
Python Rakuten Web Service Documentation

Release 0.4.3

Salem Harrache

Apr 19, 2017

Contents

1	Python Rakuten Web Service	3
1.1	Supported APIs	3
1.2	Installation	4
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	Indices and tables	13

Contents:

Python Rakuten Web Service

Unofficial Python Client for Rakuten Web Service

- Free software: MIT license
- Documentation: <https://rakuten-ws.readthedocs.io>.

Supported APIs

- Rakuten Ichiba API
- Rakuten Ichiba RMS Item API
- Rakuten Ichiba RMS Product API
- Rakuten Ichiba RMS Order API
- Rakuten Ichiba RMS Inventory API
- Rakuten Ichiba RMS Cabinet API
- Rakuten Ichiba RMS Navigation API
- Rakuten Ichiba RMS Category API
- Rakuten Books API
- Rakuten Travel API
- Rakuten Auction API
- Rakuten Kobo API
- Rakuten GORA API
- Rakuten Recipe API
- Rakuten Other APIs

Installation

Requirements:

- python >= 2.7
- python-lxml

You can install, upgrade, uninstall rakuten-ws with these commands:

```
$ pip install [--user] rakuten-ws
$ pip install [--user] --upgrade rakuten-ws
$ pip uninstall rakuten-ws
```


CHAPTER 2

Installation

Requirements:

- python \geq 2.7

You can install, upgrade, uninstall rakuten-ws with these commands:

```
$ pip install [--user] rakuten-ws
$ pip install [--user] --upgrade rakuten-ws
$ pip uninstall rakuten-ws
```

Or from git (last development version):

```
$ pip install git+https://github.com/alexandriagroup/rakuten-ws.git
```

Or if you already pulled the sources:

```
$ pip install path/to/sources
```

Or if you don't have pip:

```
$ easy_install rakuten-ws
```


CHAPTER 3

Usage

To use Python Rakuten Web Service in a project:

```
import rakuten_ws
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at <https://github.com/alexandriagroup/rakuten-ws/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

Python Rakuten Web Service could always use more documentation, whether as part of the official Python Rakuten Web Service docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/alexandriagroup/rakuten-ws/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *rakuten-ws* for local development.

1. Fork the *rakuten-ws* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/rakuten-ws.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv rakuten_ws
$ cd rakuten-ws/
$ make init
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 rakuten_ws tests
$ py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/alexandriagroup/rakuten-ws/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test tests/test_rakuten_ws
```


CHAPTER 5

Indices and tables

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