
Wikidata Documentation

Release 0.7.0

Hong Minhee

Jul 30, 2020

Contents

1	wikidata — Wikidata client library	3
1.1	wikidata.cache — Caching policies	3
1.2	wikidata.client — Client session	5
1.3	wikidata.commonsmmedia — Wikimedia Commons	6
1.4	wikidata.datavalue — Interpreting datavalues	7
1.5	wikidata.entity — Wikidata entities	8
1.6	wikidata.globecoordinate — Globe coordinate	9
1.7	wikidata.multilingual — Multilingual texts	9
1.8	wikidata.quantity — Quantity	9
2	Contributing	11
2.1	How to run tests	11
3	Changelog	13
3.1	Version 0.7.0	13
3.2	Version 0.6.1	14
3.3	Version 0.6.0	14
3.4	Version 0.5.4	14
3.5	Version 0.5.3	14
3.6	Version 0.5.2	14
3.7	Version 0.5.1	14
3.8	Version 0.5.0	15
3.9	Version 0.4.4	15
3.10	Version 0.4.3	15
3.11	Version 0.4.2	15
3.12	Version 0.4.1	15
3.13	Version 0.4.0	16
3.14	Version 0.3.0	16
3.15	Version 0.2.0	16
3.16	Version 0.1.0	16
4	Indices and tables	17
	Python Module Index	19
	Index	21

This package provides easy APIs to use [Wikidata](#) for Python.

```
>>> from wikidata.client import Client
>>> client = Client() # doctest: +SKIP
>>> entity = client.get('Q20145', load=True)
>>> entity
<wikidata.entity.Entity Q20145 'IU'>
>>> entity.description
m'South Korean singer and actress'
>>> image_prop = client.get('P18')
>>> image = entity[image_prop]
>>> image
<wikidata.commonsmmedia.File 'File:KBS "The Producers" press conference, 11 May 2015_
↪10.jpg'>
>>> image.image_resolution
(820, 1122)
>>> image.image_url
'https://upload.wikimedia.org/wikipedia/commons/6/60/KBS_%22The_Producers%22_press_
↪conference%2C_11_May_2015_10.jpg'
```


1.1 wikidata.cache — Caching policies

Changed in version 0.5.0.

`wikidata.cache.CacheKey(x)`

The type of keys to look up cached values. Alias of `str`.

class `wikidata.cache.CachePolicy`

Interface for caching policies.

get (*key*: *NewType.<locals>.new_type*) → *Optional*[*NewType.<locals>.new_type*]

Look up a cached value by its key.

Parameters **key** (*CacheKey*) – The key string to look up a cached value.

Returns The cached value if it exists. *None* if there's no such key.

Return type *Optional*[*CacheValue*]

set (*key*: *NewType.<locals>.new_type*, *value*: *Optional*[*NewType.<locals>.new_type*]) → *None*

Create or update a cache.

Parameters

- **key** (*CacheKey*) – A key string to create or update.
- **value** (*Optional*[*CacheValue*]) – A value to cache. *None* to remove cache.

`wikidata.cache.CacheValue(x)`

The type of cached values.

class `wikidata.cache.MemoryCachePolicy(max_size: int = 128)`

LRU (least recently used) cache in memory.

Parameters **max_size** (*int*) – The maximum number of values to cache. 128 by default.

get (*key*: *NewType.<locals>.new_type*) → *Optional*[*NewType.<locals>.new_type*]

Look up a cached value by its key.

Parameters **key** (*CacheKey*) – The key string to look up a cached value.

Returns The cached value if it exists. None if there's no such key.

Return type *Optional[CacheValue]*

set (*key: NewType.<locals>.new_type, value: Optional[NewType.<locals>.new_type]*) → None
Create or update a cache.

Parameters

- **key** (*CacheKey*) – A key string to create or update.
- **value** (*Optional[CacheValue]*) – A value to cache. None to remove cache.

class `wikidata.cache.NullCachePolicy`

No-op cache policy.

get (*key: NewType.<locals>.new_type*) → *Optional[NewType.<locals>.new_type]*
Look up a cached value by its key.

Parameters **key** (*CacheKey*) – The key string to look up a cached value.

Returns The cached value if it exists. None if there's no such key.

Return type *Optional[CacheValue]*

set (*key: NewType.<locals>.new_type, value: Optional[NewType.<locals>.new_type]*) → None
Create or update a cache.

