
Royal Mail Rest API Documentation

Release 0.0.6

Alex Hellier

Apr 06, 2018

Contents:

1	Royal Mail Rest API	1
1.1	Features	1
1.2	Credits	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Example Shipping object from Royal Mail	9
5	royal_mail_rest_api	11
5.1	royal_mail_rest_api package	11
6	Contributing	15
6.1	Types of Contributions	15
6.2	Get Started!	16
6.3	Pull Request Guidelines	17
6.4	Tips	17
6.5	Deploying	17
7	Credits	19
7.1	Development Lead	19
7.2	Contributors	19
8	History	21
8.1	0.0.6 (2018-04-06)	21
8.2	0.0.5	21
8.3	0.0.4	21
8.4	0.0.3 (2018-04-03)	21
8.5	0.0.2 (2018-04-03)	21
8.6	0.0.1 (2018-03-26)	21
9	Indices and tables	23
	Python Module Index	25

Royal Mail Rest API

A small helper for integrating python with Royal Mails Rest API

This is to help with getting started integrating Royal Mails shipping and Tracking API's into your project.

They do have an Open API to build this with Swagger-Codegen, but I found there were a few issues with this. 1. Horrible final code - overly verbose and complicated. 1. Half the time it would not build (needed beta versions of swagger-codegen) and then it would still not work.

This is cleaner, easier to modify and extend, and a lot lighter weight, only using the standard library.

It is VERY MUCH a work in progress, so help is hugely appreciated, and be careful, it may change as I implement more features / improve and enhance what we already have.

- Free software: MIT license
- Documentation: <https://royal-mail-rest-api.readthedocs.io>.

1.1 Features

Create Labels Update Labels Create Manifests Post Manifests

1.2 Credits

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

2.1 Stable release

To install Royal Mail Rest API, run this command in your terminal:

```
$ pip install royal_mail_rest_api
```

This is the preferred method to install Royal Mail Rest API, 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 Royal Mail Rest API can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/bobspadger/royal_mail_rest_api
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/bobspadger/royal_mail_rest_api/tarball/master
```

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

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


To use Royal Mail Rest API in a project:

Note: (this is from the example.py file which will be updated, its worth looking there)

```
import json
from royal_mail_rest_api.tools import RoyalMailBody
from royal_mail_rest_api.shipping import ShippingApi
from royal_mail_rest_api.tracking import TrackingApi
from royal_mail_rest_api.get_credentials import return_credentials

if __name__ == '__main__':
    # Get our API credentials (from credentials.ini)
    creds = return_credentials()

    # make some nice easy to read variables for authenticating with
    CLIENT_ID = creds['royal_mail']['CLIENT_ID']
    CLIENT_SECRET = creds['royal_mail']['CLIENT_SECRET']
    USERNAME = creds['royal_mail']['USERNAME']
    PASSWORD_HASHED = creds['royal_mail']['PASSWORD_HASHED']

    # Create a new delivery object to make our address etc with
    body = RoyalMailBody('Delivery')

    # Add our items to it
    body.add_ship_date(None)
    body.add_service('P', 1, 'TPN', 'T', True, ['14'])
    body.customer_reference = 'D123456'
    body.department_reference = 'Q123456'
    body.sender_reference = 'A123456'
    body.add_items(1, 100, 'g')
    body.add_recipient_contact('Joe Bloggs', 'joe.bloggs@royalmail.com', None,
    ↪ '07970810000')
    body.add_recipient_address('Broadgate Circle', 'London', None, 'EC1A 1BB',
    ↪ country='GB', building_number='1',
```

```
        address_line2='Add line 2', address_line3='Add line 3
↳', building_name='My building')

# Request our body to use to request a label from royal mail
my_rm_body = body.return_domestic_body()

# Create a shipping object, populate it with our credentials
my_shipping = ShippingApi(CLIENT_ID, CLIENT_SECRET, USERNAME, PASSWORD_HASHED)
# Request an authentication token
my_shipping.get_token()
# Post a shipping request
post_shipping = my_shipping.post_domestic(my_rm_body)

# Store our tracking id for use.
tracking_ref = post_shipping['completedShipments'][0]['shipmentItems'][0][
↳'shipmentNumber']

