

---

# **MediaWikiAPI Documentation**

***Release 1.0***

**Taras Lehinevych**

**Jan 14, 2018**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Quickstart</b>	<b>5</b>
<b>3</b>	<b>Changelog</b>	<b>7</b>
3.1	Changelog . . . . .	7
<b>4</b>	<b>API</b>	<b>9</b>
4.1	MediaWikiAPI Documentation . . . . .	9
<b>5</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



Release v1.1.



# CHAPTER 1

---

## Installation

---

Start using wikipedia for Python in less than 5 minutes! MediaWikiAPI is compatible with Python 3 and Python 2.7. If you are looking for the the full developer API, see [MediaWikiAPI Documentation](#).

Begin by installing wikipedia:

```
$ pip install wikipedia
```

As alternative you can use the source code from [Github](#).





## CHAPTER 2

---

### Quickstart

---

Now let's use the MediaWikiAPI. First you need to import the package and create MediaWikiAPI class. In order to use search and suggestion call the corresponding methods search and suggest:

```
>>> from mediawikiapi import MediaWikiAPI
>>> mediawikiapi = MediaWikiAPI()
>>> mediawikiapi.search("Barack")
[u'Barak (given name)', u'Barack Obama', u'Barack (brandy)', u'Presidency of_
↳Barack Obama', u'Family of Barack Obama', u'First inauguration of Barack Obama', u
↳'Barack Obama presidential campaign, 2008', u'Barack Obama, Sr.', u'Barack Obama_
↳citizenship conspiracy theories', u'Presidential transition of Barack Obama']

>>> mediawikiapi.suggest("Barak Obama") # returns the suggested Wikipedia title for a_
↳query or None
u'Barack Obama'
```

We can also get fewer or more results by using the results kwarg:

```
>>> mediawikiapi.search("Ford", results=3)
[u'Ford Motor Company', u'Gerald Ford', u'Henry Ford']
```

To get the summary of an article, use mediawikiapi.summary:

```
>>> mediawikiapi.summary("GitHub")
2011, GitHub was the most popular open source code repository site.\nGitHub Inc. was_
↳founded in 2008 and is based in San Francisco, California.\nIn July 2012, the_
↳company received $100 million in Series A funding, primarily from Andreessen_
↳Horowitz.'
```

```
>>> mediawikiapi.summary("Apple III", sentences=1)
u'The Apple III (often rendered as Apple ///) is a business-oriented personal_
↳computer produced and released by Apple Computer that was intended as the successor_
↳to the Apple II series, but largely considered a failure in the market. '
```

mediawikiapi.page enables you to load and access data from full Wikipedia pages. Initialize with a page title (keep in mind the errors listed above), and then access most properties using property methods:

```
>>> ny = mediawikiapi.page("New York")

>>> ny.title
u'New York'

>>> ny.url
u'http://en.wikipedia.org/wiki/NewYork'

>>> ny.content
u'New York is a state in the Northeastern region of the United States. New York is_
↳the 27th-most exten'...'

>>> ny.images[0]
u'http://upload.wikimedia.org/wikipedia/commons/9/91/New_York_quarter%2C_reverse_side
↳%2C_2001.jpg'

>>> ny.links[0]
u'1790 United States Census'
```

To change the language of the Wikipedia you are accessing, use `mediawikiapi.conf.language`. Remember to search for page titles in the language that you have set, not English!:

```
>>> mediawikiapi.conf.language = "fr"

>>> print mediawikiapi.summary("Francois Hollande")
François Hollande, né le 12 août 1954 à Rouen, en Seine-Maritime, est un homme d'État_
↳français. Il est président de la République française depuis le 15 mai 2012...
```

To get a list of all possible language prefixes, try `mediawikiapi.languages()`.

For more details and configuration option check API section.

### 3.1 Changelog

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

Contents:

#### 3.1.1 Version 1.1

- **Breaking change** - add MediaWikiAPI class, now you can have more than one api access point with different configurations (Config instances). MediaWikiAPI class contains all the mediawikiapi function from version 1.0. The Config class cloud be pass as parameter during initialization.
- Support python 2
- Rename Configuration class to Config, add language field
- Config().get\_api\_url now accept language parameter
- Add timeout for requests, field in Config class called timeout (in seconds).
- Makes the pagepropsof a wikipedia page accessible [PR #147](#) from @goldsmith repo.
- Fix suggestion, issue [#108](#) by [PR #131](#) from @goldsmith repo.
- Fixed problem with hidden files in the article [PR #132](#) @goldsmith repo.
- DisambiguationError contains now information about title and url [PR #92](#) from @goldsmith repo.
- Fix issue where pageid request => redirect raises error [PR #165](#)

