fandom-py

Release 1.5.0

Nikolaj Gade

Aug 04, 2021

CONTENTS:

fandom-py is a Python library that makes it easy to access and parse data from any fandom website.

CHAPTER

ONE

GETTING STARTED

You can start by installing the package with:

```
$ pip install fandom-py
```

1.1 Example use case

We start by importing fandom into our project.

```
import fandom
```

Now let's say we want to know about Kvothe from the *Kingkiller Chronicles* book series by Pattrick Rothfuss. We should start by setting the wiki to the *kingkiller wiki* using *fandom.set_wiki*. We can also set the wiki using a parameter each time we use a fandom function, but since we're going to be calling several different fandom functions, setting it now is probably a good idea:

```
fandom.set_wiki("kingkiller")
```

Note: Not setting a wiki will result in all your fandom function calls going to the *runescape wiki*.

We don't need to use the *fandom.set_lang* function, since the language defaults to english, which is the language we want the page in.

Now Kvothe has a lot of names, and we're not sure which one is used on his wiki page. So we can search for *Kvothe* on the wiki, and his page should probably come up first. Setting the `results` parameter to 1 ensures that we only get the top result for our search:

```
fandom.search("Kvothe", results = 1)
# [('Kvothe', 2230)]
```

Seems like Kvothe's wiki page just has the title *Kvothe*, and has the page id *2230*. We can use this information to initiate a *fandom.FandomPage* object for the wiki page. The *fandom.FandomPage* object contains all usable information about the given page.

```
page = fandom.page(title = "Kvothe")
```

Here we used the page title to initialize the page object, but we could also have used the page id:

```
page2 = fandom.page(pageid = 2230)
page2.title
# Kvothe
page == page2
# True
```

Now we have the page for Kvothe, we can gain information from it. We can start by getting the first part of the page, which we can get with the *fandom.FandomPage.summary* property:

```
page.summary
# 'Kvothe is the main character in the Kingkiller Chronicle. His name is pronounced kvōTH,
↪ much like the word quoth but beginning the same as the Yiddish term "Kvetch." '
```

Names are pretty important in *The Kingkiller Chronicles*, so we probably shouldn't be surprised that the pronounciation of his name is the first thing the wiki wants us to know.

But what if we want to know something else? Like his physical appearance. Well we can get a list of all sections on a page with the *fandom.FandomPage.sections* property:

```
page.sections
# ['Description', 'In The Chronicle', 'Early life', 'Tarbean', 'The University', 'First Term␣
↪(Spring)', 'Second Term (Summer)', 'Third Term (Fall)', 'Fourth Term (Fall)', 'Vintas', 'The␣
↪Faen Realm', 'Ademre', 'Return to the University', 'Fifth Term (Winter)', 'Sixth Term␣
↪(Spring)', 'Seventh Term (Summer)', 'The present', 'Other Names', 'Kote', 'Reshi', 'Maedre',
↪'Dulator', 'Shadicar', 'Lightfinger', 'Six-String', 'Kvothe the Bloodless', 'Kvothe the Arcane
↪', 'Kvothe Kingkiller', 'Speculation', 'Naming', 'Identity', 'Rings', 'Kvothe and Kote', 'Fan␣
↪arts', 'References']
```

All sections and subsections are included in the list, which explains the long list of names after the "other names" section title.

"Description" seems like the section with the biggest potential of telling us how he looks, so let's try getting the text from that, using the *fandom.FandomPage.section* method:

```
page.section("Description")
# "Description
# Kvothe has pale skin and green eyes, though the intensity of this color is often noted␣
↪as changing throughout the series. His eyes are similar to the description of his␣
↪mother's eyes. He has extremely red hair often compared to a flame.
# He is exceptionally intelligent, quick-witted, sharp-tongued and clever, as well as a␣
↪talented musician. He is also very curious, a quality that often gets him into trouble.
↪ He has a nasty temper, is reckless and often thoughtless.
# In the books, some evidence (mostly cover illustrations) suggest that he is left-
↪handed."
```

If we want more information about the structure of sections and subsections, we can use the *fandom.FandomPage.content* property, which returns a dict structured like this:

```
{
    'title' : 'The page title'
    'content' : 'The text before the first section starts'
    'infobox' : 'The text contained in the page's infobox'
    'sections' : [
        {
```

(continues on next page)

(continued from previous page)

```
            'title' : 'The section title'
            'content' : 'The text in the section before the first subsection starts'
            'sections' : [
                {
                    'title' : 'The subsection title'
                    'content' : 'The text in the subsection'
                },
                ...
            ]
        },
        ...
    ]
}
```

Finally, we need a good picture of Kvothe. We can use the *fandom.FandomPage.images* property for that:

```
page.images[0]
# 'https://static.wikia.nocookie.net/nameofthewind/images/6/68/The_kingkiller_chronicle_
↪kvothe_by_shilesque-d8m6yzz.jpg/revision/latest?cb=20190916153424'
```

And now you have a basic understanding of what you can do with fandom-py. Do check out the rest of the documentation if you want to know more.