# Get a label ! this is base64 encoded
label = my_shipping.put_shipment_label(tracking_ref)

# Now lets change some info about the receipt
body.add_receipt_contact('Alex Hellier', 'alex@me.com', 'Alex S Hellier',
↳'123455')
# And get our update body - this is slightly different from the original request
new_data = body.return_domestic_update_body()
# Request a change
change_name = my_shipping.put_shipment(tracking_ref, new_data)
# Request a new label
new_label = my_shipping.put_shipment_label(tracking_ref)

# Lets delete the post request
delete_shipping = my_shipping.delete_shipment(tracking_ref)

# If we have some labels to manifest - request it
manifest_info = {'yourReference': '123'}
manifest_data = my_shipping.post_manifest(manifest_info)
# Get the manifest doumentation - note, you will need the maniefest number to get
↳this
manifest_label = my_shipping.put_manifest(manifest_batch_number=5)

# Now, a period of time has passed, we can track those packages
tracking_api = TrackingApi(CLIENT_ID, CLIENT_SECRET)
try:
    test_tracking = tracking_api.summary(tracking_ref)
    print(test_tracking)
except Exception as e:
    print(e)

try:
    test_pod = tracking_api.proof_of_delivery(tracking_ref)
    print(test_pod)
except Exception as e:
    print(e)

try:
    history_tracking = tracking_api.history(tracking_ref)
```

```
print(history_tracking)
except Exception as e:
    print(e)
```

Example Shipping object from Royal Mail

```
{
  "shipmentType": "Delivery",
  "service": {
    "format": "P",
    "occurrence": "1",
    "offering": "TPN",
    "type": "T",
    "signature": "true",
    "enhancements": ["14"]
  }
},
"shippingDate": "2017-09-25",
"items": [
  {
    "count": 1,
    "weight": {
      "unitOfMeasure": "g",
      "value": 100
    }
  }
],
"recipientContact": {
  "name": "Joe Bloggs",
  "complementaryName": "null",

  "email": "joe.bloggs@royalmail.com"
},

"recipientAddress": {
  "buildingName": "Cable and Engineering Limited",
  "buildingNumber": "1",
  "addressLine1": "Broadgate Circle",
  "addressLine2": "Address line 2",
  "addressLine3": "Address Line 3",
```

```
"postTown": "London",
"country": "GB",
"postCode": "EC1A 1BB"
},
"senderReference": "Senders Ref",
"departmentReference": "Dept Ref",
"customerReference": "Do not use",
"safePlace": "null"
}
```

5.1 `royal_mail_rest_api` package

5.1.1 Submodules

5.1.2 `royal_mail_rest_api.api` module

Main module.

```
class royal_mail_rest_api.api.RoyalMailBaseClass
    Bases: object
    BASE CLASS FOR SHIPPING
    url = 'https://api.royalmail.net'
```

5.1.3 `royal_mail_rest_api.errors` module

```
exception royal_mail_rest_api.errors.GeneralError
    Bases: Exception
exception royal_mail_rest_api.errors.NotAuthorised
    Bases: Exception
exception royal_mail_rest_api.errors.RoyalMailError
    Bases: Exception
```

5.1.4 `royal_mail_rest_api.example` module

5.1.5 `royal_mail_rest_api.get_credentials` module