#### 3.1.2 Version 1.0

- Fork [Wikipedia](#)
- Add language validation for mediawikiapi.set\_lang

- Add lang title method to WikipediaPage
- Add re-usage the same requests session
- Fix installing error with version
- Fix WikipediaPage.sections
- Fix mock data
- Refactoring: seperate Language and Configuration classes

### **3.1.3 Indices and tables**

- genindex
- modindex
- search

## 4.1 MediaWikiAPI Documentation

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

Contents:

### 4.1.1 Classes

```
class mediawikiapi.MediaWikiAPI (config=None)
```

```
    __init__ (config=None)
```

```
    donate ()
```

Open up the Wikimedia donate page in your favorite browser.

```
    geosearch (latitude, longitude, title=None, results=10, radius=1000)
```

Do a wikipedia geo search for *latitude* and *longitude* using HTTP API described in <http://www.mediawiki.org/wiki/Extension:GeoData>

Arguments:

- *latitude* (float or decimal.Decimal)
- *longitude* (float or decimal.Decimal)

Keyword arguments:

- *title* - The title of an article to search for
- *results* - the maximum number of results returned
- *radius* - Search radius in meters. The value must be between 10 and 10000

```
    languages ()
```

List all the currently supported language prefixes (usually ISO language code).

Can be inputted to WikipediaPage.conf to change the Mediawiki that *wikipedia* requests results from.

Returns: dict of <prefix>: <local\_lang\_name> pairs. To get just a list of prefixes, use *wikipedia.languages().keys()*.

**page** (*title=None, pageid=None, auto\_suggest=True, redirect=True, preload=False*)

Get a WikipediaPage object for the page with title *title* or the pageid *pageid* (mutually exclusive).

Keyword arguments:

- title - the title of the page to load
- pageid - the numeric pageid of the page to load
- auto\_suggest - let Wikipedia find a valid page title for the query
- redirect - allow redirection without raising RedirectError
- preload - load content, summary, images, references, and links during initialization

**random** (*pages=1*)

Get a list of random Wikipedia article titles.

---

**Note:** Random only gets articles from namespace 0, meaning no Category, User talk, or other meta-Wikipedia pages.

---

Keyword arguments:

- pages - the number of random pages returned (max of 10)

**search** (*query, results=10, suggestion=False*)

Do a Wikipedia search for *query*.

Keyword arguments:

- results - the maximum number of results returned
- suggestion - if True, return results and suggestion (if any) in a tuple

**suggest** (*query*)

Get a Wikipedia search suggestion for *query*. Returns a string or None if no suggestion was found.

**summary** (*title, sentences=0, chars=0, auto\_suggest=True, redirect=True*)

Plain text summary of the page. .. note:: This is a convenience wrapper - auto\_suggest and redirect are enabled by default Keyword arguments: \* sentences - if set, return the first *sentences* sentences (can be no greater than 10). \* chars - if set, return only the first *chars* characters (actual text returned may be slightly longer). \* auto\_suggest - let Wikipedia find a valid page title for the query \* redirect - allow redirection without raising RedirectError

**class** mediawikiapi.**WikipediaPage** (*title=None, pageid=None, redirect=True, preload=False, original\_title="", request=None*)

Contains data from a Wikipedia page. Uses property methods to filter data from the raw HTML.

**\_\_init\_\_** (*title=None, pageid=None, redirect=True, preload=False, original\_title="", request=None*)

**categories**

List of categories of a page.

**content**

Plain text content of the page, excluding images, tables, and other data.

**coordinates**

Tuple of Decimals in the form of (lat, lon) or None

**html()**

Get full page HTML.

**Warning:** This can get pretty slow on long pages.

**images**

List of URLs of images on the page.

**lang\_title** (*lang\_code*)

Get the title in specified language code Returns None if lang code or title isn't found, otherwise returns a string with title. Raise `LanguageException` if language doesn't exists

**links**

List of titles of Wikipedia page links on a page.

---

**Note:** Only includes articles from namespace 0, meaning no Category, User talk, or other meta-Wikipedia pages.

---

**parent\_id**

Revision ID of the parent version of the current revision of this page. See `revision_id` for more information.

**references**

List of URLs of external links on a page. May include external links within page that aren't technically cited anywhere.

**revision\_id**

Revision ID of the page.

The revision ID is a number that uniquely identifies the current version of the page. It can be used to create the permalink or for other direct API calls. See [Help:Page history](#) for more information.

**section** (*section\_title*)

Get the plain text content of a section from `self.sections`. Returns None if `section_title` isn't found, otherwise returns a whitespace stripped string.

This is a convenience method that wraps `self.content`.

