ParcelBrigh API wrapper Documentation

Release 0.3.3

Marek Wywiał

October 20, 2015

Contents

Contents:

CHAPTER 1

ParcelBrigh API wrapper

ParcelBright API wrapper. For full ParcelBright API reference go to https://github.com/parcelbright/api-docs

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

1.1 Features

- create new shipment to get rates from
- find previously created shipment
- book shipment
- get tracking data
- cancel shipment

1.2 TODO

- Customs support
- Carriers liability support

CHAPTER 2

Installation

At the command line:

```
$ pip install parcelbright
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv parcelbright
$ pip install parcelbright
```

CHAPTER 3

Usage

To use ParcelBrigh API wrapper in a project:

```
>>> import parcelbright
>>> parcelbright.api_key = 'myapikey'
>>> parcelbright.sandbox = True  # use sandbox version

>>> # Create Parcel
>>> parcel = parcelbright.Parcel({
...     'width': 10, 'height': 10, 'length': 10, 'weight': 1
... })

>>> # Create from Address
>>> from_address = parcelbright.Address({
...     'name': 'office', 'postcode': 'NW1 0DU', 'town': 'London',
...     'phone': '07800000000', 'country_code': 'GB',
...     'line1': '19 Mandela Street'
... })

>>> # Create to Address
>>> to_address = parcelbright.Address({
...     'name': 'John Doe', 'postcode': 'E2 8RS', 'town': 'London',
...     'phone': '07411111111', 'country_code': 'GB',
...     'line1': '19 Mandela Street'
... })

>>> # Create shipment
>>>    shipment = parcelbright.Shipment({
...     'customer_reference': '123455667', 'estimated_value': 100,
...     'contents': 'books', 'pickup_date': '2025-01-29',
...     'parcel': parcel, 'from_address': from_address,
...     'to_address': to_address
... })
>>>  print shipment.id
None

# Call API to create
>>> shipment.create()
>>> print shipment.id

>>> print shipment.rates

>>> # Find previously created shipment
>>> shipment = parcelbright.Shipment.find('prb6c8c0')
```

```
>>> # Book created or found shipment using rates code
>>> shipment.book(rate_code='N')

>>> print shipment.label_url
>>> print shipment.consignment
>>> print shipment.pickup_confirmation

>>> # Get tracking data
>>> tracking = shipment.track()
>>> print tracking

>>> # Cancell shipment
>>> shipment.cancel()
```

CHAPTER 4

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/onjin/parcelbright-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" is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

ParcelBrigh API wrapper could always use more documentation, whether as part of the official ParcelBrigh API wrapper 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/onjin/parcelbright-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 *parcelbright-python* for local development.

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

```
$ git clone git@github.com:your_name_here/parcelbright-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 parcelbright
$ cd parcelbright-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 parcelbright tests
$ python setup.py test
$ tox
```

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

   To run integration tests against your sandbox account $ PARCELBRIGHT_TEST_API_KEY=yourapikey python setup.py test

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 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/onjin/parcelbright-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_parcelbright
```

CHAPTER 5

Credits

5.1 Development Lead

- Marek Wywiał <onjinx@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

CHAPTER 7

0.3.3 (2015-10-20)

- Remove assertions from API facade

0.3.2 (2015-08-14)

- Added shipment.track to schema

CHAPTER 9

0.3.1 (2015-08-07)

- Added missing *schematics* requirements at *setup.py*

CHAPTER 10

0.3.0 (2015-08-07)

- API CHANGED - Entities constructor takes *dict* instead of *kwargs*
- API CHANGED - *Shipment.create* becames instance method instead of class method
- Added entities validation using *schematics* package

CHAPTER 11

0.2.3 (2015-07-29)

- Support for *pickup_date* parameter at *Shipment.book()* method
- Added *__repr__* method to *Parcel*, *Address* and *Shipment* entities

0.2.2 (2015-07-29)

- Raise *TrackingError* from *Shipment.track()* if shipment has not tracking information available

CHAPTER 13

0.2.1 (2015-07-29)

- Remove *Shipment.is_booked()* method. Use *Shipment.state* instead.
- Raise *ShipmentNotCompletedException* from *Shipment.track()* method

0.2.0 (2015-07-28)

- Added support to track and cancel shipments

CHAPTER 15

0.1.0 (2015-07-27)

- First release on PyPI.

CHAPTER 16

Indices and tables

- genindex
- modindex
- search