Pakettikauppa Documentation

Release 0.1.6

Porntip Chaibamrung

Sep 03, 2023

Contents

Contents:

CHAPTER 1

Pakettikauppa

Client python modules for Pakettikauppa integration

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

1.1 Features

For merchants

- Search pickup points
- Create shipment included shipping label
- Get shipping label
- Get shipping status
- Get list of additional services
- Get list of shipping methods

For re-seller

- Create customer
- Get list of customers
- Update customer data

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 Pakettikauppa, run this command in your terminal:

```
$ pip install pakettikauppa
```

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

You can either clone the public repository:

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

Or download the tarball:

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

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

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

CHAPTER 3

Usage

To use Pakettikauppa in a project:

```
import pakettikauppa
```

CHAPTER 4

pakettikauppa

4.1 pakettikauppa package

4.1.1 Submodules

4.1.2 pakettikauppa.merchant module

This is a module for Pakettikauppa integration for merchants to use

The module provides below functionality:

1. Get shipping method list
2. Get additional service list
3. Search pickup points
4. Get shipment status
5. Create shipment
6. Get shipping label

class pakettikauppa.merchant.**PkMerchant**(*is_test_mode=0*, *api_key=None*, *secret=None*)

Bases: pakettikauppa.pakettikauppa.Pakettikauppa

create_additional_info_element(*root*, ***dict_data*)

Append additional info element to given root object. :param root: root XML object. :param dict_data: dictionary of additional info text :return:

create_shipment(***kwargs*)

Main function to send a request to Pakettikauppa to create shipment.

Parameters **kwargs** – See get_xml_shipment_req_data() function

Return dict_data See parse_xml_create_shipment_res() function

create_shipment_with_simple_data(***kwargs*)
Same as create_shipment() function expect the input parameter is in shorter format

This API send request data in XML format.

Parameters **kwargs** – see get_simple_test_data_create_shipment() function

Return dict_data See parse_xml_create_shipment_res() function

generate_additional_services_data(*list_data=None*)
Generate additional services data.

Parameters **list_data** – list of dictionary of additional services

Return list_data list of formatted dictionary of additional services

generate_parcels_data(*list_data=None*)
Generate parcels data.

Parameters **list_data** – list of dictionary of parcels

Return list_data list of formatted dictionary of parcels

get_additional_service_list(*language_code2='EN'*)
Get list of additional service for the account

Parameters **language_code2** – 2 letters of language code. Default value is 'EN'

Return list_data list of response data

get_api_config(*api_name=None*)
Constructs API configuration

Parameters **api_name** – string of API name

Return dict_data dictionary of configuration data

dict_data keys: api_post_url (string): post URL address api_key (string): API key api_secret (string): secret key

get_api_suffix(*param*)
Get API suffix for the API name :param api_name: string of API name :return api_suffix: string of API suffix

get_create_multi_parcels_shipment_test_data()
Main function to generate test data set for create shipment request call.

Return dict_data dictionary of request data

get_create_shipment_test_data()
Main function to generate test data set for create shipment request call.

Return dict_data dictionary of request data

get_create_shipment_test_req_data()
Generates test data set for create shipment request call.

Return dict_data dictionary of request data

get_one_additional_service_data(***kwargs*)
Generate one additional service data item.

Parameters **kwargs** – contain following keys: ServiceCode: string of additional service code Specifier: list of dictionary or dictionary of service details. See get_one_specifier_data() function for input format

Return dict_data dictionary for one additional service

get_one_parcel_data(***kwargs*)
Generate one parcel data.

Parameters **kwargs** – dictionary contain following keys: Reference: parcel reference PackageType: code of package type Weight: dictionary with 'weight_unit' and 'value' key. Volume: dictionary with 'unit' and 'value' key. InfoCode: Contents: product description ReturnService: return service code ContentLine: dictionary of parcel content data ParcelServices: parcel additional service (optional)

Return dict_data dictionary of parcel data

get_one_parcel_service_data(***kwargs*)
Generate one parcel service data.