Parameters

- **key** (*CacheKey*) – A key string to create or update.
- **value** (*Optional[CacheValue]*) – A value to cache. None to remove cache.

class `wikidata.cache.ProxyCachePolicy` (*cache_object, timeout: int, property_timeout: Optional[int] = None, namespace: str = 'wd_'*)

This proxy policy is a proxy or an adaptor to another cache object. Cache objects can be anything if they satisfy the following interface:

```
def get(key: str) -> Optional[bytes]: pass
def set(key: str, value: bytes, timeout: int=0) -> None: pass
def delete(key: str) -> None: pass
```

(The above methods omit `self` parameters.) It's compatible with de facto interface for caching libraries in Python (e.g. `python-memcached`, `werkzeug.contrib.cache`).

Parameters

- **cache_object** – The cache object to adapt. Read the above explanation.
- **timeout** (*int*) – Lifespan of every cache in seconds. 0 means no expiration.
- **property_timeout** (*int*) – Lifespan of caches for properties (in seconds). Since properties don't change frequently or their changes usually don't make important effect, longer lifespan of properties' cache can be useful. 0 means no expiration. Set to the same as `timeout` by default.
- **namespace** (*str*) – The common prefix attached to every cache key. 'wd_' by default.

get (*key: NewType.<locals>.new_type*) → *Optional[NewType.<locals>.new_type]*
Look up a cached value by its key.

Parameters **key** (*CacheKey*) – The key string to look up a cached value.

Returns The cached value if it exists. None if there's no such key.

Return type `Optional[CacheValue]`

set (*key*: `NewType.<locals>.new_type`, *value*: `Optional[NewType.<locals>.new_type]`) → `None`
Create or update a cache.

Parameters

- **key** (`CacheKey`) – A key string to create or update.
- **value** (`Optional[CacheValue]`) – A value to cache. `None` to remove cache.

1.2 wikidata.client — Client session

`wikidata.client.WIKIDATA_BASE_URL = 'https://www.wikidata.org/'`
(`str`) The default `base_url` of `Client` constructor.

Changed in version 0.3.0: As the meaning of `Client` constructor's `base_url` parameter, it now became to `https://www.wikidata.org/` from `https://www.wikidata.org/wiki/` (which contained the trailing path `wiki/`).

```
class wikidata.client.Client (base_url: str = 'https://www.wikidata.org/', opener: Optional[urllib.request.OpenerDirector] = None, datavalue_decoder: Union[Decoder, Callable[[Client, str, Mapping[str, object]], object], None] = None, entity_type_guess: bool = True, cache_policy: wikidata.cache.CachePolicy = <wikidata.cache.NullCachePolicy object>, repr_string: Optional[str] = None)
```

Wikidata client session.

Parameters

- **base_url** (`str`) – The base url of the Wikidata. `WIKIDATA_BASE_URL` is used by default.
- **opener** (`urllib.request.OpenerDirector`) – The opener for `urllib.request`. If omitted or `None` the default opener is used.
- **entity_type_guess** (`bool`) – Whether to guess `type` of `Entity` from its `id` for less HTTP requests. `True` by default.
- **cache_policy** – A caching policy for API calls. No cache (`NullCachePolicy`) by default.

New in version 0.5.0: The `cache_policy` option.

Changed in version 0.3.0: The meaning of `base_url` parameter changed. It originally meant `https://www.wikidata.org/wiki/` which contained the trailing path `wiki/`, but now it means only `https://www.wikidata.org/`.

New in version 0.2.0: The `entity_type_guess` option.

cache_policy = `<wikidata.cache.NullCachePolicy object>`
(`CachePolicy`) A caching policy for API calls.

New in version 0.5.0.

datavalue_decoder = `None`

(`Union[Decoder, Callable[[Client, str, Mapping[str, object]], object]]`) The function to decode the given datavalue. It's typically an instance of `Decoder` or its subclass.

decode_datavalue (*datatype*: `str`, *datavalue*: `Mapping[str, object]`) → `object`
Decode the given datavalue using the configured `datavalue_decoder`.

New in version 0.3.0.

entity_type_guess = True

(*bool*) Whether to guess *type* of *Entity* from its *id* for less HTTP requests.

New in version 0.2.0.

get (*entity_id*: *NewType.<locals>.new_type*, *load*: *bool = False*) → *wikidata.entity.Entity*

Get a Wikidata entity by its *EntityId*.

Parameters

- **entity_id** – The *id* of the *Entity* to find.
- **load** (*bool*) – Eager loading on *True*. Lazy loading (*False*) by default.

Returns The found entity.

