ePages Client Documentation

Release 0.2.0

Pekka Piispanen, Tero Kotti

Feb 11, 2020

Contents

1 ePages Client **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 Basic usage . . . 7
3.2 The dictionary parameter . . . 7
3.3 Currency and locale . . . 9

4 Methods **11**
4.1 Customers . . . 11
4.2 Legal information . . . 12
4.3 Newsletters . . . 13
4.4 Orders and carts . . . 14
4.5 Products . . . 16
4.6 Script tags . . . 22

5 Contributing **23**
5.1 Types of Contributions . . . 23
5.2 Get Started! . . . 24
5.3 Pull Request Guidelines . . . 25
5.4 Tips . . . 25

6 Credits **27**
6.1 Development Lead . . . 27
6.2 Contributors . . . 27

7 History **29**
7.1 0.2.0 (2020-02-11) . . . 29
7.2 0.1.0 (2018-01-05) . . . 29

8 Indices and tables **31**

Contents:

CHAPTER 1

ePages Client

Python 3 client for ePages REST API.

- Free software: MIT license
- Documentation: https://epages-client.readthedocs.io.

1.1 Features

- TODO

1.2 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install ePages Client, run this command in your terminal:

```
$ pip install epages_client
```

This is the preferred method to install ePages Client, 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 ePages Client can be downloaded from the Github repo.

You can either clone the public repository:

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

Or download the tarball:

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

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

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

CHAPTER 3

Usage

3.1 Basic usage

To use ePages Client in a project:

```
from epages_client.client import RestClient

# Set the api url and token for your shop
api_url = "https://yourshop.vilkasstore.com/rs/shops/yourshop/"
api_token = "shop_token_goes_here"

# Get the instance of the client
client = RestClient(api_url, api_token)

# Example method: get shop info
shop_info = client.get_shop_info()
```

3.2 The dictionary parameter

Each method accepts a single dictionary parameter called params. This parameter is not mandatory for all methods.

```
# params can have five different keys
params = {
    "data": "",
    "param1": "",
    "param2": "",
    "query": {},
    "object": "",
}
```

- **data**
  - The content here is usually a dictionary, but can have a binary file, too.

- **param1**
  - The first parameter to add to the api url.
- **param2**
  - The second parameter to add to the api url.
- **query**
  - A dictionary for query parameters, for example including hidden items in results.
- **object**
  - As the name implies, this gets an object as a value. Objects are located in the dataobjects directory.

3.2.1 Examples

```
# Example 1

# Set category id and product id
params["data"] = {
        'categoryId': "5A41E34F-BAC7-8396-336A-0A2810152BBC",
        'productId': "5A497829-7619-ACCC-E487-0A281012346F"
}

# Connect category and product
response = client.connect_category_and_product(params)

# Example 2

# Set product id
# The api url will be:
# https://yourshop.vilkasstore.com/rs/shops/yourshop/products/5A497829-7619-ACCC-E487-
↪0A281012346F
params["param1"] = "5A497829-7619-ACCC-E487-0A281012346F"

# Get product data
product = client.get_product(params)

# Example 3

# Set product id and image name
# The api url will be:
# https://yourshop.vilkasstore.com/rs/shops/yourshop/products/5A497829-7619-ACCC-E487-
↪0A281012346F/slideshow/test.jpg
params["param1"] = "5A497829-7619-ACCC-E487-0A281012346F"
params["param2"] = "test.jpg"

# Delete image from product
response = client.delete_product_image(params)

# Example 4

# Find products where name contains the word 'laptop'
# Limit search results to have 50 items
params["query"] = {
    "query": "laptop",
    "limit": 50
```

(continues on next page)

(continued from previous page)

```
}

# Search for the products
results = self.client.search_products(self.params)

# Example 5

# Create a customer
customer = CustomerCreate()
customer.billingAddress.firstName = "John"
customer.billingAddress.lastName = "Doe"
customer.billingAddress.emailAddress = "john.doe@mail.com"

# Add customer to params
params["object"] = customer

# Create a customer
response = client.add_customer(params)
```

3.3 Currency and locale

There are two ways to set currency and locale.

Note: If currency and locale are set using both setters and params["query"], values of params["query"] are used.

```
# Currency and locale are set using client setters
client.currency = "GBP"
client.locale = "en_US"

# Currency and locale are set using params["query"]
params["query"] = {
    "currency": = "GBP",
    "locale": = "en_US"
}
```

CHAPTER 4

Methods

Here's a list of available methods to use with ePages Client. Each method accepts a single dict parameter called params. The more detailed description of the param is found in the usage document. Method descriptions have a link to the ePages API, too.

4.1 Customers

4.1.1 get_customers

API Docs

This method fetches all customers from the shop.

Required parameters: none

4.1.2 get_customer

API Docs

This method fetches single customer from the shop.

Required parameters: Customer id in param1

4.1.3 add_customer

API Docs

This method adds a customer to the shop.

