
Shipfunk Documentation

Release 0.1.1

Jaana Sarajärvi

Feb 08, 2023

Contents

1	Shipfunk	3
1.1	Features	3
1.2	Credits	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
3.1	Using Shipfunk class	7
3.2	Using ShipfunkUser class	8
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
4.5	Releasing new version	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.1.1 (2019-10-28)	15
6.2	0.1.0 (2018-01-08)	15
7	Indices and tables	17

Contents:

Python module for Shipfunk API

- Free software: MIT license
- Documentation: <https://shipfunk-python.readthedocs.io>.

1.1 Features

- Deliveries handling
- Pickup point search
- Package card and tracking code creation
- User accounts handling

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install Shipfunk, run this command in your terminal:

```
$ pip install shipfunk_python
```

This is the preferred method to install Shipfunk, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Shipfunk can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/vilkasgroup/shipfunk_python
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/vilkasgroup/shipfunk_python/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


There are two different ways to use this module

- **Using class Shipfunk:**
 - Search pickup points
 - Create package cards and tracking codes
- **Using class ShipfunkUser:**
 - Handle user accounts

3.1 Using Shipfunk class

```
from shipfunk_python.shipfunk import Shipfunk

# create object
shipfunk_client = Shipfunk('your_apikey', 'order_number')

# Define parameters in dictionary
params = {
    'postal_code': 30100,
    'country': 'fi',
    'products': [{
        "amount": 1,
        "code": "product123",
        "name": "Test product",
        "weight": {
            "amount": 1,
            "unit": "kg"
        },
    },
    "dimensions": {
        "unit": "cm",
        "width": "25",
```

(continues on next page)

(continued from previous page)

```
        "depth": "15",
        "height": "3"
    },
    ]]
}

# Call method with parameters
prices = shipfunk_client.get_price(params)
```

3.2 Using ShipfunkUser class

```
from shipfunk_python.shipfunk import ShipfunkUser

# create object
shipfunk_client = ShipfunkUser('your_apikey')

# Define parameters in dictionary
params = {
    "user": {
        "email": email@email.email,
        "locale": "FI",
        "eshop_name": "Example Store",
        "business_id": "12312345",
        "customs_id": "6543210",
        "contact_person_name": "Test Tester",
        "contact_person_phone": "040 1231234",
        "contact_person_email": email@email.email,
        "web_address": "real_deal.example.com",
        "customer_contact_info": "<b>Contact us:</b> service@example.com"
    }
}

# Call method with parameters
result = shipfunk_client.create_user(params)
```

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

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/vilkasgroup/shipfunk_python/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/vilkasgroup/shipfunk_python/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 :)

4.2 Get Started!

Ready to contribute? Here's how to set up *shipfunk_python* for local development.

1. Fork the *shipfunk_python* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/shipfunk_python.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 shipfunk_python
$ cd shipfunk_python/
$ 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 shipfunk_python tests
$ python setup.py test or 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.

4.3 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 2.7 and 3.6, and for PyPy. Check https://travis-ci.org/vilkasgroup/shipfunk_python/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_shipfunk_python
```

4.5 Releasing new version

When you are ready to release a new version follow these steps:

1. Merge all changes that should be included in the new release to master. And checkout master.
2. Update HISTORY.rst with the new version number and changes. And commit your changes to master.
3. run:

```
$ bumpversion patch|minor|major
```

4. push to master with tags to trigger travis deploy:

```
$ git push --tags
$ git push
```

Travis will build the tag and when it is successfull will also deploy to pypi

5.1 Development Lead

- Jaana Sarajarvi <jaana.sarajarvi@vilkas.fi>

5.2 Contributors

None yet. Why not be the first?

6.1 0.1.1 (2019-10-28)

- Updated requirements.txt. Newer “requests” module is needed.
- Updated test cases.

6.2 0.1.0 (2018-01-08)

- First release on PyPI.

CHAPTER 7

Indices and tables

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