Bases: ckan_api_client.objects.base.BaseObject
  id = StringField(default=None, is_key=True, required=False)
  name = StringField(default=None, is_key=False, required=False)
  title = StringField(default='', is_key=False, required=False)
  approval_status = StringField(default='approved', is_key=False, required=False)
  description = StringField(default='', is_key=False, required=False)
  image_url = StringField(default='', is_key=False, required=False)
  is_organization = BoolField(default=False, is_key=False, required=False)
  state = StringField(default='active', is_key=False, required=False)
  type = StringField(default='group', is_key=False, required=False)
  extras = ExtrasField(default=<function <lambda>>, is_key=False, required=False)
  groups = GroupsField(default=<function <lambda>>, is_key=False, required=False)
  tags = ListField(default=<function <lambda>>, is_key=False, required=False)

```

Ckan organization

```

class ckan_api_client.objects.ckan_organization.CkanOrganization (values=None)
  Bases: ckan_api_client.objects.base.BaseObject
  id = StringField(default=None, is_key=True, required=False)
  name = StringField(default=None, is_key=False, required=False)
  title = StringField(default='', is_key=False, required=False)
  approval_status = StringField(default='approved', is_key=False, required=False)
  description = StringField(default='', is_key=False, required=False)
  image_url = StringField(default='', is_key=False, required=False)
  is_organization = BoolField(default=True, is_key=False, required=False)
  state = StringField(default='active', is_key=False, required=False)
  type = StringField(default='organization', is_key=False, required=False)
  extras = ExtrasField(default=<function <lambda>>, is_key=False, required=False)
  groups = GroupsField(default=<function <lambda>>, is_key=False, required=False)
  tags = ListField(default=<function <lambda>>, is_key=False, required=False)

```

ckan_api_client.utils

class `ckan_api_client.utils.IDPair`

Bases: `ckan_api_client.utils.IDPair`

A pair (named tuple) mapping a “source” id with the one used internally in Ckan.

This is mostly used associated with `IDMap`.

Keys: `source_id`, `ckan_id`

class `ckan_api_client.utils.SuppressExceptionIf` (*cond*)

Bases: `object`

Context manager used to suppress exceptions if they match a given condition.

Usage example:

```
is_404 = lambda x: isinstance(x, HTTPError) and x.status_code == 404
with SuppressExceptionIf(is_404):
    client.request(...)
```

class `ckan_api_client.utils.IDMap`

Bases: `object`

Two-way hashmap to map source ids to ckan ids and the other way back.

to_ckan (*source_id*)

Convert a source id to ckan id

to_source (*ckan_id*)

Convert a ckan id to source id

add (*pair*)

Add a new id pair

Parameters `pair` (`IDPair`) – the id pair to be added

Raises `ValueError` – if one of the two ids is found in a mismatching pair

remove (*pair*)

Remove an id pair.

Parameters `pair` (`IDPair`) – the id pair to be removed

Raises `ValueError` – if one of the two ids is found in a mismatching pair

class `ckan_api_client.utils.FrozenDict` (**a*, ***kw*)

Bases: `_abcoll.MutableMapping`

Frozen dictionary. Acts as a read-only dictionary, preventing changes and returning frozen objects when asked for values.

class `ckan_api_client.utils.FrozenSequence` (*data*)

Bases: `_abcoll.Sequence`

Base class for the `FrozenList`/`FrozenTuple` classes. Acts as a read-only sequence type, returning frozen versions of mutable objects.

class `ckan_api_client.utils.FrozenList` (*data*)

Bases: `ckan_api_client.utils.FrozenSequence`

Immutable list-like.

class `ckan_api_client.utils.FrozenTuple` (*data*)
Bases: `ckan_api_client.utils.FrozenSequence`

Immutable tuple-like.

`ckan_api_client.utils.freeze` (*obj*)
Returns the “frozen” version of a mutable type.

Raises `TypeError` – if a frozen version for that object doesn’t exist

class `ckan_api_client.utils.WrappedList` (**a, **kw*)
Bases: `_abcoll.MutableSequence`

insert (*pos, item*)

All the testing is done via `pytest`. See the pages below on how to run and write tests.

Fixtures

Documentation of the available fixtures for tests.

Fixture functions

Utility objects

Utility functions

Functions used by fixtures.

