pyPrintful Documentation

Release 1.0.0

559 Labs

Mar 15, 2017

Contents

CHAPTER 1

Quick Start

Installation

To install this package, just type:

```
pip3 install pyPrintful
```

This will load the package as well as any additional requirements to your development environment.

Basic Usage

Simply add the requisite import to your script and make the requisite call to the API wrapper:

```
>>> from pyPrintful import *
>>> import json
```

```
>>> my_key = "XXXXXXXXXXXX:XXXXXXXXXXXXXXX"
```

```
>>> p = pyPrintful(key=my_key)
>>> data = p.get_store_info()
```

```
>>> print( json.dumps( data, indent=4 ) )
```

(Note that I'm importing the 'json' package here so that the output is pretty, but it's not required).

Check out the module documentation for additional methods. You can call the named methods (such as get_store_info()), or if you need to go lower, you also have access to:

- do_get()
- do_post()
- do_delete()

- do_put()

CHAPTER 2

High-Level Vocabulary

These methods abstract all of the API details from the interface, making it simple to get and set the information you require. If you need more fine-grained control of the calls, use the *Low-Level Vocabulary* (of course, these can be used interchangeably).

"Pull/Retrieve" Methods

`pyPrintful.`**`get_countries_list`**`()`
Retrieve country list

`pyPrintful.`**`get_file_info`**`(`*pk=None*`)`
Get file info

`pyPrintful.`**`get_file_list`**`()`
Get list of files

`pyPrintful.`**`get_order_info`**`(`*pk=None*`)`
Get order data

`pyPrintful.`**`get_order_list`**`()`
Get order list

`pyPrintful.`**`get_product_info`**`(`*pk=None*`)`
Get product's variant list

Parameters **`pk`** – The Printful identifier for the variant.

`pyPrintful.`**`get_product_list`**`()`
Get all product list

`pyPrintful.`**`get_shippingrate_calc`**`(`*data*`)`
Calculate shipping rates

`pyPrintful.`**`get_store_info`**`()`
Get store info

`pyPrintful.`**`get_syncproduct_info`**(*pk=None*)
Get info about sync product

`pyPrintful.`**`get_syncproduct_list`**()
Get list of sync products

`pyPrintful.`**`get_syncvariant_info`**(*pk=None*)
Get info about sync variant

`pyPrintful.`**`get_tax_calc`**(*data=None*)
Calculate tax rate

`pyPrintful.`**`get_tax_geos`**()
Retrieve state list that requires state tax calc

`pyPrintful.`**`get_variant_info`**(*pk=None*)
Get info about a variant

Parameters **`pk`** – The Printful identifier for the variant.

`pyPrintful.`**`get_webhooks_info`**()
Get webhook configuration

“Push/Save” Methods

`pyPrintful.`**`put_file_new`**(*data*)
Add new file

`pyPrintful.`**`put_order_cancel`**(*pk=None*)
Cancel an order

`pyPrintful.`**`put_order_confirm`**(*pk=None*)
Confirm draft for fulfillment

`pyPrintful.`**`put_order_new`**(*data=None*)
Create new order

`pyPrintful.`**`put_order_update`**(*pk=None*, *data=None*)
Update order data

`pyPrintful.`**`put_store_packingslip`**(*data=None*)
Change store packing slip

`pyPrintful.`**`put_syncproduct_remove`**(*pk=None*)
Unlink all synced variants of this product

`pyPrintful.`**`put_webhooks_disable`**()
Disable webhook support. No data param, because the API doesn’t require it.

`pyPrintful.`**`put_webhooks_update`**(*data=None*)
Set up webhook configuration

CHAPTER 3

Low-Level Vocabulary

These methods provide a lower-level abstraction of the API. Basically, they simply provide standard GET, POST, DELETE and PUT functionality. You will need to provide information such as API path. If your needs are a bit simpler, use the *High-Level Vocabulary*.

