
mediawiki Documentation

Release 0.7.0

Tyler Barrus

Jan 01, 2020

Contents

1	Installation	3
2	Documentation	5
3	Automated Tests	7
4	Quickstart	9
5	Changelog	11
6	License	13
6.1	MediaWiki Documentation	13
6.2	MediaWiki Quickstart	27
6.3	Indices and tables	31
7	Read More	33
	Python Module Index	35
	Index	37

mediawiki is a python wrapper and parser for the MediaWiki API. The goal is to allow users to quickly and efficiently pull data from the MediaWiki site of their choice instead of worrying about dealing directly with the API. As such, it does not force the use of a particular MediaWiki site. It defaults to [Wikipedia](#) but other MediaWiki sites can also be used.

MediaWiki wraps the [MediaWiki API](#) so you can focus on *leveraging* your favorite MediaWiki site's data, not getting it. Please check out the code on [github!](#)

Note: this library was designed for ease of use and simplicity. If you plan on doing serious scraping, automated requests, or editing, please look into [Pywikibot](#) which has a larger API, advanced rate limiting, and other features so we may be considerate of the MediaWiki infrastructure. Pywikibot has also other extra features such as support for Wikibase (that runs Wikidata).

CHAPTER 1

Installation

Pip Installation:

```
$ pip install pymediawiki
```

To install from source:

To install mediawiki, simply clone the [repository on GitHub](#), then run from the folder:

```
$ python setup.py install
```

mediawiki supports python versions 3.5 - 3.9

For *python 2.7* support, install [release 0.6.7](#)

```
$ pip install pymediawiki==0.6.7
```


CHAPTER 2

Documentation

Documentation of the latest release is hosted on readthedocs.io

To build the documentation yourself run:

```
$ pip install sphinx
$ cd docs/
$ make html
```


CHAPTER 3

Automated Tests

To run automated tests, one must simply run the following command from the downloaded folder:

```
$ python setup.py test
```


CHAPTER 4

Quickstart

Import mediawiki and run a standard search against Wikipedia:

```
>>> from mediawiki import MediaWiki
>>> wikipedia = MediaWiki()
>>> wikipedia.search('washington')
```

Run more advanced searches:

```
>>> wikipedia.opensearch('washington')
>>> wikipedia.allpages('a')
>>> wikipedia.geosearch(title='washington, d.c.')
>>> wikipedia.geosearch(latitude='0.0', longitude='0.0')
>>> wikipedia.prefixsearch('arm')
>>> wikipedia.random(pages=10)
```

Pull a MediaWiki page and some of the page properties:

```
>>> p = wikipedia.page('Chess')
>>> p.title
>>> p.summary
>>> p.categories
>>> p.images
>>> p.links
>>> p.langlinks
```

See the [documentation](#) for more examples!

CHAPTER 5

Changelog

Please see the [changelog](#) for a list of all changes.

MIT licensed. See the [LICENSE](#) file for full details.

6.1 MediaWiki Documentation

Here you can find the full developer API for the mediawiki project.

6.1.1 Functions and Classes

MediaWiki

```
class mediawiki.MediaWiki (url='https://{lang}.wikipedia.org/w/api.php',  
                           lang='en', timeout=15.0, rate_limit=False,  
                           rate_limit_wait=datetime.timedelta(microseconds=50000),  
                           cat_prefix='Category', user_agent=None, username=None, password=None, proxies=None)
```

MediaWiki API Wrapper Instance

Parameters

- **url** (*str*) – API URL of the MediaWiki site; defaults to Wikipedia
- **lang** (*str*) – Language of the MediaWiki site; used to help change API URL
- **timeout** (*float*) – HTTP timeout setting; None means no timeout
- **rate_limit** (*bool*) – Use rate limiting to limit calls to the site
- **rate_limit_wait** (*timedelta*) – Amount of time to wait between requests
- **cat_prefix** (*str*) – The prefix for categories used by the mediawiki site; defaults to Category (en)

- **user_agent** (*str*) – The user agent string to use when making requests; defaults to a library version but per the MediaWiki API documentation it recommends setting a unique one and not using the library’s default user-agent string
- **username** (*str*) – The username to use to log into the MediaWiki
- **password** (*str*) – The password to use to log into the MediaWiki
- **proxies** (*str*) – A dictionary of specific proxies to use in the Requests library.

