Wikipedia-API Documentation

Release 0.6.0

Martin Majlis

Jun 29, 2023

Contents

`Wikipedia-API` is easy to use Python wrapper for Wikipedias' API. It supports extracting texts, sections, links, categories, translations, etc from Wikipedia. Documentation provides code snippets for the most common use cases.

CHAPTER 1

Installation

This package requires at least Python 3.4 to install because it's using IntEnum.

```
pip3 install wikipedia-api
```

CHAPTER 2

Usage

Goal of `Wikipedia-API` is to provide simple and easy to use API for retrieving informations from Wikipedia. Bellow are examples of common use cases.

2.1 Importing

```
import wikipediaapi
```

2.2 How To Get Single Page

Getting single page is straightforward. You have to initialize `Wikipedia` object and ask for page by its name. To initialize it, you have to provide:

- *user_agent* to identify your project. Please follow the recommended format.
- *language* to specify language mutation. It has to be one of supported languages.

```
import wikipediaapi
    wiki_wiki = wikipediaapi.Wikipedia('MyProjectName (merlin@example.com)', 'en')

    page_py = wiki_wiki.page('Python_(programming_language)')
```

2.3 How To Check If Wiki Page Exists

For checking, whether page exists, you can use function `exists`.

```
page_py = wiki_wiki.page('Python_(programming_language)')
print("Page - Exists: %s" % page_py.exists())
# Page - Exists: True

page_missing = wiki_wiki.page('NonExistingPageWithStrangeName')
print("Page - Exists: %s" %     page_missing.exists())
# Page - Exists: False
```

2.4 How To Get Page Summary

Class `WikipediaPage` has property `summary`, which returns description of Wiki page.

```
import wikipediaapi
    wiki_wiki = wikipediaapi.Wikipedia('MyProjectName (merlin@example.com)', 'en')

    print("Page - Title: %s" % page_py.title)
    # Page - Title: Python (programming language)

    print("Page - Summary: %s" % page_py.summary[0:60])
    # Page - Summary: Python is a widely used high-level programming language for
```

2.5 How To Get Page URL

`WikipediaPage` has two properties with URL of the page. It is `fullurl` and `canonicalurl`.

```
print(page_py.fullurl)
# https://en.wikipedia.org/wiki/Python_(programming_language)

print(page_py.canonicalurl)
# https://en.wikipedia.org/wiki/Python_(programming_language)
```

2.6 How To Get Full Text

To get full text of Wikipedia page you should use property `text` which constructs text of the page as concatanation of summary and sections with their titles and texts.

```
wiki_wiki = wikipediaapi.Wikipedia(
    user_agent='MyProjectName (merlin@example.com)',
        language='en',
        extract_format=wikipediaapi.ExtractFormat.WIKI
)

p_wiki = wiki_wiki.page("Test 1")
print(p_wiki.text)
# Summary
# Section 1
# Text of section 1
# Section 1.1
# Text of section 1.1
```

(continues on next page)

(continued from previous page)

```
# ...


wiki_html = wikipediaapi.Wikipedia(
    user_agent='MyProjectName (merlin@example.com)',
        language='en',
        extract_format=wikipediaapi.ExtractFormat.HTML
)
p_html = wiki_html.page("Test 1")
print(p_html.text)
# <p>Summary</p>
# <h2>Section 1</h2>
# <p>Text of section 1</p>
# <h3>Section 1.1</h3>
# <p>Text of section 1.1</p>
# ...
```

2.7 How To Get Page Sections

To get all top level sections of page, you have to use property `sections`. It returns list of `WikipediaPageSection`, so you have to use recursion to get all subsections.

```
def print_sections(sections, level=0):
        for s in sections:
                print("%s: %s - %s" % ("*" * (level + 1), s.title, s.text[0:40]))
                print_sections(s.sections, level + 1)


print_sections(page_py.sections)
# *: History - Python was conceived in the late 1980s,
# *: Features and philosophy - Python is a multi-paradigm programming l
# *: Syntax and semantics - Python is meant to be an easily readable
# **: Indentation - Python uses whitespace indentation, rath
# **: Statements and control flow - Python's statements include (among other
# **: Expressions - Some Python expressions are similar to l
```