Return type *Entity*

New in version 0.3.0: The *load* option.

guess_entity_type (*entity_id*: *NewType.<locals>.new_type*) → *Optional[wikidata.entity.EntityType]*

Guess *EntityType* from the given *EntityId*. It could return *None* when it fails to guess.

Note: It always fails to guess when *entity_type_guess* is configured to *False*.

Returns The guessed *EntityId*, or *None* if it fails to guess.

Return type *Optional[EntityType]*

New in version 0.2.0.

1.3 wikidata.commonsmmedia — Wikimedia Commons

New in version 0.3.0.

class *wikidata.commonsmmedia.File* (*client*: *wikidata.client.Client*, *title*: *str*)

Represent a file on [Wikimedia Commons](#).

image_mimetype

(*Optional[str]*) The MIME type of the image. It may be *None* if it's not an image.

image_resolution

(*Optional[Tuple[int, int]]*) The (width, height) pair of the image. It may be *None* if it's not an image.

image_size

(*Optional[int]*) The size of the image in bytes. It may be *None* if it's not an image.

image_url

(*Optional[str]*) The image url. It may be *None* if it's not an image.

page_url

(*str*) The canonical url of the page.

exception *wikidata.commonsmmedia.FileError*

Exception raised when something goes wrong with *File*.

1.4 wikidata.datavalue — Interpreting datavalues

This module provides the decoder interface for customizing how datavalues are decoded, and the default *Decoder* implementation.

Technically the interface is just a callable so that its implementation doesn't necessarily have to be an instance of *Decoder* or its subclass, but only need to satisfy:

```
typing.Callable[[wikidata.client.Client, str, typing.Mapping[str, object]],
                object]
```

New in version 0.3.0.

exception `wikidata.datavalue.DatavalueError` (*args)

Exception raised during decoding datavalues. It subclasses `ValueError` as well.

datavalue

The datavalue which caused the decoding error.

class `wikidata.datavalue.Decoder`

Decode the given datavalue to a value of the appropriate Python type. For extensibility it uses visitor pattern and is intended to be subclassed. To customize decoding of datavalues subclass it and configure `datavalue_decoder` option of *Client* to the customized decoder.

It automatically invokes an appropriate visitor method using a simple rule of name: `{datatype}__{datavalue[type]}`. For example, if the following call to a decoder was made:

```
decoder(client, 'mydatatype', {'type': 'mytype', 'value': '...'})
```

it's delegated to the following visitor method call:

```
decoder.mydatatype__mytype(client, {'type': 'mytype', 'value': '...'})
```

If a decoder failed to find a visitor method matched to `{datatype}__{datavalue[type]}` pattern it secondly try to find a general version of visitor method: `{datavalue[type]}` which lacks double underscores. For example, for the following call:

```
decoder(client, 'mydatatype', {'type': 'mytype', 'value': '...'})
```

It firstly try to find the following visitor method:

```
decoder.mydatatype__mytype
```

but if there's no such method it secondly try to find the following general visitor method:

```
decoder.mytype
```

This twice-try dispatch is useful when to make a visitor method to be matched regardless of datatype.

If its `datavalue[type]` contains hyphens they're replaced by underscores. For example:

```
decoder(client, 'string',
        {'type': 'wikibase-entityid', 'value': 'a text value'})
```

the above call is delegated to the following visitor method call:

```
decoder.string__wikibase_entityid(
    #     Note that the ^ underscore
    client,
    {'type': 'wikibase-entityid', 'value': 'a text value'})
)
```

1.5 wikidata.entity — Wikidata entities

class `wikidata.entity.Entity` (*id*: `NewType.<locals>.new_type`, *client*: `Client`)

Wikidata entity. Can be an item or a property. Its attributes can be lazily loaded.

To get an entity use `Client.get()` method instead of the constructor of `Entity`.

Note: Although it implements `Mapping[EntityId, object]`, it actually is multidict. See also `getlist()` method.

Changed in version 0.2.0: Implemented `Mapping[EntityId, object]` protocol for easy access of statement values.

Changed in version 0.2.0: Implemented `Hashable` protocol and `==/=` operators for equality test.

state

(`EntityState`) The loading state.

New in version 0.7.0.

getlist (*key*: `wikidata.entity.Entity`) → `Sequence[object]`

Return all values associated to the given *key* property in sequence.

Parameters *key* (`Entity`) – The property entity.

Returns A sequence of all values associated to the given *key* property. It can be empty if nothing is associated to the property.

