Python Flipkart Marketplace API Client Documentation

Release 0.2.0

Fulfi.IO Inc.

August 14, 2015

Contents

1 Python Flipkart Marketplace API Client **3**
- 1.1 Installing . . . 3
- 1.2 Example Usage . . . 3
- 1.3 Features . . . 6

2 Contributing **7**
- 2.1 Types of Contributions . . . 7
- 2.2 Get Started! . . . 8
- 2.3 Pull Request Guidelines . . . 8
- 2.4 Tips . . . 9

3 Credits **11**
- 3.1 Development Lead . . . 11
- 3.2 Contributors . . . 11

4 Indices and tables **13**

Contents:

CHAPTER 1

Python Flipkart Marketplace API Client

Python Flipkart Marketplace API Client

- Free software: BSD license
- Documentation: https://python-flipkart.readthedocs.org.

1.1 Installing

From PYPI:

```
$ pip install python-flipkart
```

From source code (advanced users and for development):

```
$ git clone https://github.com/fulfilio/python-flipkart.git
$ cd python-flipkart
$ python setup.py install
```

1.2 Example Usage

```
from flipkart import FlipkartAPI, Authentication

auth = Authentication('app id', 'app secret', sandbox=True)
token = auth.get_token_from_client_credentials()

flipkart = FlipkartAPI(token['access_token'], sandbox=True, debug=True)
orders = flipkart.search_orders()
```

1.2.1 Get listings of a SKU

```
sku = flipkart.sku('my-special-sku', fsn='TSHDBN3326TEZHQZ')
for listing in sku.listings:
    print listing.attributes['mrp']
```

1.2.2 Create a listing

```
sku = flipkart.sku('my-special-sku', fsn='TSHDBN3326TEZHQZ')
listing = sku.create_listing(
    mrp=2400,
    selling_price=2300,
    listing_status="INACTIVE",
    fulfilled_by="seller",
    national_shipping_charge=20,
    zonal_shipping_charge=20,
    local_shipping_charge=20,
    procurement_sla=3,
    stock_count=23,
)
listing.save()
print listing.mrp
```

1.2.3 Update a listing

```
listing = flipkart.listing('LSTTSHDBN332XDYBZ5MHX30XI')
listing.attributes['mrp'] = 2600
listing.save()
```

1.2.4 Searching for orders

```
orders = flipkart.search_orders()
```

Find only orders of selected SKUs:

```
orders = flipkart.search_orders(
    filters={'sku': ['my-sku-1', 'my-sku-2']}
)
```

Filter by state:

```
orders = flipkart.search_orders(
    filters={'states': ['Approved']}
)
```

Tip: For a list of valid state see API documentation

1.2.5 Fetching a specific order item

```
order_item = flipkart.order_item('1731')
order_item.attributes['quantity']
```

Or to get several order items at once

```
order_items = flipkart.order_items('1731', '1732')
```

Once the order is ready to pack, generate a label

```
label_request = order_item.generate_label(
    date.today(),    # Invoice date
    'INV12345',      # Invoice number
)
```

When there are items that need serial numbers

```
label_request = order_item.generate_label(
    date.today(),    # Invoice date
    'INV12345',      # Invoice number
    [['IMEI1']],
)
```

If the item was dual sim

```
label_request = order_item.generate_label(
    date.today(),    # Invoice date
    'INV12345',      # Invoice number
    [['IMEI1', 'IMEI2']],
)
```

If 2 units of dual sim mobiles

```
label_request = order_item.generate_label(
    date.today(),    # Invoice date
    'INV12345',      # Invoice number
    [['IMEI1', 'IMEI2'], ['IMEI3', 'IMEI4']],
)
```

The response of `generate_label` is a Label Request. The label request is a lazy API. The status can be refreshed by calling

```
label_request.refresh_status()
```

Once the status is cleared, the item can be shipped out. To get the label to ship call the `get_label` method to get a PDF of the label and possibly the invoice.

```
pdf = order_item.get_label()
```

Once your shipment is ready to be picked by Flipkart logistics partner, call the ready to `dispatch` API.

```
order_item.dispatch()
```

1.2.6 Getting shipment details

The Shipments API gives the shipping details for orderitems

```
order_item.get_shipment_details()
```

the response items can be seen on Flipkart API documentation

1.2.7 Getting Access Token

If you have registered an application with your seller credentials and would like to access resources in your account, you could use the application id and secret alone to do so. The authentication helper in the API gives you a convenient way to get tokens

```
from auth import Authentication

auth = Authentication(
    '<application id>',
    '<application secret>',
    sandbox=True,           # If you are using sandbox
)
auth.get_token_from_client_credentials()
```

1.3 Features

- TODO

CHAPTER 2

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

2.1 Types of Contributions

2.1.1 Report Bugs

Report bugs at https://github.com/fulfilio/python-flipkart/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.

2.1.2 Fix Bugs

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

2.1.3 Implement Features

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

2.1.4 Write Documentation

Python Flipkart Marketplace API Client could always use more documentation, whether as part of the official Python Flipkart Marketplace API Client docs, in docstrings, or even on the web in blog posts, articles, and such.

2.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/fulfilio/python-flipkart/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 :)

2.2 Get Started!

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

1. Fork the *python-flipkart* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/python-flipkart.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 python-flipkart
$ cd python-flipkart/
$ python setup.py develop
```

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 python-flipkart tests
$ python setup.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.

2.3 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/fulfilio/python-flipkart/pull_requests and make sure that the tests pass for all supported Python versions.

2.4 Tips

To run a subset of tests:

```
$ py.test tests/test_flipkart.py
```

CHAPTER 3

Credits

3.1 Development Lead

- Sharoon Thomas, Fulfi.IO Inc. <sharoon.thomas@fulfil.io>

3.2 Contributors

None yet. Why not be the first?

CHAPTER 4

Indices and tables

- genindex
- modindex
- search