login (*username, password*)
Login as specified user

Parameters

- **username** (*str*) – The username to log in with
- **password** (*str*) – The password for the user
- **strict** (*bool*) – *True* to throw an error on failure

Returns *True* if successfully logged in; *False* otherwise

Return type `bool`

Raises `mediawiki.exceptions.MediaWikiLoginError()` – if unable to login

Note: Per the MediaWiki API, one should use the *bot password*; see <https://www.mediawiki.org/wiki/API:Login> for more information

suggest (*query*)
Gather suggestions based on the provided title or `None` if no suggestions found

Parameters **query** (*str*) – Page title

Returns Suggested page title or `None` if no suggestion found

Return type `String` or `None`

search (*query, results=10, suggestion=False*)
Search for similar titles

Parameters

- **query** (*str*) – Page title
- **results** (*int*) – Number of pages to return
- **suggestion** (*bool*) – Use suggestion

Returns tuple (list results, suggestion) if suggestion is **True**; list of results otherwise

Return type tuple or list

Note: Could add ability to continue past the limit of 500

allpages (*query="", results=10*)
Request all pages from mediawiki instance

Parameters

- **query** (*str*) – Search string to use for pulling pages
- **results** (*int*) – The number of pages to return

Returns The pages that meet the search query

Return type list

Note: Could add ability to continue past the limit of 500

summary (*title*, *sentences=0*, *chars=0*, *auto_suggest=True*, *redirect=True*)

Get the summary for the title in question

Parameters

- **title** (*str*) – Page title to summarize
- **sentences** (*int*) – Number of sentences to return in summary
- **chars** (*int*) – Number of characters to return in summary
- **auto_suggest** (*bool*) – Run auto-suggest on title before summarizing
- **redirect** (*bool*) – Use page redirect on title before summarizing

Returns The summarized results of the page

Return type str

Note: Precedence for parameters: sentences then chars; if both are 0 then the entire first section is returned

geosearch (*latitude=None*, *longitude=None*, *radius=1000*, *title=None*, *auto_suggest=True*, *results=10*)

Search for pages that relate to the provided geocoords or near the page

Parameters

- **latitude** (*Decimal or None*) – Latitude geocoord; must be coercible to decimal
- **longitude** (*Decimal or None*) – Longitude geocoord; must be coercible to decimal
- **radius** (*int*) – Radius around page or geocoords to pull back; in meters
- **title** (*str*) – Page title to use as a geocoordinate; this has precedence over lat/long
- **auto_suggest** (*bool*) – Auto-suggest the page title
- **results** (*int*) – Number of pages within the radius to return

Returns A listing of page titles

Return type list

Note: The Geosearch API does not support pulling more than the maximum of 500

Raises `ValueError` – If either the passed latitude or longitude are not coercible to a `Decimal`

prefixsearch (*prefix*, *results=10*)

Perform a prefix search using the provided prefix string

Parameters

- **prefix** (*str*) – Prefix string to use for search

- **results** (*int*) – Number of pages with the prefix to return

Returns List of page titles

Return type list

Note: Per the documentation: “The purpose of this module is similar to `action=opensearch`: to take user input and provide the best-matching titles. Depending on the search engine backend, this might include typo correction, redirect avoidance, or other heuristics.”

Note: Could add ability to continue past the limit of 500

opensearch (*query, results=10, redirect=True*)

Execute a MediaWiki opensearch request, similar to search box suggestions and conforming to the OpenSearch specification

Parameters

- **query** (*str*) – Title to search for
- **results** (*int*) – Number of pages within the radius to return
- **redirect** (*bool*) – If **False** return the redirect itself, otherwise resolve redirects

Returns List of results that are stored in a tuple (Title, Summary, URL)

Return type List

Note: The Opensearch API does not support pulling more than the maximum of 500

Raises:

categorymembers (*category, results=10, subcategories=True*)

Get information about a category: pages and subcategories

Parameters

- **category** (*str*) – Category name
- **results** (*int*) – Number of result
- **subcategories** (*bool*) – Include subcategories (**True**) or not (**False**)