Return type `Sequence[object]`

lists () → `Sequence[Tuple[wikidata.entity.Entity, Sequence[object]]]`

Similar to `items()` except the returning pairs have each list of values instead of each single value.

Returns The pairs of (*key*, *values*) where *values* is a sequence.

Return type `Sequence[Tuple[Entity, Sequence[object]]]`

type

(`EntityType`) The type of entity, *item* or *property*.

New in version 0.2.0.

`wikidata.entity.EntityId` (*x*)

The identifier of each `Entity`. Alias of `str`.

class `wikidata.entity.EntityState`

Define state of `Entity`.

New in version 0.7.0.

loaded = `'loaded'`

(`EntityState`) The entity exists and is already loaded.

non_existent = `'non_existent'`

(`EntityState`) The entity does not exist.

not_loaded = `'not_loaded'`

(`EntityState`) Not loaded yet. Unknown whether the entity does exist or not.

class `wikidata.entity.EntityType`

The enumerated type which consists of two possible values:

- *item*

- *property*

New in version 0.2.0.

item = 'item'

(*EntityType*) Items are *Entity* objects that are typically represented by Wikipage (at least in some Wikipedia languages). They can be viewed as “the thing that a Wikipage is about,” which could be an individual thing (the person *Albert Einstein*), a general class of things (the class of all *Physicists*), and any other concept that is the subject of some Wikipedia page (including things like *History of Berlin*).

See also:

Items — Wikibase Data Model The data model of Wikibase describes the structure of the data that is handled in Wikibase.

property = 'property'

(*EntityType*) Properties are *Entity* objects that describe a relationship between items (or other *Entity* objects) and values of the property. Typical properties are *population* (using numbers as values), *binomial name* (using strings as values), but also *has father* and *author of* (both using items as values).

See also:

Properties — Wikibase Data Model The data model of Wikibase describes the structure of the data that is handled in Wikibase.

1.6 wikidata.globecoordinate — Globe coordinate

New in version 0.7.0.

class wikidata.globecoordinate.GlobeCoordinate (*latitude: float, longitude: float, globe: wikidata.entity.Entity, precision: float*)

Literal data for a geographical position given as a latitude-longitude pair in gms or decimal degrees for the given stellar body.

1.7 wikidata.multilingual — Multilingual texts

wikidata.multilingual.Locale (*x*)

The locale of each *MonolingualText* or internal mapping of each *MultilingualText*. Alias of *str*.

New in version 0.7.0.

class wikidata.multilingual.MonolingualText

Locale-denoted text. It’s almost equivalent to *str* (and indeed subclasses *str*) except that it has an extra attribute, *locale*, that denotes what language the text is written in.

locale = None

(*Locale*) The code of *locale*.

1.8 wikidata.quantity — Quantity

New in version 0.7.0.

class `wikidata.quantity.Quantity` (*amount: float, lower_bound: Optional[float], upper_bound: Optional[float], unit: Optional[wikidata.entity.Entity]*)

A Quantity value represents a decimal number, together with information about the uncertainty interval of this number, and a unit of measurement.

2.1 How to run tests

As this project supports various Python interpreters (CPython and PyPy) and versions, to ensure it works well with them, we use `tox`. You don't need to create a virtual environment by yourself. `tox` automatically creates virtual environments for various Python versions and run the same test suite on all of them.

The easiest to install `tox` is to use `pip`¹:

```
pip install tox
```

Once you've installed `tox`, it's very simple to run the test suite on all Python versions this project aims to support:

```
tox
```

Note that you need to install Python interpreters besides `tox`. If you don't want to install all of them use `--skip-missing-interpreters` option:

```
tox --skip-missing-interpreters
```

To run tests on multiple interpreters at a time, use `--parallel` option:

```
tox --parallel
```

¹ See also the `tox`'s official docs.

3.1 Version 0.7.0

Released on July 31, 2020.

- Marked the package as supporting type checking by following [PEP 561](#).
- Now non-existent entities became able to be handled. [#11]
 - Added `EntityState` enum class.
 - Added `Entity.state` attribute.
 - Fixed a bug that raised `HTTPError` when non-existent `Entity` was requested.
- Languages (locales) became no more represented as `babel.core.Locale`, but represented `wikidata.multilingual.Locale` instead. [#2, #27, #30 by Nelson Liu]
 - Removed `Babel` from the dependencies.
 - Added `wikidata.multilingual.Locale` type.
To replace the `babel.core.Locale` type, the `wikidata.multilingual.Locale` type has been aliased to `str`. This is a *breaking change* for all Wikidata public API functions that formerly returned or ingested `babel.core.Locale`.
- Added support for `time` datatypes with precision 9 (year-only). [#26 by Nelson Liu]
- Added support for globe coordinate datatype. [#28 by Nelson Liu]
 - Added support for decoding the `globe-coordinate` datatype.
 - Added `wikidata.globecoordinate` module.
- Added support for quantity datatype. [#29 by Nelson Liu]
 - Added support for decoding the `quantity` datatype.
 - Added `wikidata.quantity` module. [#29]
- Fixed `KeyError` from `Entity.getlist()` if the property is explicitly associated with “no value”. [#18]