2.8 How To Get Page Section By Title

To get last section of page with given title, you have to use function `section_by_title`. It returns the last `WikipediaPageSection` with this title.

```
section_history = page_py.section_by_title('History')
print("%s - %s" % (section_history.title, section_history.text[0:40]))

# History - Python was conceived in the late 1980s b
```

2.9 How To Get All Page Sections By Title

To get all sections of page with given title, you have to use function `sections_by_title`. It returns the all `WikipediaPageSection` with this title.

```
    page_1920 = wiki_wiki.page('1920')
    sections_january = page_1920.sections_by_title('January')
    for s in sections_january:
        print("* %s - %s" % (s.title, s.text[0:40]))

# * January - January 1
# Polish-Soviet War in 1920: The
# * January - January 2
# Isaac Asimov, American author
# * January - January 1 - Zygmunt Gorazdowski, Polish
```

2.10 How To Get Page In Other Languages

If you want to get other translations of given page, you should use property `langlinks`. It is map, where key is language code and value is `WikipediaPage`.

```
def print_langlinks(page):
        langlinks = page.langlinks
        for k in sorted(langlinks.keys()):
            v = langlinks[k]
            print("%s: %s - %s: %s" % (k, v.language, v.title, v.fullurl))

print_langlinks(page_py)
# af: af - Python (programmeertaal): https://af.wikipedia.org/wiki/Python_
↪(programmeertaal)
# als: als - Python (Programmiersprache): https://als.wikipedia.org/wiki/Python_
↪(Programmiersprache)
# an: an - Python: https://an.wikipedia.org/wiki/Python
# ar: ar - : https://ar.wikipedia.org/wiki/%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86
# as: as - : https://as.wikipedia.org/wiki/%E0%A6%AA%E0%A6%BE%E0%A6%87%E0%A6%A5%E0%A6
↪%A8

page_py_cs = page_py.langlinks['cs']
print("Page - Summary: %s" % page_py_cs.summary[0:60])
# Page - Summary: Python (anglická výslovnost [paiθtn]) je vysokoúrovňový sk
```

2.11 How To Get Links To Other Pages

If you want to get all links to other wiki pages from given page, you need to use property `links`. It's map, where key is page title and value is `WikipediaPage`.

```
def print_links(page):
        links = page.links
        for title in sorted(links.keys()):
            print("%s: %s" % (title, links[title]))

print_links(page_py)
# 3ds Max: 3ds Max (id: ??, ns: 0)
# ?:: ?: (id: ??, ns: 0)
# ABC (programming language): ABC (programming language) (id: ??, ns: 0)
# ALGOL 68: ALGOL 68 (id: ??, ns: 0)
```

(continues on next page)

(continued from previous page)

```
# Abaqus: Abaqus (id: ??, ns: 0)
# ...
```

2.12 How To Get Page Categories

If you want to get all categories under which page belongs, you should use property `categories`. It's map, where key is category title and value is `WikipediaPage`.

```
def print_categories(page):
        categories = page.categories
        for title in sorted(categories.keys()):
            print("%s: %s" % (title, categories[title]))


print("Categories")
print_categories(page_py)
# Category:All articles containing potentially dated statements: ...
# Category:All articles with unsourced statements: ...
# Category:Articles containing potentially dated statements from August 2016: ...
# Category:Articles containing potentially dated statements from March 2017: ...
# Category:Articles containing potentially dated statements from September 2017: ...
```

2.13 How To Get All Pages From Category

To get all pages from given category, you should use property `categorymembers`. It returns all members of given category. You have to implement recursion and deduplication by yourself.

```
def print_categorymembers(categorymembers, level=0, max_level=1):
        for c in categorymembers.values():
            print("%s: %s (ns: %d)" % ("*" * (level + 1), c.title, c.ns))
            if c.ns == wikipediaapi.Namespace.CATEGORY and level < max_level:
                print_categorymembers(c.categorymembers, level=level + 1, max_
↪level=max_level)


cat = wiki_wiki.page("Category:Physics")
print("Category members: Category:Physics")
print_categorymembers(cat.categorymembers)

# Category members: Category:Physics
# * Statistical mechanics (ns: 0)
# * Category:Physical quantities (ns: 14)
# ** Refractive index (ns: 0)
# ** Vapor quality (ns: 0)
# ** Electric susceptibility (ns: 0)
# ** Specific weight (ns: 0)
# ** Category:Viscosity (ns: 14)
# *** Brookfield Engineering (ns: 0)
```