Returns Either a tuple ([pages], [subcategories]) or just the list of pages

Return type Tuple or List

Note: Set results to **None** to get all results

categorytree (*category, depth=5*)

Generate the Category Tree for the given categories

Parameters

- **category** (*str or list of strings*) – Category name(s)
- **depth** (*int*) – Depth to traverse the tree

Returns Category tree structure

Return type dict

Note: Set depth to **None** to get the whole tree

Note: Return Data Structure: Subcategory contains the same recursive structure

```
>>> {
    'category': {
        'depth': Number,
        'links': list,
        'parent-categories': list,
        'sub-categories': dict
    }
}
```

New in version 0.3.10.

clear_memoized()

Clear memoized (cached) values

page (*title=None, pageid=None, auto_suggest=True, redirect=True, preload=False*)

Get MediaWiki page based on the provided title or pageid

Parameters

- **title** (*str*) – Page title
- **pageid** (*int*) – MediaWiki page identifier
- **auto-suggest** (*bool*) – **True:** Allow page title auto-suggest
- **redirect** (*bool*) – **True:** Follow page redirects
- **preload** (*bool*) – **True:** Load most page properties

Raises `ValueError` – when title is blank or None and no pageid is provided

Raises `mediawiki.exceptions.PageError()` – if page does not exist

Note: Title takes precedence over pageid if both are provided

random (*pages=1*)

Request a random page title or list of random titles

Parameters **pages** (*int*) – Number of random pages to return

Returns A list of random page titles or a random page title if pages = 1

Return type list or int

set_api_url (*api_url='https://{lang}.wikipedia.org/w/api.php', lang='en', username=None, password=None*)

Set the API URL and language

Parameters

- **api_url** (*str*) – API URL to use
- **lang** (*str*) – Language of the API URL

- **username** (*str*) – The username, if needed, to log into the MediaWiki site
- **password** (*str*) – The password, if needed, to log into the MediaWiki site

Raises `mediawiki.exceptions.MediaWikiAPIURL_Error()` – if the url is not a valid MediaWiki site or login fails

wiki_request (*params*)

Make a request to the MediaWiki API using the given search parameters

Parameters **params** (*dict*) – Request parameters

Returns A parsed dict of the JSON response

Note: Useful when wanting to query the MediaWiki site for some value that is not part of the wrapper API

api_url

API URL of the MediaWiki site

Note: Not settable; See `mediawiki.MediaWiki.set_api_url()`

Type str

api_version

API Version of the MediaWiki site

Note: Not settable

Type str

extensions

Extensions installed on the MediaWiki site

Note: Not settable

Type list

language

The API URL language, if possible this will update the API URL

Note: Use correct language titles with the updated API URL

Note: Some API URLs do not encode language; unable to update if this is the case

Type str

memoized

Return the memoize cache

Note: Not settable; see `mediawiki.MediaWiki.clear_memoized()`

Type dict

rate_limit

Turn on or off Rate Limiting

Type bool

rate_limit_min_wait

Time to wait between calls

Note: Only used if `rate_limit` is **True**

Type timedelta

refresh_interval

The interval at which the memoize cache is to be refresh

Type int

supported_languages

All supported language prefixes on the MediaWiki site

Note: Not Settable

Type dict

timeout

Response timeout for API requests

Note: Use **None** for no response timeout

Type float

user_agent

User agent string

Note: If using in as part of another project, this should be changed

Type str

version

The version of the pymediawiki library

Note: Not settable

Type `str`

MediaWikiPage

```
class mediawiki.MediaWikiPage(mediawiki, title=None, pageid=None, redirect=True,  
                               preload=False, original_title="")
```

MediaWiki Page Instance

Parameters

- **mediawiki** (`MediaWiki`) – MediaWiki class object from which to pull
- **title** (`str`) – Title of page to retrieve
- **pageid** (`int`) – MediaWiki site pageid to retrieve
- **redirect** (`bool`) – **True:** Follow redirects
- **preload** (`bool`) – **True:** Load most properties after getting page
- **original_title** (`str`) – Not to be used from the caller; used to help follow redirects

Raises `mediawiki.exceptions.PageError()` – if page provided does not exist

Raises `mediawiki.exceptions.DisambiguationError()` – if page provided is a disambiguation page