`pyPrintful.`**`do_get`**(*path*, *params=None*)

Perform a GET request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`params`** – Additional GET parameters as a dictionary

`pyPrintful.`**`do_post`**(*path*, *data=None*, *params=None*)

Perform a POST request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`data`** – Request body data as a dictionary
- **`params`** – Additional GET parameters as a dictionary

`pyPrintful.`**`do_delete`**(*path*, *params=None*)

Perform a DELETE request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`params`** – Additional GET parameters as a dictionary

`pyPrintful.`**`do_put`**(*path*, *data=None*, *params=None*)

Perform a PUT request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`data`** – Request body data as a dictionary

- **`params`** – Additional PUT parameters as a dictionary

CHAPTER 4

Contributing

Set Up Your Environment

This project is designed for the Python 3 series... so you'll need to have that available. To set up everyting else, do the following:

```
$ mkdir pyPrintful
$ cd pyPrintful
$ git clone https://github.com/559Labs/pyPrintful.git .
$ python3 -m venv env
$ source env/bin/activate
$ pip install -r requirements.txt
```

Build the Documentation

If you set up your project as described above, you should have Sphinx commands available. So, do the following from the project root folder:

```
$ cd docs
$ make html
```

The rendered files should now be saved in `./docs/_build/html/`.

CHAPTER 5

pyPrintful Module

exception `pyPrintful.`**`pfAPIException`**(*message*, *code*)

Bases: `pyPrintful.pfException`

API Exception Class

exception `pyPrintful.`**`pfException`**

Bases: `Exception`

Printful exception returned from the API.

class `pyPrintful.`**`pyPrintful`**(*key=None*, *connect=True*)

Bases: `object`

Printful API class. Initializes the connection to the API server.

Parameters **`key`** – API Key (Get it from your store's dashboard). Note that this is not the consumer key and secret found under store info. Rather, the API key can be found under Store > API and will be two strings separated by a ':'.

Returns A stateful object with an authenticated connection.

`VERSION` **= '1.0.0'**

`connect`()

Configures the connection to the remote server.

`do_delete`(*path*, *params=None*)

Perform a DELETE request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`params`** – Additional GET parameters as a dictionary

`do_get`(*path*, *params=None*)

Perform a GET request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`params`** – Additional GET parameters as a dictionary

`do_post`(*path*, *data=None*, *params=None*)
Perform a POST request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`data`** – Request body data as a dictionary
- **`params`** – Additional GET parameters as a dictionary

`do_put`(*path*, *data=None*, *params=None*)
Perform a PUT request to the API

Parameters

- **`path`** – Request path (e.g. 'orders' or 'orders/123')
- **`data`** – Request body data as a dictionary
- **`params`** – Additional PUT parameters as a dictionary

`get_countries_list`()
Retrieve country list

`get_file_info`(*pk=None*)
Get file info

`get_file_list`()
Get list of files

`get_item_count`()
Returns total available item count from the last request if it supports paging (e.g order list) or nil otherwise

`get_order_info`(*pk=None*)
Get order data

`get_order_list`()
Get order list

`get_product_info`(*pk=None*)
Get product's variant list

Parameters **`pk`** – The Printful identifier for the variant.

`get_product_list`()
Get all product list

`get_shippingrate_calc`(*data*)
Calculate shipping rates

`get_store_info`()
Get store info

`get_syncproduct_info`(*pk=None*)
Get info about sync product

`get_syncproduct_list`()
Get list of sync products

`get_syncvariant_info`(*pk=None*)
Get info about sync variant

`get_tax_calc`(*data=None*)
Calculate tax rate

`get_tax_geos`()
Retrieve state list that requires state tax calc

`get_variant_info`(*pk=None*)
Get info about a variant

Parameters **`pk`** – The Printful identifier for the variant.

`get_webhooks_info`()
Get webhook configuration

`put_file_new`(*data*)
Add new file

`put_order_cancel`(*pk=None*)
Cancel an order

`put_order_confirm`(*pk=None*)
Confirm draft for fulfillment

`put_order_new`(*data=None*)
Create new order

`put_order_update`(*pk=None*, *data=None*)
Update order data

`put_store_packingslip`(*data=None*)
Change store packing slip

`put_syncproduct_remove`(*pk=None*)
Unlink all synced variants of this product

`put_webhooks_disable`()
Disable webhook support. No data param, because the API doesn't require it.

`put_webhooks_update`(*data=None*)
Set up webhook configuration

CHAPTER 6

Indices and Tables

- genindex
- modindex
- search

Index