- Fixed a bug that raised `KeyError` when accessing an image more than once and `MemoryCachePolicy` was enabled. [#24 by Héctor Cordobés]

3.2 Version 0.6.1

Released on September 18, 2017.

- Fixed `ImportError` on Python 3.4 due to lack of `typing` module. [#4]

3.3 Version 0.6.0

Released on September 12, 2017.

- Fixed `KeyError` from `Client.get()` on an entity is redirected to its canonical entity.

3.4 Version 0.5.4

Released on September 18, 2017.

- Fixed `ImportError` on Python 3.4 due to lack of `typing` module. [#4]

3.5 Version 0.5.3

Released on June 30, 2017.

- Fixed `ValueError` from `Entity.label/Entity.description` with languages `ISO 639-1` doesn't cover (e.g. `cbk-zam`). [#2]

Although this fix prevents these properties from raising `ValueError`, it doesn't completely fix the problem. `babel.core.Locale` type, which Wikidata depends on, currently doesn't support languages other than `ISO 639-1`. In order to completely fix the problem, we need to patch `Babel` to support them, or make Wikidata independent from `Babel`.

3.6 Version 0.5.2

Released on June 28, 2017.

- Fixed `AssertionError` from empty `multilingual_attributes`.

3.7 Version 0.5.1

Released on June 28, 2017.

- Fixed `AssertionError` from `len()` or iterating (`iter()`) on `Entity` objects with empty claims.

3.8 Version 0.5.0

Released on June 13, 2017.

- Wikidata API calls over network became possible to be cached.
 - `Client` now has `cache_policy` attribute and constructor option. Nothing is cached by default.
 - Added `wikidata.cache` module and `CachePolicy` interface in it. Two built-in implementation of the interface were added:
 - `NullCachePolicy` No-op.
 - `MemoryCachePolicy` LRU cache in memory.
 - `ProxyCachePolicy` Proxy/adapter to another proxy object. Useful for utilizing third-party cache libraries.
 - `wikidata.client.Client.request` logger became to record logs about cache hits as `DEBUG` level.

3.9 Version 0.4.4

Released on June 30, 2017.

- Fixed `ValueError` from `Entity.label/Entity.description` with languages `ISO 639-1` doesn't cover (e.g. `cbk-zam`). [#2]
- Although this fix prevents these properties from raising `ValueError`, it doesn't completely fix the problem. `babel.core.Locale` type, which Wikidata depends on, currently doesn't support languages other than `ISO 639-1`. In order to completely fix the problem, we need to patch `Babel` to support them, or make Wikidata independent from `Babel`.

3.10 Version 0.4.3

Released on June 28, 2017.

- Fixed `AssertionError` from empty `multilingual_attributes`.

3.11 Version 0.4.2

Released on June 28, 2017.

- Fixed `AssertionError` from `len()` or iterating (`iter()`) on `Entity` objects with empty claims.

3.12 Version 0.4.1

Released on April 30, 2017.

- Fixed `AssertionError` from `getlist()` on entities with empty claims.

3.13 Version 0.4.0

Released on April 24, 2017.

- Monolingual texts became able to be handled.
 - Added *MonolingualText* type which is a true subtype of *str*.

3.14 Version 0.3.0

Released on February 23, 2017.

- Now *Client* became able to customize how it decodes datavalues to Python objects.
 - Added *wikidata.datavalue* module and *Decoder* class inside it.
 - Added *datavalue_decoder* option to *Client*.
- Now files on Wikimedia Commons became able to be handled.
 - New decoder became able to parse Wikimedia Commons files e.g. images.
 - Added *wikidata.commonsmidia* module and *File* class inside it.
- The meaning of *Client* constructor's *base_url* parameter became not to contain the trailing path *wiki/* from *https://www.wikidata.org/wiki/*. As its meaning changed, the value of *WIKIDATA_BASE_URL* constant also changed to not have the trailing path.
- Added *load* option to *Client.get()* method.