Raises `mediawiki.exceptions.RedirectError()` – if redirect is **False** and the pageid or title provided redirects to another page

Warning: This should never need to be used directly! Please use `mediawiki.MediaWiki.page()`

parse_section_links (`section_title`)

Parse all links within a section

Parameters **section_title** (`str`) – Name of the section to pull or, if `None` is provided, the links between the main heading and the first section

Returns List of (title, url) tuples

Return type list

Note: Use **None** to pull the links from the header section

Note: Returns **None** if section title is not found

Note: Side effect is to also pull the html which can be slow

Note: This is a parsing operation and not part of the standard API

section (`section_title`)

Plain text section content

Parameters `section_title` (*str*) – Name of the section to pull or None for the header section

Returns The content of the section

Return type str

Note: Use **None** if the header section is desired

Note: Returns **None** if section title is not found; only text between title and next section or sub-section title is returned

Note: Side effect is to also pull the content which can be slow

Note: This is a parsing operation and not part of the standard API

summarize (*sentences=0, chars=0*)

Summarize page either by number of sentences, chars, or first section (**default**)

Parameters

- **sentences** (*int*) – Number of sentences to use in summary (first *x* sentences)
- **chars** (*int*) – Number of characters to use in summary (first *x* characters)

Returns The summary of the MediaWiki page

Return type str

Note: Precedence for parameters: sentences then chars; if both are 0 then the entire first section is returned

backlinks

Pages that link to this page

Note: Not settable

Type list

categories

Non-hidden categories on the page

Note: Not settable

Type list

content

The page content in text format

Note: Not settable

Note: Side effect is to also get revision_id and parent_id

Type str

coordinates

GeoCoordinates of the place referenced; results in lat/long tuple or None if no geocoordinates present

Note: Not settable

Note: Requires the GeoData extension to be installed

Type Tuple

hatnotes

Parse hatnotes from the HTML

Note: Not settable

Note: Side effect is to also pull the html which can be slow

Note: This is a parsing operation and not part of the standard API

Type list

html

HTML representation of the page

Note: Not settable

Warning: This can be slow for very large pages

Type str

images

Images on the page

Note: Not settable

Type list

langlinks

Names of the page in other languages for which page is where the key is the language code and the page name is the name of the page in that language.

Note: Not settable

Note: list of all language links from the provided pages to other languages according to: <https://www.mediawiki.org/wiki/API:Langlinks>

Type dict

links

List of all MediaWiki page links on the page

Note: Not settable

Type list

logos

Parse images within the infobox signifying either the main image or logo

Note: Not settable

Note: Side effect is to also pull the html which can be slow

Note: This is a parsing operation and not part of the standard API

Type list

parent_id

The parent id of the page

Note: Not settable

Note: Side effect is to also get content and revision_id

Type int

redirects

List of all redirects to this page; **i.e.**, the titles listed here will redirect to this page title

Note: Not settable

Type list

references

External links, or references, listed anywhere on the MediaWiki page .. note:: Not settable

Note May include external links within page that are not technically cited anywhere

Type list

revision_id

The current revision id of the page

Note: Not settable

Note: Side effect is to also get content and parent_id

Type int

sections

Table of contents sections

Note: Not settable

Type list

summary

Default page summary

Note: Not settable

Type str

table_of_contents

Dictionary of sections and sub-sections

Note: Leaf nodes are empty OrderedDict objects

Note: Not Settable

Type OrderedDict

wikitext

Wikitext representation of the page

Note: Not settable

Type str

6.1.2 Exceptions

MediaWiki Exceptions

exception mediawiki.exceptions.**DisambiguationError** (*title*, *may_refer_to*, *url*, *details=None*)

Exception raised when a page resolves to a Disambiguation page

Parameters

- **title** (*str*) – Title that resulted in a disambiguation page
- **may_refer_to** (*list*) – List of possible titles
- **url** (*str*) – Full URL to the disambiguation page
- **details** (*dict*) – A list of dictionaries with more information of possible results

Note: *options* only includes titles that link to valid MediaWiki pages

details

The details of the proposed non-disambigous pages

Type list

options

The list of possible page titles

Type list

title

The title of the page

Type str

url

The url, if possible, of the disambiguation page

Type str

exception mediawiki.exceptions.**HTTPTimeoutError** (*query*)

Exception raised when a request to the Mediawiki site times out.