Testing utilities

Data generation

`ckan_api_client.tests.utils.generate.generate_organization()`

Generate a random organization object, with:

- `name`, random, example: "org-abc123"
- `title`, random, example: "Organization abc123"
- `description`, random
- `image`, url pointing to a random-generated pic

`ckan_api_client.tests.utils.generate.generate_group()`

Generate a random group object, with:

- `name`, random, example: "grp-abc123"
- `title`, random, example: "Group abc123"
- `description`, random
- `image`, url pointing to a random-generated pic

`ckan_api_client.tests.utils.generate.generate_dataset()`

Generate a dataset, populated with random data.

Fields:

- `name` – random string, in the form `dataset-{random}`
- `title` – random string, in the form `Dataset {random}`
- `author` – random-generated name
- `author_email` – random-generated email address
- `license_id` – random license id. One of `cc-by`, `cc-zero`, `cc-by-sa` or `notspecified`.
- `maintainer` – random-generated name
- `maintainer_email` – random-generated email address
- `notes` – random string, containing some markdown
- `owner_org` – set to `None`
- `private` – Fixed to `False`
- `tags` – random list of tags (strings)
- `type` – fixed string: "dataset"
- `url` – random url of dataset on an “external source”
- `extras` – dictionary containing random key / value pairs
- `groups` – empty list
- `resources` – list of random resources
- `relationships` – empty list

Note: The `owner_org` and `groups` fields will be blank, as they must match with existing groups / organizations and we don’t have access to database from here (nor is it in the scope of this function!)

`ckan_api_client.tests.utils.generate.generate_resource()`

Generate a random resource, to be put in a dataset.

Fields:

- `url` – resource URL on an “external source”
- `resource_type` – one of `api` or `file`
- `name` – random-generated name
- `format` – a random format (eg: `csv`, `json`)
- `description` – random generated string

`ckan_api_client.tests.utils.generate.generate_tags (amount)`
 Generate amount random tags. Each tag is in the form `tag-<random-int>`.

Returns a list of tag names

`ckan_api_client.tests.utils.generate.generate_extras (amount)`
 Generate a dict with amount random key/value pairs.

`ckan_api_client.tests.utils.generate.generate_data (dataset_count=50,
 orgs_count=10,
 groups_count=15)`
 Generate a bunch of random data. Will also associate datasets with random organizations / groups.

Returns a dict with the dataset, organization and group keys; each of them a dict of `{key: object}`.

`ckan_api_client.tests.utils.generate.generate_id (length=10)`

HTTP Utilities

Utilities for handling / checking HTTP responses

`ckan_api_client.tests.utils.http.check_response_ok (response, status_code=200)`

Warning: deprecated function. Use `check_api_v3_response()`.

`ckan_api_client.tests.utils.http.check_response_error (response, status_code)`

Warning: deprecated function. Use `check_api_v3_error()`.

`ckan_api_client.tests.utils.http.check_api_v3_response (response, status_code=200)`
 Make sure that `response` is a valid successful response from API v3.

- check http status code to be in the 200-299 range
- check http status code to match `status_code`
- check content-type to be `application/json`
- check charset to be `utf-8`
- check content body to be valid json
- make sure response object contains the `success`, `result` and `help` keys.
- check that `success` is `True`
- check that `error` key is not in the response

Parameters

- **response** – a `requests` response
- **status_code** – http status code to be checked (default: 200)

`ckan_api_client.tests.utils.http.check_api_v3_error (response, status_code)`
 Make sure that `response` is a valid error response from API v3.

- check http status code to match `status_code`

Parameters

- **response** – a requests response
- **status_code** – http status code to be checked

String-related

String generation functions.

`ckan_api_client.tests.utils.strings.generate_password(length=20)`
Generate random password of the given length.

Beware that the string will be generate as random data from urandom, and returned as hexadecimal string of twice the length.

