goodreads Documentation

Release 0.1.1

Sefa Kilic

Aug 24, 2017

Contents

This package provides a Python interface for the Goodreads API. Using it, you can do pretty much anything that Goodreads allows to do with their own data.

CHAPTER 1

Dependencies

This package depends on the following packages:

- xmltodict
- requests
- rauth

They can be installed using `pip`.

```
sudo pip install -r requirements.txt
```

If you want to contribute to this package, you will need the `nose` package as well.

CHAPTER 2

Installation

To install, run the following command from the top-level package directory.

```
sudo python setup.py install
```

CHAPTER 3

Getting Started

The first thing is to request an API key from Goodreads here. Once you have it, you can create a client instance to query Goodreads.

```
from goodreads import client
gc = client.GoodreadsClient(<api_key>, <api_secret>)
```

To access some of the methods, you need OAuth for authorization.

```
gc.authenticate(<access_token>, <access_token_secret>)
```

Note that `access_token` and `access_token_secret` are different from developer key and secret. For the development step, you can call the same function with no parameters to get authorization. It will open a URL pointing a Goodreads page for OAuth permission. For your application, you can direct the user to that particular URL, ask him/her to authorize your app and save the returning `access_token` and `access_token_secret` in your database.

CHAPTER 4

Examples

This package provides a Python interface for most Goodreads API methods. Here are a few examples demonstrating how to access data on Goodreads.

Books

Let's access the first book added to Goodreads! It is the book with id 1.

```
book = gc.book(1)
```

Once you have the `GoodreadsBook` instance for the book, you can access data for the queried book.

```
>>> book.title
u'Harry Potter and the Half-Blood Prince (Harry Potter, #6)'
>>> authors = book.authors
>>> authors[0].name
u'J.K. Rowling'
>>> book.average_rating
u'4.49'
```

Authors

You can get information about an author as well.

```
>>> author = gc.author(2617)
>>> author.name
u'Jonathan Safran Foer'
>>> author.works_count
u'13'
>>> author.books
[Extremely Loud and Incredibly Close, Everything Is Illuminated, Eating Animals, Tree␣
↪of Codes, Everything is Illuminated & Extremely Loud and Incredibly Close, The␣
↪unabridged pocketbook of lightning, The Future Dictionary of America, A Convergence␣
↪of Birds: Original Fiction and Poetry Inspired by Joseph Cornell, New American␣
↪Haggadah, The Sixth Borough]
```

Users

User data can be retrieved by user id or username.

```
>>> user = gc.user(1)
>>> user.name
u'Otis Chandler'
>>> user.user_name
u'otis'
>>> user.small_image_url
u'http://d.gr-assets.com/users/1189644957p2/1.jpg'
```

Groups

Let's find a group discussing Python and get more information about it.

```
>>> g = gc.find_groups("Python")
>>> g = groups[0]
>>> g['title']
u'The Computer Scientists'
>>> group = gc.group(g['id'])
>>> group.description
u'Only for Committed Self Learners and Computer Scientists Who are Starving for
Information, and Want to Advance their Skills Through: Reading, Practicing and
Discussion Computer Science and Programming Books.'
```

Events

Goodreads API also allows to list events happening in an area.

```
>>> events = gc.list_events(21229)
>>> event = events[0]
>>> event.title
u'Books and Cocktails'
>>> event.address
u'120 N. Front St.'
>>> event.city
u'Wrightsville'
```

CHAPTER 5

Contribution

If you find an API method that is not supported by this package, feel free to create a Github issue. Also, you are more than welcome to submit a pull request for a bug fix or additional feature.

CHAPTER 6

License

MIT License

Contents:

client

request

session

book

author

user

user_status

comment

event

group

owned_book

review