Required parameters: Instance of CustomerCreate in object

Note: When adding a customer, only the Address object must have something in *some instance variable*. It doesn't matter which variable it is. The Address object is in the billingAddress instance variable of CustomerCreate.

4.1.4 update_customer

API Docs

This method updates an existing customer in the shop.

Required parameters: Customer id in param1

Note: When updating customer, an instance of CustomerUpdate is not required. If the instance of CustomerUpdate is not sent or it is empty, nothing is updated.

4.2 Legal information

4.2.1 get_legal_information

API Docs

This method gets hyperlinks of legal information for a shop.

Required parameters: none

4.2.2 get_contact_information

API Docs

This method gets the contact information of a shop.

Required parameters: none

4.2.3 get_privacy_policy

API Docs

This method gets the privacy policy of a shop.

Required parameters: none

4.2.4 get_terms_and_conditions

API Docs

This method gets the terms and conditions of a shop.

Required parameters: none

4.2.5 get_rights_of_withdrawal

API Docs

This method gets the customer rights of withdrawal of a shop.

Required parameters: none

4.2.6 get_shipping_information

API Docs

This method gets the detailed information on possible shipping types and the costs incurred.

Required parameters: none

4.2.7 update_contact_information

API Docs

This method updates the contact information of a shop.

Required parameters: locale must be set

4.2.8 update_privacy_policy

API Docs

This method updates the privacy policy of a shop.

Required parameters: locale must be set

4.2.9 update_terms_and_conditions

API Docs

This method updates the terms and conditions of a shop.

Required parameters: locale must be set

4.2.10 update_rights_of_withdrawal

API Docs

This method updates the customer rights of withdrawal of a shop.

Required parameters: locale must be set

4.2.11 update_shipping_information

API Docs

This method updates the shipping information of a shop.

Required parameters: locale must be set

4.3 Newsletters

4.3.1 get_newsletter_campaigns

API Docs

This method gets the newsletter campaigns from a shop.

Required parameters: none

4.3.2 get_newsletter_campaign_subscribers

API Docs

This method gets the subscribers of a newsletter campaign from a shop.

Required parameters: Newsletter campaign id in param1

4.4 Orders and carts

4.4.1 get_orders

API Docs

This method gets the orders from a shop.

Required parameters: none

4.4.2 get_order

API Docs

This method gets the information of a single order.

Required parameters: Order id in param1

4.4.3 get_order_documents

API Docs

This method gets finalized invoice and credit note order documents of a single order.

Required parameters: Order id in param1

4.4.4 get_sales

API Docs

This method gets the summary of sales figures.

Required parameters: none

4.4.5 get_cart

API Docs

This method gets a single cart from a shop.

Required parameters: Cart id in param1

4.4.6 add_cart

API Docs

This method adds a cart for a shop.

Required parameters: none

4.4.7 add_coupon

API Docs

This method applies a coupon code on a cart of a shop.

Required parameters: Cart id in param1, coupon code in data

4.4.8 delete_coupon

API Docs

This method deletes a coupon from a cart and recalculates cart.

Required parameters: Cart id in param1, coupon line item id in param2

4.4.9 add_cart_line_item

API Docs

This method adds a product line item in a cart.

Required parameters: Cart id in param1, instance of ProductLineItemCreate in object

4.4.10 update_cart_line_item

API Docs

This method updates a product line item in a cart.

Required parameters: Cart id in param1, product line item id in param2, instance of ProductLineItemUpdate in object

4.4.11 delete_cart_line_item

API Docs

This method deletes a product line item from a cart.

Required parameters: Cart id in param1, product line item id in param2

4.4.12 add_order

API Docs

This method adds an order to a shop.

Required parameters: Cart id in param1

Note: Before creating an order, the billing address must be set in a cart. Billing address can be set after cart creation using the update_billing_address method.

4.4.13 update_order

API Docs

This method updates an order.

Required parameters: Order id in param1

4.4.14 update_billing_address

API Docs

This method updates the billing address for a cart.

Required parameters: Cart id in param1

4.4.15 delete_billing_address

API Docs

This method deletes the billing address from a cart.

Required parameters: Cart id in param1

4.4.16 update_shipping_address

API Docs

This method updates the shipping address for a cart.

Required parameters: Cart id in param1

4.4.17 delete_shipping_address

API Docs

This method deletes the shipping address from a cart.

Required parameters: Cart id in param1

4.5 Products

4.5.1 get_shop_info

API Docs

This method gets the public information of a shop, like name, slogan and logo.

Required parameters: none

4.5.2 get_categories

API Docs

This method gets the product categories of a shop.

Required parameters: none

4.5.3 get_category

API Docs

This method gets a single product category of a shop.

Required parameters: Category id in param1

4.5.4 get_currencies

API Docs

This method gets the currency information from a shop.

Required parameters: none

4.5.5 get_locales

API Docs

This method gets the locale information from a shop.

Required parameters: none

4.5.6 get_products

API Docs

This method gets all of the products from a shop.