2.14 How To See Underlying API Call

If you have problems with retrieving data you can get URL of undrerlying API call. This will help you determine if the problem is in the library or somewhere else.

```
import wikipediaapi
import sys
wikipediaapi.log.setLevel(level=wikipediaapi.logging.DEBUG)

# Set handler if you use Python in interactive mode
out_hdlr = wikipediaapi.logging.StreamHandler(sys.stderr)
out_hdlr.setFormatter(wikipediaapi.logging.Formatter('%(asctime)s %(message)s'))
out_hdlr.setLevel(wikipediaapi.logging.DEBUG)
wikipediaapi.log.addHandler(out_hdlr)

wiki = wikipediaapi.Wikipedia(user_agent='MyProjectName (merlin@example.com)',␣
↪language='en')

page_ostrava = wiki.page('Ostrava')
print(page_ostrava.summary)
# logger prints out: Request URL: http://en.wikipedia.org/w/api.php?action=query&
↪prop=extracts&titles=Ostrava&explaintext=1&exsectionformat=wiki
```

CHAPTER 3

External Links

- GitHub
- PyPi
- Travis
- ReadTheDocs

CHAPTER 4

Other Badges

CHAPTER 5

Other Pages

5.1 API

5.1.1 Wikipedia

- `__init__(user_agent: str, language='en', extract_format=ExtractFormat.WIKI, headers: Optional[Dict[str, Any]] = None, **kwargs)`
- `page(title)`

5.1.2 WikipediaPage

- `exists()`
- `pageid`
- `title` - title
- `summary` - summary of the page
- `text` - returns text of the page
- `sections` - list of all sections (list of `WikipediaPageSection`)
- `langlinks` - language links to other languages ({lang: `WikipediaLangLink`})
- `section_by_title(name)` - finds last section by title (`WikipediaPageSection`)
- `sections_by_title(name)` - finds all section by title (`WikipediaPageSection`)
- `links` - links to other pages ({title: `WikipediaPage`})
- `categories` - all categories ({title: `WikipediaPage`})
- `displaytitle`
- `canonicalurl`

- ns
- contentmodel
- pagelanguage
- pagelanguagehtmlcode
- pagelanguagedir
- touched
- lastrevid
- length
- protection
- restrictiontypes
- watchers
- notificationtimestamp
- talkid
- fullurl
- editurl
- readable
- preload

5.1.3 WikipediaPageSection

- title
- level
- text
- sections
- section_by_title(title)

5.1.4 ExtractFormat

- WIKI
- HTML

5.2 Changelog

5.2.1 0.6.0

- Make user agent mandatory - Issue 63
- This breaks the API since *user_agent* is now the first parameter.

5.2.2 0.5.8

- Adds support for retrieving all sections with given name - Issue 39

5.2.3 0.5.4

- Namespace could be arbitrary integer - Issue 29

5.2.4 0.5.3

- **Adds persistent HTTP connection - Issue 26**
  - Downloading 50 pages reduced from 13s to 8s => 40% speed up

5.2.5 0.5.2

- Adds namespaces 102 - 105 - Issue 24

5.2.6 0.5.1

- Adds tox for testing different Python versions

5.2.7 0.5.0

- Allows modifying API call parameters
- Fixes Issue 16 - hidden categories
- Fixes Issue 21 - summary extraction

5.2.8 0.4.5

- Handles missing sections correctly
- Fixes Issue 20

5.2.9 0.4.4

- Uses HTTPS directly instead of HTTP to avoid redirect

5.2.10 0.4.3

- Correctly extracts text from pages without sections
- Adds support for quoted page titles

```
api = wikipediaapi.Wikipedia(
    language='hi',
)
python = api.article(
    title='%E0%A4%AA%E0%A4%BE%E0%A4%87%E0%A4%A5%E0%A4%A8',
    unquote=True,
)
print(python.summary)
```

5.2.11 0.4.2

- Adds support for Python 3.4 by not using f-strings

5.2.12 0.4.1

- Uses code style enforced by flake8
- Increased code coverage

5.2.13 0.4.0

- Uses type annotations => minimal requirement is now Python 3.5
- Adds possibility to use more parameters for request. For example:

```
api = wikipediaapi.Wikipedia(
    language='en',
    proxies={'http': 'http://localhost:1234'}
)
```