**Warning:** Calling `section` on a section that has subheadings will NOT return the full text of all of the subsections. It only gets the text between `section_title` and the next subheading, which is often empty.

**sections**

List of section titles from the table of contents on the page.

**summary**

Plain text summary of the page.

**class** mediawikiapi.**Language** (*language=None*)

Language used in mediawiki, if language is not defined English will be used

**\_\_init\_\_** (*language=None*)**language**

Return language

**class** mediawikiapi.**RequestSession**

Request wrapper class for request

**\_\_init\_\_** ()

Require configuration instance as argument

**request** (*params, config, language=None*)

Make a request to the Wikipedia API using the given search parameters, language and configuration

Arguments:

- *params* (dictionary)
- *config* - the configuration to be used for request

Keyword arguments:

- *language* - the wiki language

**class** mediawikiapi.**Config** (*language=None, user\_agent=None, rate\_limit=None*)

Contains global configuration

**\_\_init\_\_** (*language=None, user\_agent=None, rate\_limit=None*)

**classmethod** **donate\_url** ()

Return media wiki donate url

**get\_api\_url** (*language=None*)

Return api for specified language Arguments: \* *language* - (string or Language instance) specifying the language

**language**

Return current global language

## 4.1.2 Exceptions

Global wikipedia exception and warning classes.

**exception** mediawikiapi.exceptions.**DisambiguationError** (*title, may\_refer\_to, details=None*)

Exception raised when a page resolves to a Disambiguation page.

The *options* property contains a list of titles of Wikipedia pages that the query may refer to.

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

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

**exception** mediawikiapi.exceptions.**LanguageError** (*language*)

Exception raised when a language prefix is set which is not available

**exception** mediawikiapi.exceptions.**MediaWikiAPIException** (*error*)

Base Wikipedia exception class.

**exception** mediawikiapi.exceptions.**PageError** (*pageid=None, \*args*)

Exception raised when no Wikipedia matched a query.

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

Exception raised when a page title unexpectedly resolves to a redirect.



### 4.1.3 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)



## CHAPTER 5

---

### Indices and tables

---

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



### m

`mediawikiapi`, [9](#)

`mediawikiapi.exceptions`, [12](#)



## Symbols

`__init__()` (mediawikiapi.Config method), 12  
`__init__()` (mediawikiapi.Language method), 11  
`__init__()` (mediawikiapi.MediaWikiAPI method), 9  
`__init__()` (mediawikiapi.RequestSession method), 12  
`__init__()` (mediawikiapi.WikipediaPage method), 10

## C

categories (mediawikiapi.WikipediaPage attribute), 10  
Config (class in mediawikiapi), 12  
content (mediawikiapi.WikipediaPage attribute), 10  
coordinates (mediawikiapi.WikipediaPage attribute), 10

## D

DisambiguationError, 12  
donate() (mediawikiapi.MediaWikiAPI method), 9  
donate\_url() (mediawikiapi.Config class method), 12

## G

geosearch() (mediawikiapi.MediaWikiAPI method), 9  
get\_api\_url() (mediawikiapi.Config method), 12

## H

html() (mediawikiapi.WikipediaPage method), 10  
HTTPTimeoutError, 12

## I

images (mediawikiapi.WikipediaPage attribute), 11

## L

lang\_title() (mediawikiapi.WikipediaPage method), 11  
Language (class in mediawikiapi), 11  
language (mediawikiapi.Config attribute), 12  
language (mediawikiapi.Language attribute), 11  
LanguageError, 12  
languages() (mediawikiapi.MediaWikiAPI method), 9  
links (mediawikiapi.WikipediaPage attribute), 11

## M

MediaWikiAPI (class in mediawikiapi), 9  
mediawikiapi (module), 9  
mediawikiapi.exceptions (module), 12  
MediaWikiAPIException, 12

## P

page() (mediawikiapi.MediaWikiAPI method), 10  
PageError, 12  
parent\_id (mediawikiapi.WikipediaPage attribute), 11

## R

random() (mediawikiapi.MediaWikiAPI method), 10  
RedirectError, 12  
references (mediawikiapi.WikipediaPage attribute), 11  
request() (mediawikiapi.RequestSession method), 12  
RequestSession (class in mediawikiapi), 11  
revision\_id (mediawikiapi.WikipediaPage attribute), 11

## S

search() (mediawikiapi.MediaWikiAPI method), 10  
section() (mediawikiapi.WikipediaPage method), 11  
sections (mediawikiapi.WikipediaPage attribute), 11  
suggest() (mediawikiapi.MediaWikiAPI method), 10  
summary (mediawikiapi.WikipediaPage attribute), 11  
summary() (mediawikiapi.MediaWikiAPI method), 10

## W

WikipediaPage (class in mediawikiapi), 10