Required parameters: none

4.5.7 get_product

API Docs

This method gets a single product from a shop.

Required parameters: Product id in param1

4.5.8 get_product_variations

API Docs

This method gets links to product variations.

Required parameters: Product id in param1

4.5.9 get_product_images

API Docs

This method gets product images with links to different sizes of the images.

Required parameters: Product id in param1

4.5.10 get_product_image_names

API Docs

This method gets product image names in the order they appear in a shop.

Required parameters: Product id in param1

4.5.11 get_product_custom_attributes

API Docs

This method gets the user-defined product attributes with their values.

Required parameters: Product id in param1

4.5.12 get_product_lowest_price

API Docs

This method gets the lowest price of all variations of a product.

Required parameters: Product id in param1

4.5.13 search_products

API Docs

This method searches products with a query.

Required parameters: Query string in query

4.5.14 get_shipping_methods

API Docs

This method gets the shipping methods of a shop.

Required parameters: none

4.5.15 get_shipping_method

API Docs

This method gets a single shipping method of a shop.

Required parameters: Shipping method id in param1

4.5.16 get_tax_classes

API Docs

This method gets the tax classes of a shop.

Required parameters: none

4.5.17 get_tax_class

API Docs

This method gets a single tax class of a shop.

Required parameters: Tax class id in param1

4.5.18 get_tax_model

API Docs

This method gets the tax model of a shop.

Required parameters: none

4.5.19 add_category

API Docs

This method adds a subcategory to existing main category.

Required parameters: Main category id in param1, instance of CategoryCreate in object

4.5.20 update_category

API Docs

This method updates a single category.

Required parameters: Category id in param1, instance of CategoryUpdate

Note: When updating a category, at least category id and category alias must be set. Category id must be the same that is set in param1, and alias can't be the same than some other category has. So, alias must be set always and it must be the same it was or something else that other categories have.

4.5.21 delete_category

API Docs

This method deletes a single category.

Required parameters: Category id in param1

4.5.22 get_subcategory_sequence

API Docs

This method gets the order of subcategories for the main category.

Required parameters: Main category id in param1

4.5.23 update_subcategory_sequence

API Docs

This method updates the order of subcategories.

Required parameters: Main category id in param1, instance of CategorySequenceUpdate in object

4.5.24 add_product

API Docs

This method adds a new product for a shop.

Required parameters: Instance of ProductCreate

4.5.25 update_product

API Docs

This method updates an existing product of a shop.

Required parameters: Product id in param1, instance of ProductUpdate

4.5.26 delete_product

API Docs

This method deletes a product from a shop.

Required parameters: Product id in param1

4.5.27 upload_product_image

API Docs

This method uploads an image for a product.

Required parameters: Product id in param1, image file in binary in data

Note: This doesn't set the uploaded image in the main image of a product, even if uploaded image is the first image of the product. It must be set using update_product method.

4.5.28 delete_product_image

API Docs

This method deletes a image from a product.

Required parameters: Product id in param1, image name in param2

4.5.29 update_product_image_sequence

API Docs

This method updates the order of product images.

Required parameters: Product id in param1, instance of ProductSlideshowSequenceUpdate in object

4.5.30 get_updated_products

API Docs

This method gets updated products by product attributes.

Required parameters: Product attribute in param1

Note: At the time of writing only stocklevel attribute works for this.

4.5.31 connect_category_and_product

API Docs

This method connects categories and products.

Required parameters: Category and product id in data

Note: There can be more than one category or product id when connecting them to each other. Category id and product id values can be a list of ids, too.

4.5.32 disconnect_product_and_category

API Docs

This method disconnects categories and products.

Required parameters: Category and product id in query

Note: There can be more than one category or product id when disconnecting them from each other. Category id and product id values can be a list of ids, too.

4.5.33 get_watched_products

API Docs

This method lists products that are watched by customers.

Required parameters: none

4.5.34 get_product_csv

API Docs

This method returns a CSV file with all products of the shop. This doesn't work at the time of writing.

Required parameters: none

4.6 Script tags

4.6.1 get_script_tags

API Docs

This method gets a list of all script tags for a shop.

Required parameters: none

4.6.2 add_script_tag

API Docs

This method adds a new script tag.

Required parameters: Instance of ScriptTagCreate in object

4.6.3 delete_script_tag

API Docs

This method deletes a script tag from a shop.

Required parameters: Script tag id in param1

CHAPTER 5

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:

5.1 Types of Contributions

5.1.1 Report Bugs

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

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

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

5.2 Get Started!

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

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

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

5.3 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 3.3, 3.4, 3.5, 3.6 and 3.7, and for PyPy. Check https://travis-ci.org/vilkasgroup/epages_client/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_epages_client
```

CHAPTER 6

Credits

6.1 Development Lead

- Pekka Piispanen <pekka@vilkas.fi>
- Tero Kotti <tero@vilkas.fi>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

7.1 0.2.0 (2020-02-11)

- Dropped Python 2 support

7.2 0.1.0 (2018-01-05)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search