```
royal_mail_rest_api.get_credentials.return_credentials()
```

5.1.6 royal_mail_rest_api.shipping module

class royal_mail_rest_api.shipping.**ShippingApi** (*client_id, client_secret, username, password*)

Bases: *royal_mail_rest_api.api.RoyalMailBaseClass*

Royal Mail Shipping Class, used to communicate with the Royal Mail Rest API to create labels

delete_shipment (*shipment_number*)

Description

Delete a shipment. Send a shipment identifier in Url. Successful response will be 200 with no content.

Returns

delete_shipment_url = '/shipping/v2/'

get_token ()

Method to get a JWT token

This method will accept a DMO/NEOPOST user name and password. On successful validation of the user credential it will issue a JWT token to the user which will be valid for 4 hours. On subsequent requests, user will pass the JWT token in the request header.

Returns

post_domestic (*data*)

Operation to create a shipment

This method will take a domestic shipment request in the body and on successful response, it will return the shipment numbers and item details. :return:

post_domestic_url = '/shipping/v2/domestic'

post_manifest (*manifest_options=None*)

This method creates a shipping manifest

Returns

post_manifest_url = '/shipping/v2/manifest'

put_manifest (*sales_order_number=None, manifest_batch_number=None*)

This method return a manifest label for a previously manifested shipment.

Returns

put_shipment (*shipment_number, data*)

updateShipment

Update a shipment. Send a shipment request in body. On successful response, it will return shipment number and warnings. Service related information can not be updated, and if passed as part of request, it will be ignored.

Returns

put_shipment_label (*shipment_number*)

This method returns a label for the shipment identifier passed in the url.

Returns

put_shipment_label_url = '/shipping/v2/'

put_shipment_update_url = '/shipping/v2/'

token_url = '/shipping/v2/token'

5.1.7 royal_mail_rest_api.tools module

class royal_mail_rest_api.tools.RoyalMailBody (*shipment_type*)

Bases: object

add_items (*number, weight, unit_of_measure*)

Add items- currently only a single item :param number: :param weight: :param unit_of_measure: :return:

add_recipient_address (*address_line1, post_town, county, postcode, country, building_name=None, building_number=None, address_line2=None, address_line3=None*)

Add address of recipient :param address_line1: :param post_town: :param county: :param postcode: :param country: :param building_name: :param building_number: :param address_line2: :param address_line3: :return:

add_recipient_contact (*name, email, complementary_name=None, telephone=None*)

Add the name and contact of who this is being sent to :param name: :param email: :param complementary_name: :param telephone: :return:

add_service_enhancements (*enhancement*)

add a single service enhancement, can be called multiple times to append required items :param enhancement: :return:

add_service_format (*format=None*)

add a valid service format using our friendly names :param format: :return:

add_service_occurence ()

add_service_offering (*service_offering=None*)

add a valid service offering using our friendly names :param service_offering: :return:

add_service_type (*service_type=None*)

add a valid service type using our friendly names :param service_type: :return:

add_ship_date (*date_obj=None*)

take a datetime object and format it to royal mails Y-m-d format :param date_obj: :return:

add_signature (*signature=False*)

Do we want a signature on delivery :param signature: :return:

static remove_none_values (*iterable*)

take out values of None by removing the key :param iterable: :return: dictionary

return_domestic_body ()

build domestic body from items :return:

return_domestic_update_boy ()

build domestic body from items :return:

service_enhancements = {'loss_7500': '4', 'signature': '12', 'local_collect': '22',

service_formats = {'inland_letter': 'L', 'inland_large_letter': 'F', 'inland_format_

service_offerings = {'intl_bus_parcels_signed_extra_comp': 'MP6', 'intl_signed_on_acc

service_types = {'special_delivery': 'D', 'tracked_returns': 'R', 'international':

5.1.8 royal_mail_rest_api.tracking module

class royal_mail_rest_api.tracking.TrackingApi (*client_id, client_secret*)

Bases: royal_mail_rest_api.api.RoyalMailBaseClass

Start class for royal mail shipping api

history (*tracking_number*)

Return history for a tracked item :param tracking_number: :return:

history_url = 'mailPieces/{}/history'

pod_url = 'mailPieces/{}/proofOfDelivery'

proof_of_delivery (*tracking_number*)

recover proof of delivery :param tracking_number: :return:

summary (*tracking_number*)

takes 13 digit tracking number and requests summary data :param tracking_number: :return: tracking_summary

summary_url = 'mailPieces/{}/summary'

5.1.9 Module contents

Top-level package for Royal Mail Rest API.

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

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/bobspadger/royal_mail_rest_api/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.

6.1.2 Fix Bugs

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

6.1.3 Implement Features

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

6.1.4 Write Documentation

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

6.1.5 Submit Feedback

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

6.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/royal_mail_rest_api.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 royal_mail_rest_api
$ cd royal_mail_rest_api/
$ 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 royal_mail_rest_api 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.

6.3 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, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/bobspadger/royal_mail_rest_api/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ py.test tests.test_royal_mail_rest_api
```

6.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

7.1 Development Lead

- Alex Hellier <alex.hellier@gmail.com>

7.2 Contributors

None yet. Why not be the first?