3.15 Version 0.2.0

Released on February 19, 2017.

- Made *Entity* multidict. Now it satisfies *Mapping[Entity, object]* protocol.
- Added *Entity.type* property and *EntityType* enum class to represent it.
- Added *entity_type_guess* option and *guess_entity_type()* method to *Client* class.
- Implemented *Hashable* protocol and *==/=* operators to *Entity* for equality test.

3.16 Version 0.1.0

Initial version. Released on February 15, 2017.

CHAPTER 4

Indices and tables

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

W

- wikidata, 1
- wikidata.cache, 3
- wikidata.client, 5
- wikidata.commonsmidia, 6
- wikidata.datavalue, 6
- wikidata.entity, 7
- wikidata.globecoordinate, 9
- wikidata.multilingual, 9
- wikidata.quantity, 9

C

cache_policy (*wikidata.client.Client* attribute), 5
 CacheKey() (*in module wikidata.cache*), 3
 CachePolicy (*class in wikidata.cache*), 3
 CacheValue() (*in module wikidata.cache*), 3
 Client (*class in wikidata.client*), 5

D

datavalue (*wikidata.datavalue.DatavalueError* attribute), 7
 datavalue_decoder (*wikidata.client.Client* attribute), 5
 DatavalueError, 7
 decode_datavalue() (*wikidata.client.Client* method), 5
 Decoder (*class in wikidata.datavalue*), 7

E

Entity (*class in wikidata.entity*), 8
 entity_type_guess (*wikidata.client.Client* attribute), 6
 EntityId() (*in module wikidata.entity*), 8
 EntityState (*class in wikidata.entity*), 8
 EntityType (*class in wikidata.entity*), 8

F

File (*class in wikidata.commonsmmedia*), 6
 FileError, 6

G

get() (*wikidata.cache.CachePolicy* method), 3
 get() (*wikidata.cache.MemoryCachePolicy* method), 3
 get() (*wikidata.cache.NullCachePolicy* method), 4
 get() (*wikidata.cache.ProxyCachePolicy* method), 4
 get() (*wikidata.client.Client* method), 6
 getlist() (*wikidata.entity.Entity* method), 8
 GlobeCoordinate (*class in wikidata.globecoordinate*), 9

guess_entity_type() (*wikidata.client.Client* method), 6

I

image_mimetype (*wikidata.commonsmmedia.File* attribute), 6
 image_resolution (*wikidata.commonsmmedia.File* attribute), 6
 image_size (*wikidata.commonsmmedia.File* attribute), 6
 image_url (*wikidata.commonsmmedia.File* attribute), 6
 item (*wikidata.entity.EntityType* attribute), 9

L

lists() (*wikidata.entity.Entity* method), 8
 loaded (*wikidata.entity.EntityState* attribute), 8
 locale (*wikidata.multilingual.MonolingualText* attribute), 9
 Locale() (*in module wikidata.multilingual*), 9

M

MemoryCachePolicy (*class in wikidata.cache*), 3
 MonolingualText (*class in wikidata.multilingual*), 9

N

non_existent (*wikidata.entity.EntityState* attribute), 8
 not_loaded (*wikidata.entity.EntityState* attribute), 8
 NullCachePolicy (*class in wikidata.cache*), 4

P

page_url (*wikidata.commonsmmedia.File* attribute), 6
 property (*wikidata.entity.EntityType* attribute), 9
 ProxyCachePolicy (*class in wikidata.cache*), 4
 Python Enhancement Proposals
 PEP 561, 13

Q

Quantity (*class in wikidata.quantity*), 9

S

`set()` (*wikidata.cache.CachePolicy method*), 3
`set()` (*wikidata.cache.MemoryCachePolicy method*), 4
`set()` (*wikidata.cache.NullCachePolicy method*), 4
`set()` (*wikidata.cache.ProxyCachePolicy method*), 5
`state` (*wikidata.entity.Entity attribute*), 8

T

`type` (*wikidata.entity.Entity attribute*), 8

W

`wikidata` (*module*), 1
`wikidata.cache` (*module*), 3
`wikidata.client` (*module*), 5
`wikidata.commonsmmedia` (*module*), 6
`wikidata.datavalue` (*module*), 6
`wikidata.entity` (*module*), 7
`wikidata.globecoordinate` (*module*), 9
`wikidata.multilingual` (*module*), 9
`wikidata.quantity` (*module*), 9
`WIKIDATA_BASE_URL` (*in module wikidata.client*), 5