Parameters **query** (*str*) – The query that timed out

query

The query that timed out

Type str

exception mediawiki.exceptions.**MediaWikiAPIURLError** (*api_url*)

Exception raised when the MediaWiki server does not support the API

Parameters **api_url** (*str*) – The API URL that was not recognized

api_url

The api url that raised the exception

Type str

exception mediawiki.exceptions.**MediaWikiBaseException** (*message*)

Base MediaWikiException

Parameters **message** – The message of the exception

message

The MediaWiki exception message

Type str

exception mediawiki.exceptions.**MediaWikiCategoryTreeError** (*category*)

Exception when the category tree is unable to complete for an unknown reason

Parameters **category** (*str*) – The category that threw an exception

category

The category that threw an exception during category tree generation

Type str

exception mediawiki.exceptions.**MediaWikiException** (*error*)

MediaWiki Exception Class

Parameters **error** (*str*) – The error message that the MediaWiki site returned

error

The error message that the MediaWiki site returned

Type str

exception mediawiki.exceptions.**MediaWikiGeoCoordError** (*error*)

Exceptions to handle GeoData exceptions

Parameters **error** (*str*) – Error message from the MediaWiki site related to GeoCoordinates

error

The error that was thrown when pulling GeoCoordinates

Type str

exception mediawiki.exceptions.**MediaWikiLoginError** (*error*)

Exception raised when unable to login to the MediaWiki site

Parameters **error** (*str*) – The error message that the MediaWiki site returned

error

The error message that the MediaWiki site returned

Type str

exception mediawiki.exceptions.**PageError** (*title=None, pageid=None*)

Exception raised when no MediaWiki page matched a query

Parameters

- **title** (*str*) – Title of the page
- **pageid** (*int*) – MediaWiki page id of the page

pageid

The page id that caused the page error

Type int

title

The title that caused the page error

Type str

exception mediawiki.exceptions.**RedirectError** (*title*)

Exception raised when a page title unexpectedly resolves to a redirect

Parameters **title** (*str*) – Title of the page that redirected

Note: This should only occur if both `auto_suggest` and `redirect` are set to **False**

title

The title that was redirected

Type str

6.1.3 Indices and tables

- *MediaWiki*
- *MediaWiki Quickstart*
- genindex
- modindex
- search

6.2 MediaWiki Quickstart

Quickly get started using the *mediawiki* python library. This page is designed to help users understand the basics of using the *mediawiki* library.

To understand all possible parameters for each function and properties, please see *MediaWiki Documentation*.

6.2.1 Install

Using pip

```
$ pip install pymediawiki
```

From source

Begin by installing pymediawiki: simply clone the [repository on GitHub](#), then run the following command from the extracted folder:

```
$ python setup.py install
```

6.2.2 Setup

Setting up the library is as easy as:

```
>>> from mediawiki import MediaWiki
>>> wikipedia = MediaWiki()
```

6.2.3 Change API URL

To change the API URL, one can either set the url parameter

```
>>> from mediawiki import MediaWiki
>>> asoiaf = MediaWiki(url='http://awoiaf.westeros.org/api.php')
```

Or one can update an already setup MediaWiki object:

```
>>> wikipedia.set_api_url('http://awoiaf.westeros.org/api.php')
```

6.2.4 Set the User-Agent String

Per the [MediaWiki API Etiquette](#) it is recommended to not use a library's default user-agent string. Therefore, it is easy to change the *user-agent* string either during initialization or by setting the `user_agent` property:

```
>>> from mediawiki import MediaWiki
>>> wikipedia = MediaWiki(user_agent='pyMediaWiki-User-Agent-String')
>>>
>>> # Or reset it!
>>> wikipedia.user_agent = 'my-new-user-agent-string'
```