Parameters **kwargs** – dictionary of parcel service code, contain following key: ServiceCode: parcel service code - string

Return dict_data dictionary of one parcel service

get_one_specifier_data(***kwargs*)
Generate one specifier data item

Parameters **kwargs** – contain following keys: name: attribute name for additional service value: attribute value

Return dict_data dictionary of one specifier

get_pickup_point_req_data(*api_key*, ***kwargs*)
Constructs request data for search pickup point API

Parameters

- **api_key** – string of API key
- **kwargs** –

Kwargs: postal_code (string): postal code (mandatory) country_code2 (string): 2 letters of country code i.e 'FI'. Default is 'FI' street_address (string): street address service_provider (string): Limits search to a single service provider, possible values: "Posti", "Matkahuolto" or "Db Schenker". Case insensitive values. max_result (integer): maximum number of search results. Default is 5

Return dict_data dictionary of request data

get_proper_req_data_create_shipment(***simple_dict_data*)
Generate low lever of request input data for create shipment API.

Parameters **simple_dict_data** – see output of get_simple_test_data_create_shipment() function

Return dict_data dictionary of request data

get_routing_key(*routing_id*)
Calculate routing key.

Parameters **routing_id** – string of routing ID

Return digest_string digest string of MD5 hash

get_shipment_status(*tracking_code*)
Get shipment status from Pakettikauppa.

Parameters **tracking_code** – string of tracking code for checking

Returns

get_shipment_status_req_data(*tracking_code*)
Construct request data for get shipment status API.

Parameters **tracking_code** – string of tracking code

Return dict_data dictionary of request data

get_shipping_label(***kwargs*)
Get shipping labels from Pakettikauppa. This API send request data in XML format.

Parameters **kwargs** – See get_xml_shipping_label_req_data() function

Returns See parse_xml_get_shipping_label_res() function

get_shipping_label_req_test_data()
Generate test request data for getting shipping label.

Return dict_data dictionary

get_shipping_method_list(*language_code2='EN'*)
Get list of available shipping method for the account

Parameters **language_code2** – 2 letters of language code. Default value is 'EN'

Return list_data list of response data

get_simple_test_data_create_shipment()
Generate more simplify version of request data of create shipment API. Output of this function should then pass to get_proper_req_data_create_shipment() function to get right data structure for request input data.

For full version see get_create_shipment_test_data() function.

Return dict_data dictionary of request data

get_xml_shipment_req_data(***kwargs*)
Construct XML string of request data for create shipment API

Parameters **kwargs** –

Kwargs:

eChannel: dictionary with following keys ROUTING: see _create_routing_elements() function

Shipment: see _create_shipment_elements() function

Return xml_string string of XML request data for create shipment API

get_xml_shipping_label_req_data(***kwargs*)
Construct XML string request data for getting shipping label

Parameters **kwargs** – contains following keys

eChannel: contains following keys ROUTING: dictionary of routing data PrintLabel: dictionary of tracking codes

ROUTING: Routing.Account: API key Routing.Key: MD5 hash string Routing.Id: could be order id or shop id or combination of both Routing.Name: could be order alias or shop alias or combination of both Routing.Time: string of current datetime with following format '%Y%m%d%H%M%S'

PrintLabel: responseFormat: expected response data format. Possible values are "File" and "inline", "File" is default. content: dictionary of tracking code with 'TrackingCode' as a key.

TrackingCode: could be dictionary with 'Code' key or list of dictionary of 'Code' key for asking multiple lab
Code: string of tracking code

Return xml_string

parse_xml_create_shipment_res (*xml_string*)
Constructs dictionary from response XML string

Parameters **xml_string** – string of XML data

Return dict_data dictionary of response data

dict_data contains following keys: status (integer): 1 = status OK message (string): response message from Pakettikauppa reference (dictionary): contains two keys: 'uuid' and 'value' trackingcode (dictionary): contains two keys: 'tracking_url' and 'value' (tracking code)

parse_xml_get_shipping_label_res (*xml_string*)
Construct dictionary from response data.