`ckan_api_client.tests.utils.strings.generate_random_alphanum(length=10)`
Generate a random string, made of ascii letters + digits

`ckan_api_client.tests.utils.strings.gen_random_id(length=10)`
Generate a random id, made of lowercase ascii letters + digits

`ckan_api_client.tests.utils.strings.gen_dataset_name()`
Generate a random dataset name

`ckan_api_client.tests.utils.strings.gen_picture(s, size=200)`
Generate URL to picture from some text hash

`ckan_api_client.tests.utils.strings.gen_gravatar(s, size=200)`
Return URL for gravatar of md5 of string s

`ckan_api_client.tests.utils.strings.gen_robohash(s, size=200)`
Return URL for robohash pic for sha1 hash of string s

Validation

Utility functions to validate expectations

`ckan_api_client.tests.utils.validation.check_dataset(dataset, expected)`
Make sure dataset matches the expected one

`ckan_api_client.tests.utils.validation.check_group(group, expected)`
Make sure group matches the expected one

`ckan_api_client.tests.utils.validation.check_organization(organization, expected)`
Make sure organization matches the expected one

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

C

- `ckan_api_client.exceptions`, 11
- `ckan_api_client.high_level`, 5
- `ckan_api_client.low_level`, 3
- `ckan_api_client.objects`, 12
 - `ckan_api_client.objects.base`, 12
 - `ckan_api_client.objects.ckan_dataset`, 14
 - `ckan_api_client.objects.ckan_group`, 15
 - `ckan_api_client.objects.ckan_organization`, 15
 - `ckan_api_client.objects.fields`, 13
- `ckan_api_client.syncing`, 8
- `ckan_api_client.tests.conftest`, 19
 - `ckan_api_client.tests.utils.generate`, 19
 - `ckan_api_client.tests.utils.http`, 21
 - `ckan_api_client.tests.utils.strings`, 22
 - `ckan_api_client.tests.utils.validation`, 22
- `ckan_api_client.utils`, 16

Symbols

`__init__()` (ckan_api_client.syncing.SynchronizationClient method), 8

A

`add()` (ckan_api_client.utils.IDMap method), 16

`anonymous` (ckan_api_client.low_level.CkanLowlevelClient attribute), 3

`approval_status` (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

`approval_status` (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

`author` (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

`author_email` (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

B

`BadApiError`, 11

`BadApiWarning`, 11

`BaseField` (class in ckan_api_client.objects.base), 12

`BaseObject` (class in ckan_api_client.objects.base), 12

C

`check_api_v3_error()` (in module ckan_api_client.tests.utils.http), 21

`check_api_v3_response()` (in module ckan_api_client.tests.utils.http), 21

`check_dataset()` (in module ckan_api_client.tests.utils.validation), 22

`check_group()` (in module ckan_api_client.tests.utils.validation), 22

`check_organization()` (in module ckan_api_client.tests.utils.validation), 22

`check_response_error()` (in module ckan_api_client.tests.utils.http), 21

`check_response_ok()` (in module ckan_api_client.tests.utils.http), 21

`ckan_api_client.exceptions` (module), 11

`ckan_api_client.high_level` (module), 5

`ckan_api_client.low_level` (module), 3

`ckan_api_client.objects` (module), 12

`ckan_api_client.objects.base` (module), 12

`ckan_api_client.objects.ckan_dataset` (module), 14

`ckan_api_client.objects.ckan_group` (module), 15

`ckan_api_client.objects.ckan_organization` (module), 15

`ckan_api_client.objects.fields` (module), 13

`ckan_api_client.syncing` (module), 8

`ckan_api_client.tests.conftest` (module), 19

`ckan_api_client.tests.utils.generate` (module), 19

`ckan_api_client.tests.utils.http` (module), 21

`ckan_api_client.tests.utils.strings` (module), 22

`ckan_api_client.tests.utils.validation` (module), 22

`ckan_api_client.utils` (module), 16

`CkanDataset` (class in ckan_api_client.objects.ckan_dataset), 14

`CkanGroup` (class in ckan_api_client.objects.ckan_group), 15

`CkanHighlevelClient` (class in ckan_api_client.high_level), 5

`CkanLowlevelClient` (class in ckan_api_client.low_level), 3

`CkanOrganization` (class in ckan_api_client.objects.ckan_organization), 15

`CkanResource` (class in ckan_api_client.objects.ckan_dataset), 14

`create_dataset()` (ckan_api_client.high_level.CkanHighlevelClient method), 6

`create_group()` (ckan_api_client.high_level.CkanHighlevelClient method), 8

`create_organization()` (ckan_api_client.high_level.CkanHighlevelClient method), 7

D

`default` (ckan_api_client.objects.base.BaseField attribute), 12

`default` (ckan_api_client.objects.fields.StringField attribute), 13

default() (ckan_api_client.objects.fields.DictField static method), 13	gen_random_id() (in ckan_api_client.tests.utils.strings), 22	module
default() (ckan_api_client.objects.fields.ListField static method), 13	gen_robohash() (in ckan_api_client.tests.utils.strings), 22	module
delete() (ckan_api_client.objects.base.BaseField method), 12	generate_data() (in ckan_api_client.tests.utils.generate), 21	module
delete_dataset() (ckan_api_client.high_level.CkanHighlevelClient method), 6	generate_dataset() (in ckan_api_client.tests.utils.generate), 20	module