6.2.5 Searching

To search the MediaWiki site, it is as easy as calling one of the search functions: *random*, *search*, *geosearch*, *opensearch*, or *prefixsearch*

random

Get a random page:

```
>>> wikipedia.random(pages=3)
# ['Sutton House, London', 'Iolous violacea', 'Epigenetics & Chromatin']
```


search

Search for the provided title:

```
>>> wikipedia.search('washington', results=3)
# ['Washington', 'Washington, D.C.', 'List of Governors of Washington']
```

allpages

Search for the provided title:

```
>>> wikipedia.allpages('a', results=3)
# ['A', 'A!', 'A! (Alexa Feser album)']
```

geosearch

Search based on geocoords (latitude/longitude):

```
>>> wikipedia.geosearch(latitude=0.0, longitude=0.0)
# ['Null Island', 'Mirdif 35']
```

opensearch

Search using the OpenSearch specification:

```
>>> wikipedia.opensearch('new york', results=1)
# [('New York', 'New York is a state in the Northeastern United States
and is the 27th-most extensive, fourth-most populous, and seventh-most
densely populated U.S.', 'https://en.wikipedia.org/wiki/New_York')]
```

prefixsearch

Search for pages whose title has the defined prefix:

```
>>> wikipedia.prefixsearch('ba', results=5)
# ['Ba', 'Barack Obama', 'Baseball', "Bahá'í Faith", 'Basketball']
```

6.2.6 Page

Load and access information from full MediaWiki pages. Load the page using a title or page id and then access individual properties:

Initialize Page

Initializing a page is easily accomplished in one line of code

```
>>> p = wikipedia.page('grid compass')
```

title

The page title

```
>>> p.title
# 'Grid Compass'
```

pageid

The page id of the page

```
>>> p.pageid
# 3498511
```

revision_id

The revision id of the page

```
>>> p.revision_id
# 740685101
```

parent_id

The parent id of the page

```
>>> p.parent_id
# 740682666
```

links

Links to other MediaWiki pages

```
>>> p.links
# ['Astronaut', 'Bill Moggridge', 'CP/M', 'Central processing unit',
'Dynabook', 'Electroluminescent display', 'FTP', 'Flip (form)',
'GRiD Systems Corporation', 'GRiD-OS', 'Gavilan SC', 'Grid compass',
'Hard drive', 'IEEE-488', 'Industrial design', 'Intel 8086',
'John Oliver Creighton', 'Kilobyte', 'Laptop computer',
'Magnetic bubble memory', 'Modem', 'NASA', 'Operating system',
'Osborne 1', 'Paratrooper', 'Patent rights', 'Perfect (film)',
'Portable computer', 'RadioShack', 'Riptide (American TV series)',
'STS-51-G', 'Sharp PC-5000', 'Space Shuttle Discovery',
'Tandy Corporation', 'U.S. government', 'United Kingdom',
'United States Army Special Forces', 'Xerox PARC']
```

Other Properties

Other properties for a page include:

- *content*
- *html*

- *images*
- *references*
- *categories*
- *coordinates*
- *redirects*
- *backlinks*
- *langlinks*
- *summary*
- *sections*
- *logos*
- *hatnotes*

Summarize

Summarize a page using additional parameters:

```
>>> p.summarize(chars=50)
# The Grid Compass (written GRiD by its manufacturer...
```

6.3 Indices and tables

- *MediaWiki*
- *MediaWiki Documentation*
- *genindex*
- *modindex*
- *search*

CHAPTER 7

Read More

- *MediaWiki Documentation*
- *MediaWiki Quickstart*
- `genindex`
- `modindex`
- `search`

m

`mediawiki.exceptions`, 25

A

allpages() (*mediawiki.MediaWiki method*), 14
 api_url (*mediawiki.exceptions.MediaWikiAPIURLError attribute*), 26
 api_url (*mediawiki.MediaWiki attribute*), 18
 api_version (*mediawiki.MediaWiki attribute*), 18

B

backlinks (*mediawiki.MediaWikiPage attribute*), 21

C

categories (*mediawiki.MediaWikiPage attribute*), 21
 category (*mediawiki.exceptions.MediaWikiCategoryTreeError attribute*), 26
 categorymembers() (*mediawiki.MediaWiki method*), 16
 categorytree() (*mediawiki.MediaWiki method*), 16
 clear_memoized() (*mediawiki.MediaWiki method*), 17
 content (*mediawiki.MediaWikiPage attribute*), 21
 coordinates (*mediawiki.MediaWikiPage attribute*), 22