Parameters **xml_string** – XML string of response data

Returns dict_data: dictionary, contains below keys status (integer): 1 = operation OK message (string): response message PDFcontent (string): binary data of PDF content ContentEncoded (boolean): True = PDFContent data is encoded

search_pickup_points (***kwargs*)
Main method to search pickup points

Parameters **kwargs** – see get_pickup_point_req_data() function

Returns list of pickup point data

validate_input_params_create_shipment (***kwargs*)

validate_routing_data (***kwargs*)

pakettikauppa.merchant.**create_currency_element** (*root*, *value=None*)
Append 'Consignment.Currency' element to given root object.

Parameters

- **root** – root XML element object
- **value** – string of currency code i.e. 'EUR'

Returns

pakettikauppa.merchant.**create_merchandise_value_element** (*root*, *value=None*)
Append 'Consignment.Merchandisevalue' element to given root object.

Parameters

- **root** – root XML element object
- **value** – string of merchandise value

Returns

pakettikauppa.merchant.**create_product_element** (*root*, *product_code*)
Append 'Consignment.Product' element to given root object.

Parameters

- **root** – root XML object
- **product_code** – string of Posti's product code

Returns

pakettikauppa.merchant.**create_reference_element**(*root*, *value=None*)
Append 'Consignment.Reference' element to given root object.

Parameters

- **root** – root XML object
- **value** – string of reference number

Returns

pakettikauppa.merchant.**decode_pdf_content**(*encoded_pdf_content*)

pakettikauppa.merchant.**write_pdf_to_file**(*target_file_path*, *pdf_content*)

4.1.3 pakettikauppa.pakettikauppa module

Pakettikauppa app

This module provides base functionality for Pakettikauppa integration

class pakettikauppa.pakettikauppa.**Pakettikauppa**(*is_test_mode=0*)
Bases: object

Base class for Pakettikauppa integration.

get_api_end_point()
Get API base end point string

Return base_end_point API base end point string

get_hash_sha256(*secret_key*, ***kwargs*)
Calculate SHA256 digest string.

Parameters

- **secret_key** – string of secret key
- **kwargs** – dictionary of parameters for caluclation

Return digest_string digest string

get_logger()
Get logger object.

Returns

get_md5_hash(*api_key=None*, *secret_key=None*, *routing_id=None*)
Calculate MD5 digest string.

Parameters

- **api_key** – string of API key
- **secret_key** – string of secret key
- **routing_id** – string of routing id

Return digest_string digest string

get_post_url(*api_suffix=None*)
Get API post URL address

Parameters **api_suffix** – string of API suffix

Return api_post_url (string) Post URL

logger = None

parse_res_json_data(*res_obj*)
Parse response JSON data (Not yet completely implemented)

Parameters **res_obj** – response object

Returns

parse_res_to_list(*res_obj=None*)
Parse response object to list data.

Parameters **res_obj** – response object

Return list_data list data of response data from Pakettikauppa

send_request(*send_method='POST'*, *_api_post_url=None*, *req_input=None*, ***headers*)
Send a request to Pakettikauppa.

Parameters

- **send_method** – type of request method. Possible value are 'POST' and 'GET', 'POST' is default value.
- **_api_post_url** – string of post URL
- **req_input** – request input data
- **headers** – dictionary of header data

Return res_obj response object

set_logger()
Set logger object.

Returns

exception pakettikauppa.pakettikauppa.**PakettikauppaException**
Bases: exceptions.Exception

pakettikauppa.pakettikauppa.**check_api_name**(*in_function*)
Validate API name :type in_function: input function

4.1.4 pakettikauppa.reseller module

This is a module for Pakettikauppa integration for resellers to use

The module provides below functionality:

1. Create customer
2. Update customer
3. Get list of customer
4. De-activate customer

class pakettikauppa.reseller.**PkReseller**(*is_test_mode=0*, *api_key=None*, *secret=None*)
Bases: pakettikauppa.pakettikauppa.Pakettikauppa

Pakettikauppa reseller class is mean for reseller.

accept_payment_service_provider = ('CHECKOUT', 'CREDIT_CARD')

clean_up_phone_data(*phone_string=None*)
Remove none-digit data from phone number.