8.1 0.0.6 (2018-04-06)

Raise useful errors from the api calls so they are returned to the calling app

8.2 0.0.5

Tests Service helpers

8.3 0.0.4

8.4 0.0.3 (2018-04-03)

Store token for 4 hours (issue #7) Fix some poor documentation Fix some examples

8.5 0.0.2 (2018-04-03)

Fixing PyPI install not working

8.6 0.0.1 (2018-03-26)

- First release on PyPI.

CHAPTER 9

Indices and tables

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

r

royal_mail_rest_api, 14
royal_mail_rest_api.api, 11
royal_mail_rest_api.errors, 11
royal_mail_rest_api.example, 11
royal_mail_rest_api.get_credentials, 11
royal_mail_rest_api.shipping, 12
royal_mail_rest_api.tools, 13
royal_mail_rest_api.tracking, 13

A

add_items() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_recipient_address() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_recipient_contact() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_service_enhancements() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_service_format() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_service_occurrence() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_service_offering() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_service_type() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_ship_date() (royal_mail_rest_api.tools.RoyalMailBody method), 13
 add_signature() (royal_mail_rest_api.tools.RoyalMailBody method), 13

D

delete_shipment() (royal_mail_rest_api.shipping.ShippingApi method), 12
 delete_shipment_url (royal_mail_rest_api.shipping.ShippingApi attribute), 12

G

GeneralError, 11
 get_token() (royal_mail_rest_api.shipping.ShippingApi method), 12

H

history() (royal_mail_rest_api.tracking.TrackingApi method), 14

history_url (royal_mail_rest_api.tracking.TrackingApi attribute), 14

N

NotAuthorised, 11

P

pod_url (royal_mail_rest_api.tracking.TrackingApi attribute), 14
 post_domestic() (royal_mail_rest_api.shipping.ShippingApi method), 12
 post_domestic_url (royal_mail_rest_api.shipping.ShippingApi attribute), 12
 post_manifest() (royal_mail_rest_api.shipping.ShippingApi method), 12
 post_manifest_url (royal_mail_rest_api.shipping.ShippingApi attribute), 12
 proof_of_delivery() (royal_mail_rest_api.tracking.TrackingApi method), 14
 put_manifest() (royal_mail_rest_api.shipping.ShippingApi method), 12
 put_shipment() (royal_mail_rest_api.shipping.ShippingApi method), 12
 put_shipment_label() (royal_mail_rest_api.shipping.ShippingApi method), 12
 put_shipment_label_url (royal_mail_rest_api.shipping.ShippingApi attribute), 12
 put_shipment_update_url (royal_mail_rest_api.shipping.ShippingApi attribute), 12

R

remove_none_values() (royal_mail_rest_api.tools.RoyalMailBody static method), 13
 return_credentials() (in module royal_mail_rest_api.get_credentials), 11
 return_domestic_body() (royal_mail_rest_api.tools.RoyalMailBody method), 13

return_domestic_update_boy()
 (royal_mail_rest_api.tools.RoyalMailBody
 method), 13

royal_mail_rest_api (module), 14

royal_mail_rest_api.api (module), 11

royal_mail_rest_api.errors (module), 11

royal_mail_rest_api.example (module), 11

royal_mail_rest_api.get_credentials (module), 11

royal_mail_rest_api.shipping (module), 12

royal_mail_rest_api.tools (module), 13

royal_mail_rest_api.tracking (module), 13

RoyalMailBaseClass (class in royal_mail_rest_api.api),
11

RoyalMailBody (class in royal_mail_rest_api.tools), 13

RoyalMailError, 11

S

service_enhancements (royal_mail_rest_api.tools.RoyalMailBody
attribute), 13

service_formats (royal_mail_rest_api.tools.RoyalMailBody
attribute), 13

service_offerings (royal_mail_rest_api.tools.RoyalMailBody
attribute), 13

service_types (royal_mail_rest_api.tools.RoyalMailBody
attribute), 13

ShippingApi (class in royal_mail_rest_api.shipping), 12

summary() (royal_mail_rest_api.tracking.TrackingApi
method), 14

summary_url (royal_mail_rest_api.tracking.TrackingApi
attribute), 14

T

token_url (royal_mail_rest_api.shipping.ShippingApi at-
tribute), 12

TrackingApi (class in royal_mail_rest_api.tracking), 13

U

url (royal_mail_rest_api.api.RoyalMailBaseClass at-
tribute), 11