D

details (*mediawiki.exceptions.DisambiguationError attribute*), 25
 DisambiguationError, 25

E

error (*mediawiki.exceptions.MediaWikiException attribute*), 26
 error (*mediawiki.exceptions.MediaWikiGeoCoordError attribute*), 26
 error (*mediawiki.exceptions.MediaWikiLoginError attribute*), 26
 extensions (*mediawiki.MediaWiki attribute*), 18

G

geosearch() (*mediawiki.MediaWiki method*), 15

H

hatnotes (*mediawiki.MediaWikiPage attribute*), 22
 html (*mediawiki.MediaWikiPage attribute*), 22
 HTTPTimeoutError, 25

I

images (*mediawiki.MediaWikiPage attribute*), 22

L

langlinks (*mediawiki.MediaWikiPage attribute*), 23
 language (*mediawiki.MediaWiki attribute*), 18
 links (*mediawiki.MediaWikiPage attribute*), 23
 login() (*mediawiki.MediaWiki method*), 14
 logos (*mediawiki.MediaWikiPage attribute*), 23

M

MediaWiki (*class in mediawiki*), 13
 mediawiki.exceptions (*module*), 25
 MediaWikiAPIURLError, 26
 MediaWikiBaseException, 26
 MediaWikiCategoryTreeError, 26
 MediaWikiException, 26
 MediaWikiGeoCoordError, 26
 MediaWikiLoginError, 26
 MediaWikiPage (*class in mediawiki*), 20
 memoized (*mediawiki.MediaWiki attribute*), 18
 message (*mediawiki.exceptions.MediaWikiBaseException attribute*), 26

O

opensearch() (*mediawiki.MediaWiki method*), 16
 options (*mediawiki.exceptions.DisambiguationError attribute*), 25

P

page() (*mediawiki.MediaWiki method*), 17
 PageError, 26
 pageid (*mediawiki.exceptions.PageError attribute*), 27
 parent_id (*mediawiki.MediaWikiPage attribute*), 23

`parse_section_links()` (*mediawiki.MediaWikiPage method*), 20
`prefixsearch()` (*mediawiki.MediaWiki method*), 15

Q

`query` (*mediawiki.exceptions.HTTPTimeoutError attribute*), 25

R

`random()` (*mediawiki.MediaWiki method*), 17
`rate_limit` (*mediawiki.MediaWiki attribute*), 19
`rate_limit_min_wait` (*mediawiki.MediaWiki attribute*), 19
`RedirectError`, 27
`redirects` (*mediawiki.MediaWikiPage attribute*), 24
`references` (*mediawiki.MediaWikiPage attribute*), 24
`refresh_interval` (*mediawiki.MediaWiki attribute*), 19
`revision_id` (*mediawiki.MediaWikiPage attribute*), 24

S

`search()` (*mediawiki.MediaWiki method*), 14
`section()` (*mediawiki.MediaWikiPage method*), 20
`sections` (*mediawiki.MediaWikiPage attribute*), 24
`set_api_url()` (*mediawiki.MediaWiki method*), 17
`suggest()` (*mediawiki.MediaWiki method*), 14
`summarize()` (*mediawiki.MediaWikiPage method*), 21
`summary` (*mediawiki.MediaWikiPage attribute*), 24
`summary()` (*mediawiki.MediaWiki method*), 15
`supported_languages` (*mediawiki.MediaWiki attribute*), 19

T

`table_of_contents` (*mediawiki.MediaWikiPage attribute*), 24
`timeout` (*mediawiki.MediaWiki attribute*), 19
`title` (*mediawiki.exceptions.DisambiguationError attribute*), 25
`title` (*mediawiki.exceptions.PageError attribute*), 27
`title` (*mediawiki.exceptions.RedirectError attribute*), 27

U

`url` (*mediawiki.exceptions.DisambiguationError attribute*), 25
`user_agent` (*mediawiki.MediaWiki attribute*), 19

V

`version` (*mediawiki.MediaWiki attribute*), 19

W

`wiki_request()` (*mediawiki.MediaWiki method*), 18
`wikitext` (*mediawiki.MediaWikiPage attribute*), 25