- Extends documentation

5.2.14 0.3.4

- Adds support for property Categorymembers
- Adds property `text` for retrieving complete text of the page

5.2.15 0.3.3

- Added support for request timeout
- Add header: Accept-Encoding: gzip

5.2.16 0.3.2

- Added support for property Categories

5.2.17 0.3.1

- Removing `WikipediaLangLink`
- Page keeps track of its own language, so it's easier to jump between different translations of the same page

5.2.18 0.3.0

- Rename directory from `wikipedia` to `wikipediaapi` to avoid collisions

5.2.19 0.2.4

- Handle redirects properly

5.2.20 0.2.3

- Usage method `page` instead of `article` in `Wikipedia`

5.2.21 0.2.2

- Added support for property Links

5.2.22 0.2.1

- Added support for property Langlinks

5.2.23 0.2.0

- Use properties instead of functions
- Added support for property Info

5.2.24 0.1.6

- Support for extracting texts with HTML markdown
- Added initial version of unit tests

5.2.25 0.1.4

- It's possible to extract summary and sections of the page
- Added support for property Extracts

5.3 Development

5.3.1 Makefile targets

- `make release` - based on version specified in `wikipedia/__init__.py` creates new release as well as git tag
- `make run-tests` - run unit tests
- `make run-coverage` - run code coverage
- `make pypi-html` - generates single HTML documentation into `pypi-doc.html`
- `make html` - generates HTML documentation similar to RTFD into folder `_build/html/`
- `make requirements` - install requirements
- `make requirements-dev` - install development requirements

5.3.2 Usage Statistics

- PIP Downloads

5.3.3 Underlying API

- API - HP
- Module - Parse
- Module - Query

5.4 wikipediaapi

Wikipedia-API is easy to use wrapper for extracting information from Wikipedia.

It supports extracting texts, sections, links, categories, translations, etc. from Wikipedia. Documentation provides code snippets for the most common use cases.

`wikipediaapi.`**`namespace2int`**(*namespace: Union[wikipediaapi.Namespace, int]*) → int
Converts namespace into integer

class `wikipediaapi.`**`Wikipedia`**(*user_agent: str*, *language: str = 'en'*, *extract_format: wikipediaapi.ExtractFormat = <ExtractFormat.WIKI: 1>*, *headers: Optional[Dict[str, Any]] = None*, ***kwargs*)
Wikipedia is wrapper for Wikipedia API.

`__del__`() → None
Closes session.

`__init__`(*user_agent: str*, *language: str = 'en'*, *extract_format: wikipediaapi.ExtractFormat = <ExtractFormat.WIKI: 1>*, *headers: Optional[Dict[str, Any]] = None*, ***kwargs*) → None
Constructs Wikipedia object for extracting information Wikipedia.

Parameters

- **`user_agent`** – HTTP User-Agent used in requests https://meta.wikimedia.org/wiki/User-Agent_policy

- **language** – Language mutation of Wikipedia - http://meta.wikimedia.org/wiki/List_of_Wikipedias
- **extract_format** – Format used for extractions *ExtractFormat* object.
- **headers** – Headers sent as part of HTTP request
- **kwargs** – Optional parameters used in - http://docs.python-requests.org/en/master/api/#requests.request

Examples:

- Proxy: `Wikipedia('foo (merlin@example.com)', proxies={'http': 'http://proxy:1234'})`

article(*title: str*, *ns: Union[wikipediaapi.Namespace, int] = <Namespace.MAIN: 0>*, *unquote: bool = False*) → wikipediaapi.WikipediaPage

Constructs Wikipedia page with title *title*.

This function is an alias for *page()*

Parameters

- **title** – page title as used in Wikipedia URL
- **ns** – WikiNamespace
- **unquote** – if true it will unquote title

Returns object representing *WikipediaPage*

backlinks(*page: wikipediaapi.WikipediaPage*, ***kwargs*) → Dict[str, wikipediaapi.WikipediaPage]

Returns backlinks from other pages with respect to parameters

API Calls for parameters:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bbacklinks
- https://www.mediawiki.org/wiki/API:Backlinks

Parameters

- **page** – *WikipediaPage*
- **kwargs** – parameters used in API call

Returns backlinks from other pages

categories(*page: wikipediaapi.WikipediaPage*, ***kwargs*) → Dict[str, wikipediaapi.WikipediaPage]