Parameters **phone_string** – string of phone number

Returns

create_customer(***kwargs*)
Create customer in Pakettikauppa's system.

Parameters **kwargs** – See get_create_customer_req_data() function

Return list_data list of response data

deactivate_customer(*customer_id*)
De-activate customer account in Pakettikauppa's system.

Parameters **customer_id** – Pakettikauppa's customer id

Return list_data list of response data

get_api_config(*api_name=None*)
Constructs API configuration

Parameters **api_name** – string of API name

Return dict_data dictionary of configuration data

dict_data keys: api_post_url (string): post URL address api_key (string): API key api_secret (string): secret key

get_api_suffix(*param*)
Get API suffix for given API name.

Parameters **api_name** – string of API name

Return api_suffix string of API suffix

get_create_customer_req_data(***kwargs*)
Construct request data for create customer API.

Parameters **kwargs** – contains following key name: customer name business_id: VAT ID payment_service_provider: possible values are 'CHECKOUT' and 'CREDIT_CARD', can be empty string psp_merchant_id: Required if payment_service_provider is CHECKOUT. Value is merchant id in the Checkout service. marketing_name: optional field street_address: street address post_office: city name postcode: postal code country: country name phone: phone number email: email address contact_person_name: contact person name contact_person_phone: contact person phone number contact_person_email: contact person email address customer_service_phone: customer service phone number customer_service_email: customer service email address

Return dict_data dictionary of request data

get_customer_list()
Get list of customer for your account.

Return list_data list of response data

get_update_customer_req_data(*customer_id*, ***kwargs*)
Construct request data for updating customer details.

Parameters

- **customer_id** – Pakettikauppa's customer id

- **kwargs** – attributes that you wish to update. See possible keys from get_create_customer_req_data()

Return dict_data dictionary of request data

update_customer(*customer_id=None*, ***kwargs*)
Update customer details in Pakettikauppa's system.

Parameters

- **customer_id** – Pakettikauppa's customer id
- **kwargs** – attributes that you wish to update. See possible keys from get_create_customer_req_data()

Returns

4.1.5 Module contents

Top-level package for Pakettikauppa.

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/Pakettikauppa/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

Pakettikauppa could always use more documentation, whether as part of the official Pakettikauppa 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/Pakettikauppa/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 *Pakettikauppa* for local development.

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

```
$ git clone git@github.com:vilkasgroup/Pakettikauppa.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 Pakettikauppa
$ cd Pakettikauppa/
$ 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 pakettikauppa 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/vilkasgroup/Pakettikauppa/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

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

CHAPTER 6

Credits

6.1 Development Lead

- Porntip Chaibamrung <tipi@vilkas.fi>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

7.1 0.1.6 (2019-05-07)

- Updated pip and tox in 'requirements_dev.txt' file.
- Updated Pipfile.lock

7.2 0.1.5 (2019-04-02)

- Fixed security issue in urllib3 package version in Pipfile.

7.3 0.1.4 (2019-04-02)

- Fixed security issue in Jinja2 package version in Pipfile.

7.4 0.1.3 (2019-04-02)

- Update version number due to changes in 'requirements_dev.txt' file

7.5 0.1.2 (2019-07-01)

- Fix Pip files and requirement files for fixing security issue in pyyaml module

7.6 0.1.1 (2018-11-02)

- Fix related Pip files for requests module
- Fixed test data generating function in merchant.py
- Fixed tracking code for testing in test_get_shipping_label.py

7.7 0.1.0 (2018-01-22)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Index

P