CHAPTER

TWO

FANDOM PACKAGE

2.1 Submodules

2.1.1 fandom.error module

Global fandom exception and warning classes.

exception fandom.error.**FandomError**(*query*, *wiki*, *language*)
Bases: *fandom.error.FandomException*

Exception raised when the requested query can't be found

exception fandom.error.**FandomException**(*error*)
Bases: Exception

Base fandom exception class.

exception fandom.error.**HTTPTimeoutError**(*query*)
Bases: *fandom.error.FandomException*

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

exception fandom.error.**PageError**(*pageid=None*, **args*)
Bases: *fandom.error.FandomException*

Exception raised when no fandom matched a query.

exception fandom.error.**RedirectError**(*title*)
Bases: *fandom.error.FandomException*

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

exception fandom.error.**RequestError**(*url*, *params*)
Bases: *fandom.error.FandomException*

Exception raised when the request does not return usable data. Usually raised when the wiki doesn't exist in the requested language

2.1.2 FandomPage class

class fandom.**FandomPage**(*wiki*, *language*, *title=None*, *pageid=None*, *redirect=True*, *preload=False*)

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

> **Warning:** Do not manually init fandom.FandomPage. Instead call *fandom.page()*.

Variables

- **title** – The title of the page
- **pageid** – The page id of the page
- **language** – The language of the page
- **wiki** – The wiki the page is on
- **url** – The url to the page

property **content**

Text content of each section of the page, excluding images, tables, and other data. The content is returned as dict, imitating the section and subsection structure of the page.

> **Note:** If you just want the plain text of the page without the section structure, you can use FandomPage.plain_text

Returns dict

property **html**

Get full page HTML.

Returns str

property **images**

List of URLs of images on the page.

Returns list

property **plain_text**

The plain text contents of a page.

> **Note:** If you want the section and subsection structure of the page as well as the text, you can use FandomPage.content.

Returns str

property **revision_id**

Revision ID of the page.

> **Note:** 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.

Returns int

section(*section_title: str*)
Get the plain text content of a section from *self.sections*. Returns None if *section_title* isn't found, otherwise returns a str.

> **Warning:** When calling this function, subheadings in the section you asked for are part of the plain text. If you want more control of what data you get, you should use FandomPage.content

Parameters **section_title** (*str*) – The title of the section you want the text from.

Returns str

property sections
List of section titles.

Returns list

property summary
Plain text summary of the page. The summary is usually the first section up until the first newline.

Returns str

2.2 Module functions

fandom.**search**(*query: str*, *wiki: str = ''*, *language: str = ''*, *results: int = 10*)
Do a fandom search.

The search returns a list of tuples, with the page title and the page id.

Parameters

- **query** (*str*) – What to search for
- **wiki** (*str*) – The wiki to search in (defaults to the global wiki variable)
- **language** (*str*) – The language to search in (defaults to the global language variable)
- **results** (*int*) – The maximum number of results to be returned

Returns list of tuple

fandom.**summary**(*title: str*, *wiki: str = ''*, *language: str = ''*, *sentences: int = - 1*, *redirect: bool = True*)
Plain text summary of the page with the requested title. Is just an implementation of *FandomPage.summary*, but with the added functionality of requesting a specific amount of sentences.

Parameters

- **title** (*str*) – The title of the page to get the summary of
- **wiki** (*str*) – The wiki to search (defaults to the global wiki variable. If the global wiki variable is not set, defaults to "runescape")
- **language** (*str*) – The language to search in (defaults to the global language variable. If the global language variable is not set, defaults to english)
- **sentences** (*int*) – The maximum number of sentences to output. Defaults to the whole summary
- **redirect** (*bool*) – Allow redirection without raising RedirectError

fandom.**page**(*title: str = ''*, *pageid: int = - 1*, *wiki: str = ''*, *language: str = ''*, *redirect: bool = True*, *preload: bool = False*)

Get a FandomPage object for the page in the sub fandom with title or the pageid (mutually exclusive).

Parameters

- **title** (*str*) –
  - the title of the page to load
- **pageid** (*int*) – The numeric pageid of the page to load
- **wiki** (*str*) – The wiki to search (defaults to the global wiki variable. If the global wiki variable is not set, defaults to "runescape")
- **language** (*str*) – The language to search in (defaults to the global language variable. If the global language variable is not set, defaults to english)
- **redirect** (*bool*) – Allow redirection without raising RedirectError
- **preload** (*bool*) – Load content, summary, images, references, and links during initialization

fandom.**random**(*pages: int = 1*, *wiki: str = ''*, *language: str = ''*)

Get a list of random fandom article titles.

Returns the results as tuples with the title and page id.

Note: Random only gets articles from namespace 0, meaning only articles

---

Parameters

- **pages** (*int*) – the number of random pages returned (max of 10)
- **wiki** (*str*) – The wiki to search (defaults to the global wiki variable. If the global wiki variable is not set, defaults to "runescape")
- **language** (*str*) – The language to search in (defaults to the global language variable. If the global language variable is not set, defaults to english)

Returns `tuple` if the pages parameter was 1, `list` of `tuple` if it was larger

fandom.**set_lang**(*language: str*)

Sets the global language variable

Parameters **language** (*str*) – The language to set as the global language variable

fandom.**set_rate_limiting**(*rate_limit: bool*, *min_wait: int = 50*)

Enable or disable rate limiting on requests to the fandom servers. If rate limiting is not enabled, under some circumstances (depending on load on fandom, the number of requests you and other *fandom* users are making, and other factors), fandom may return an HTTP timeout error.

Note: Enabling rate limiting generally prevents that issue, but please note that HTTPTimeoutError still might be raised.

Parameters

- **rate_limit** (*bool*) – Whether to enable rate limiting or not

- **min_wait** (*int*) – If rate limiting is enabled, *min_wait* is the minimum time to wait before requests in milliseconds.

fandom.**set_user_agent**(*user_agent_string: str*)
Set the User-Agent string to be used for all requests.

Parameters **user_agent_string** (*str*) – A string specifying the User-Agent header

fandom.**set_wiki**(*wiki: str*)
Sets the global wiki variable

Parameters **wiki** (*str*) – The wiki to set as the global wiki variable

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

f

INDEX