Returns categories for page with respect to parameters

API Calls for parameters:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bcategories
- https://www.mediawiki.org/wiki/API:Categories

Parameters

- **page** – *WikipediaPage*
- **kwargs** – parameters used in API call

Returns categories for page

categorymembers (*page: wikipediaapi.WikipediaPage*, ***kwargs*) → Dict[str, wikipediaapi.WikipediaPage]
Returns pages in given category with respect to parameters

API Calls for parameters:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bcategorymembers
- https://www.mediawiki.org/wiki/API:Categorymembers

Parameters

- **page** – *WikipediaPage*
- **kwargs** – parameters used in API call

Returns pages in given category

extracts (*page: wikipediaapi.WikipediaPage*, ***kwargs*) → str
Returns summary of the page with respect to parameters

Parameter *exsectionformat* is taken from *Wikipedia* constructor.

API Calls for parameters:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bextracts
- https://www.mediawiki.org/wiki/Extension:TextExtracts#API

Example:

```
import wikipediaapi
wiki = wikipediaapi.Wikipedia('en')

page = wiki.page('Python_(programming_language)')
print(wiki.extracts(page, exsentences=1))
print(wiki.extracts(page, exsentences=2))
```

Parameters

- **page** – *WikipediaPage*
- **kwargs** – parameters used in API call

Returns summary of the page

info (*page: wikipediaapi.WikipediaPage*) → wikipediaapi.WikipediaPage
https://www.mediawiki.org/w/api.php?action=help&modules=query%2Binfo https://www.mediawiki.org/wiki/API:Info

langlinks (*page: wikipediaapi.WikipediaPage*, ***kwargs*) → Dict[str, wikipediaapi.WikipediaPage]
Returns langlinks of the page with respect to parameters

API Calls for parameters:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Blanglinks
- https://www.mediawiki.org/wiki/API:Langlinks

Parameters

- **page** – *WikipediaPage*
- **kwargs** – parameters used in API call

Returns links to pages in other languages

links(*page: wikipediaapi.WikipediaPage*, ***kwargs*) → Dict[str, wikipediaapi.WikipediaPage]
Returns links to other pages with respect to parameters

API Calls for parameters:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Blinks
- https://www.mediawiki.org/wiki/API:Links

Parameters

- **page** – *WikipediaPage*
- **kwargs** – parameters used in API call

Returns links to linked pages

page(*title: str*, *ns: Union[wikipediaapi.Namespace, int] = <Namespace.MAIN: 0>*, *unquote: bool = False*) → wikipediaapi.WikipediaPage
Constructs Wikipedia page with title *title*.

Creating *WikipediaPage* object is always the first step for extracting any information.

Example:

```
wiki_wiki = wikipediaapi.Wikipedia('en')
page_py = wiki_wiki.page('Python_(programming_language)')
print(page_py.title)
# Python (programming language)

wiki_hi = wikipediaapi.Wikipedia('hi')

page_hi_py = wiki_hi.article(
    title='%E0%A4%AA%E0%A4%BE%E0%A4%87%E0%A4%A5%E0%A4%A8',
    unquote=True,
)
print(page_hi_py.title)
#
```

Parameters

- **title** – page title as used in Wikipedia URL
- **ns** – WikiNamespace
- **unquote** – if true it will unquote title

Returns object representing *WikipediaPage*

class wikipediaapi.**WikipediaPage**(*wiki: wikipediaapi.Wikipedia*, *title: str*, *ns: Union[wikipediaapi.Namespace, int] = <Namespace.MAIN: 0>*, *language: str = 'en'*, *url: Optional[str] = None*)
Represents Wikipedia page.

Except properties mentioned as part of documentation, there are also these properties available:

- *fullurl* - full URL of the page
- *canonicalurl* - canonical URL of the page
- *pageid* - id of the current page

- *displaytitle* - title of the page to display
- *talkid* - id of the page with discussion

__init__(*wiki: wikipediaapi.Wikipedia*, *title: str*, *ns: Union[wikipediaapi.Namespace, int] = <Namespace.MAIN: 0>*, *language: str = 'en'*, *url: Optional[str] = None*) → None
Initialize self. See help(type(self)) for accurate signature.

__repr__()
Return repr(self).

backlinks
Returns all pages linking to the current page.

This is wrapper for:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bbacklinks
- https://www.mediawiki.org/wiki/API:Backlinks

Returns PagesDict

categories
Returns categories associated with the current page.