delete_dataset() (ckan_api_client.low_level.CkanLowlevelClient method), 4	generate_extras() (in ckan_api_client.tests.utils.generate), 21	module
delete_group() (ckan_api_client.high_level.CkanHighlevelClient method), 8	generate_group() (in ckan_api_client.tests.utils.generate), 19	module
delete_group() (ckan_api_client.low_level.CkanLowlevelClient method), 5	generate_id() (in ckan_api_client.tests.utils.generate), 21	module
delete_organization() (ckan_api_client.high_level.CkanHighlevelClient method), 7	generate_organization() (in ckan_api_client.tests.utils.generate), 19	module
delete_organization() (ckan_api_client.low_level.CkanLowlevelClient method), 5	generate_password() (in ckan_api_client.tests.utils.strings), 22	module
description (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14	generate_random_alphanum() (in ckan_api_client.tests.utils.strings), 22	module
description (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15	generate_resource() (in ckan_api_client.tests.utils.generate), 20	module
description (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15	generate_tag() (in ckan_api_client.tests.utils.generate), 20	module
DictField (class in ckan_api_client.objects.fields), 13	get() (ckan_api_client.objects.base.BaseField method), 12	
E	get_dataset() (ckan_api_client.high_level.CkanHighlevelClient method), 5	
extras (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14	get_dataset() (ckan_api_client.low_level.CkanLowlevelClient method), 4	
extras (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15	get_dataset_by_name() (ckan_api_client.high_level.CkanHighlevelClient method), 5	
extras (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15	get_default() (ckan_api_client.objects.base.BaseField method), 12	
ExtrasField (class in ckan_api_client.objects.fields), 13	get_group() (ckan_api_client.high_level.CkanHighlevelClient method), 7	
F	get_group() (ckan_api_client.low_level.CkanLowlevelClient method), 5	
format (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14	get_group_by_name() (ckan_api_client.high_level.CkanHighlevelClient method), 7	
freeze() (in module ckan_api_client.utils), 17	get_organization() (ckan_api_client.high_level.CkanHighlevelClient method), 6	
from_dict() (ckan_api_client.objects.base.BaseObject class method), 12	get_organization() (ckan_api_client.low_level.CkanLowlevelClient method), 5	
FrozenDict (class in ckan_api_client.utils), 16	get_organization_by_name() (ckan_api_client.high_level.CkanHighlevelClient method), 6	
FrozenList (class in ckan_api_client.utils), 16	groups (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14	
FrozenSequence (class in ckan_api_client.utils), 16	groups (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15	
FrozenTuple (class in ckan_api_client.utils), 16	groups (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15	
G	GroupsField (class in ckan_api_client.objects.fields), 13	
gen_dataset_name() (in ckan_api_client.tests.utils.strings), 22		
gen_gravatar() (in ckan_api_client.tests.utils.strings), 22		
gen_picture() (in ckan_api_client.tests.utils.strings), 22		

H

HTTPError, 11

I

id (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

id (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14

id (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

id (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

IDMap (class in ckan_api_client.utils), 16

IDPair (class in ckan_api_client.utils), 16

image_url (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

image_url (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

insert() (ckan_api_client.utils.WrappedList method), 17

is_equivalent() (ckan_api_client.objects.base.BaseField method), 12

is_equivalent() (ckan_api_client.objects.base.BaseObject method), 13

is_equivalent() (ckan_api_client.objects.ckan_dataset.CkanDataset method), 14

is_equivalent() (ckan_api_client.objects.fields.ExtrasField method), 13

is_key (ckan_api_client.objects.base.BaseField attribute), 12

is_modified() (ckan_api_client.objects.base.BaseField method), 12

is_modified() (ckan_api_client.objects.base.BaseObject method), 13

is_organization (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

is_organization (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

iter_datasets() (ckan_api_client.high_level.CkanHighlevelClient method), 5

iter_datasets() (ckan_api_client.low_level.CkanLowlevelClient method), 4

iter_fields() (ckan_api_client.objects.base.BaseObject method), 13

iter_groups() (ckan_api_client.high_level.CkanHighlevelClient method), 7

iter_groups() (ckan_api_client.low_level.CkanLowlevelClient method), 5

iter_organizations() (ckan_api_client.high_level.CkanHighlevelClient method), 6

iter_organizations() (ckan_api_client.low_level.CkanLowlevelClient method), 5

L

license_id (ckan_api_client.objects.ckan_dataset.CkanDataset

attribute), 14

list_datasets() (ckan_api_client.high_level.CkanHighlevelClient method), 5

list_datasets() (ckan_api_client.low_level.CkanLowlevelClient method), 4

list_group_names() (ckan_api_client.high_level.CkanHighlevelClient method), 7