This is wrapper for:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bcategories
- https://www.mediawiki.org/wiki/API:Categories

Returns PagesDict

categorymembers
Returns all pages belonging to the current category.

This is wrapper for:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bcategorymembers
- https://www.mediawiki.org/wiki/API:Categorymembers

Returns PagesDict

exists() → bool
Returns *True* if the current page exists, otherwise *False*.

Returns if current page existst or not

langlinks
Returns all language links to pages in other languages.

This is wrapper for:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Blanglinks
- https://www.mediawiki.org/wiki/API:Langlinks

Returns PagesDict

language
Returns language of the current page.

Returns language

links
Returns all pages linked from the current page.

This is wrapper for:

- https://www.mediawiki.org/w/api.php?action=help&modules=query%2Blinks
- https://www.mediawiki.org/wiki/API:Links

Returns `PagesDict`

namespace
Returns namespace of the current page.

Returns namespace

section_by_title(*title: str*) → Optional[wikipediaapi.WikipediaPageSection]
Returns last section of the current page with given *title*.

Parameters **title** – section title

Returns *WikipediaPageSection*

sections
Returns all sections of the curent page.

Returns List of *WikipediaPageSection*

sections_by_title(*title: str*) → List[wikipediaapi.WikipediaPageSection]
Returns all section of the current page with given *title*.

Parameters **title** – section title

Returns *WikipediaPageSection*

summary
Returns summary of the current page.

Returns summary

text
Returns text of the current page.

Returns text of the current page

title
Returns title of the current page.

Returns title

class wikipediaapi.**WikipediaPageSection**(*wiki: wikipediaapi.Wikipedia*, *title: str*, *level: int = 0*, *text: str = ''*)
WikipediaPageSection represents section in the page.

__init__(*wiki: wikipediaapi.Wikipedia*, *title: str*, *level: int = 0*, *text: str = ''*) → None
Constructs WikipediaPageSection.

__repr__()
Return repr(self).

full_text(*level: int = 1*) → str
Returns text of the current section as well as all its subsections.

Parameters **level** – indentation level

Returns text of the current section as well as all its subsections

level
Returns indentation level of the current section.

Returns indentation level of the current section

section_by_title(*title: str*) → Optional[wikipediaapi.WikipediaPageSection]
Returns subsections of the current section with given title.

Parameters **title** – title of the subsection

Returns subsection if it exists

sections
Returns subsections of the current section.

Returns subsections of the current section

text
Returns text of the current section.

Returns text of the current section

title
Returns title of the current section.

Returns title of the current section

class wikipediaapi.**ExtractFormat**
Represents extraction format.

WIKI = 1
Allows recognizing subsections

Example: https://goo.gl/PScNVV

HTML = 2
Alows retrieval of HTML tags

Example: https://goo.gl/1Jwwpr

class wikipediaapi.**Namespace**
Represents namespace in Wikipedia

You can gen list of possible namespaces here:

- https://en.wikipedia.org/wiki/Wikipedia:Namespace
- https://en.wikipedia.org/wiki/Wikipedia:Namespace#Programming

Currently following namespaces are supported:

MAIN = 0

TALK = 1

USER = 2

USER_TALK = 3

WIKIPEDIA = 4

WIKIPEDIA_TALK = 5

FILE = 6

FILE_TALK = 7

MEDIAWIKI = 8

```
MEDIAWIKI_TALK = 9
TEMPLATE = 10
TEMPLATE_TALK = 11
HELP = 12
HELP_TALK = 13
CATEGORY = 14
CATEGORY_TALK = 15
PORTAL = 100
PORTAL_TALK = 101
PROJECT = 102
PROJECT_TALK = 103
REFERENCE = 104
REFERENCE_TALK = 105
BOOK = 108
BOOK_TALK = 109
DRAFT = 118
DRAFT_TALK = 119
EDUCATION_PROGRAM = 446
EDUCATION_PROGRAM_TALK = 447
TIMED_TEXT = 710
TIMED_TEXT_TALK = 711
MODULE = 828
MODULE_TALK = 829
GADGET = 2300
GADGET_TALK = 2301
GADGET_DEFINITION = 2302
GADGET_DEFINITION_TALK = 2303
```

Python Module Index

W

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

P

R

S

T

U

W