list_groups() (ckan_api_client.high_level.CkanHighlevelClient method), 7

list_groups() (ckan_api_client.low_level.CkanLowlevelClient method), 4

list_licenses() (ckan_api_client.low_level.CkanLowlevelClient method), 5

list_organization_names() (ckan_api_client.high_level.CkanHighlevelClient method), 6

list_organizations() (ckan_api_client.high_level.CkanHighlevelClient method), 6

list_organizations() (ckan_api_client.low_level.CkanLowlevelClient method), 5

ListField (class in ckan_api_client.objects.fields), 13

M

maintainer (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

maintainer_email (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

message (ckan_api_client.exceptions.HTTPError attribute), 11

mimetype (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14

mimetype_inner (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14

N

name (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

name (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14

name (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

name (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

notes (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

O

OperationFailure, 11

owner_org (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

P

post_dataset() (ckan_api_client.low_level.CkanLowlevelClient method), 4

post_group() (ckan_api_client.low_level.CkanLowlevelClient method), 5

post_organization() (ckan_api_client.low_level.CkanLowlevelClient method), 5

private (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

put_dataset() (ckan_api_client.low_level.CkanLowlevelClient method), 4

put_group() (ckan_api_client.low_level.CkanLowlevelClient method), 5

put_organization() (ckan_api_client.low_level.CkanLowlevelClient method), 5

R

remove() (ckan_api_client.utils.IDMap method), 16

request() (ckan_api_client.low_level.CkanLowlevelClient method), 3

resource_type (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 14

resources (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

ResourcesField (class in ckan_api_client.objects.ckan_dataset), 14

S

save_dataset() (ckan_api_client.high_level.CkanHighlevelClient method), 6

save_group() (ckan_api_client.high_level.CkanHighlevelClient method), 7

save_organization() (ckan_api_client.high_level.CkanHighlevelClient method), 7

serialize() (ckan_api_client.objects.base.BaseField method), 12

serialize() (ckan_api_client.objects.base.BaseObject method), 13

serialize() (ckan_api_client.objects.ckan_dataset.ResourcesField method), 14

set() (ckan_api_client.objects.base.BaseField method), 12

set_initial() (ckan_api_client.objects.base.BaseField method), 12

set_initial() (ckan_api_client.objects.base.BaseObject method), 12

size (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 15

state (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

state (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

state (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

status_code (ckan_api_client.exceptions.HTTPError attribute), 11

StringField (class in ckan_api_client.objects.fields), 13

SuppressExceptionIf (class in ckan_api_client.utils), 16

sync() (ckan_api_client.syncing.SynchronizationClient method), 9

SynchronizationClient (class in ckan_api_client.syncing), 8

T

tags (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

tags (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

tags (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

title (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

title (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

title (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

to_dict() (ckan_api_client.objects.base.BaseObject method), 12

to_source() (ckan_api_client.utils.IDMap method), 16

type (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

type (ckan_api_client.objects.ckan_group.CkanGroup attribute), 15

type (ckan_api_client.objects.ckan_organization.CkanOrganization attribute), 15

U

update_dataset() (ckan_api_client.high_level.CkanHighlevelClient method), 6

update_group() (ckan_api_client.high_level.CkanHighlevelClient method), 8

update_organization() (ckan_api_client.high_level.CkanHighlevelClient method), 7

url (ckan_api_client.objects.ckan_dataset.CkanDataset attribute), 14

url (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 15

url_type (ckan_api_client.objects.ckan_dataset.CkanResource attribute), 15

V

validate() (ckan_api_client.objects.base.BaseField method), 12

validate() (ckan_api_client.objects.ckan_dataset.ResourcesField method), 14

validate() (ckan_api_client.objects.fields.DictField method), 13

validate() (ckan_api_client.objects.fields.ExtrasField method), 13

validate() (ckan_api_client.objects.fields.GroupsField method), 13

validate() (ckan_api_client.objects.fields.ListField
method), 13

validate() (ckan_api_client.objects.fields.StringField
method), 13

W

wipe_dataset() (ckan_api_client.high_level.CkanHighlevelClient
method), 6

WrappedList (class in ckan_api_client.utils), 17