
Fakturownia Documentation

Release 0.2.1

Janusz Skonieczny

Jan 18, 2018

Contents

1	Fakturownia	3
1.1	Features	4
1.2	Quickstart	4
1.3	Running Tests	5
1.4	Help wanted	5
1.5	Credits	5
2	Installation	7
2.1	Stable release	7
2.2	From sources	7
3	Usage	9
4	Api docs	11
4.1	fakturownia package	11
5	Contributing	15
5.1	Types of Contributions	15
5.2	Get Started!	16
5.3	Pull Request Guidelines	17
5.4	Tips	17
6	Credits	19
6.1	Development Lead	19
6.2	Contributors	19
7	History	21
7.1	0.1.0 (2018-01-03)	21
7.2	0.1.2 (2018-01-10)	21
7.3	0.2.1 (2018-01-12)	21
7.4	0.2.2 (2018-01-18)	21
8	Indices and tables	23
	Python Module Index	25

Contents:

CHAPTER 1

Fakturownia

Python client library for the Fakturownia API

Quickstart

Running Tests

Help wanted

Credits

1.1 Features

- Invoice CRUD plus *send_by_email* and 'mark_paid'
- EU member states VAT rate helpers for B2C transactions
- ApiClient CRUD
- Payments CRUD
- Products CRUD
- Warehouse documents CRUD
- Accounts management

Please refer to the [fakturownia APIs](#) for full API features

1.2 Quickstart

Install Fakturownia

```
pip install fakturownia python
```

Play with [fakturownia APIs](#) in python interpreter:

```
>>> import os
>>> os.environ.get('FAKTUROWNIA_API_TOKEN', 'Missing key')
'...'
>>> from fakturownia import get_api_client
>>> api = get_api_client()
>>> invoice = api.invoices.create(
...     seller_name='Kabaret Starszych Panów',
...     buyer_name='Odrażający drab',
...     positions=[{
...         'name': 'Smolna szczapa',
...         'quantity': 5,
...         'total_price_gross': 7.33,
...     }],
... )
>>> invoice.view_url
'...'
```

This instance is only partially updated as create returns only subset of data properties, to get all we need to update our instance.

If you have payments enabled you can call get to fetch all data and check payment_url:

```
>>> invoice.get()
<fakturownia.endpoints.Invoice object at 0x...>
>>> invoice.payment_url
'...'
```

We can mark this invoice as paid:

```
>>> invoice.mark_paid()
<fakturownia.endpoints.Invoice object at 0x...>
```

You can chain your calls:


```
>>> invoice.put(buyer_email='kominek@niepodam.pl').send_by_email()
<fakturownia.endpoints.Invoice object at 0x...>
```

You can play and test your scenarios with factories:

```
pip install fakturownia[factories]
python
```

Now you can do this:

```
>>> from fakturownia.factories import InvoiceFactory
>>> InvoiceFactory(api_client='<your api key here>', kind='proforma').post().get().
    ↳ payment_url
    '...'
```

Also checkout VAT tax normalization based on [EU country specific VAT rates](#):

```
>>> InvoiceFactory(
...     api_client=api,
...     seller_country='PL',
...     buyer_country='DE',
...     buyer_tax_no=None,
... ).normalize_vat().post().view_url
'...'
```

Neat! :)

1.3 Running Tests

Does the code actually work?

```
pipenv install --dev
pipenv shell
tox
```

We recommend using [pipenv](#) but a legacy approach to creating virtualenv and installing requirements should also work. Please install *requirements/base.txt* and *requirements/development.txt* to setup virtual env for testing and development.

1.4 Help wanted

This library is not yet complete. It does what was needed by up to date contributors, but more can be done. You can implement new api endpoints and write test for them, it's actually straightforward and new classes will be simple, but tests need some effort. We are lazy test writers and because we don't want to compromise coverage so we postponed new apis until someone would want to write test.

If want to help please refer to the [contributing section](#) in the docs for more info.

1.5 Credits

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

2.1 Stable release

To install Fakturownia, run this command in your terminal:

```
$ pip install fakturownia
```

This is the preferred method to install Fakturownia, 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

You can either clone the public repository:

```
$ git clone git@github.com:wooyek/fakturownia-python.git
```

Or download the download source from [project website](#). Once you have a copy of the source, you can install it with:

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


CHAPTER 3

Usage

To use Fakturownia in a project:

```
import fakturownia
```


4.1 fakturownia package

4.1.1 Submodules

4.1.2 fakturownia.base module

```
class fakturownia.base.BaseEndpoint (api_client)
    Bases: object

    create (**kwargs)

class fakturownia.base.BaseModel (api_client, **kwargs)
    Bases: object

    delete ()

    get ()

    get_endpoint (extra="")

    get_raw_data ()

    post (**kwargs)

    prepare_post_data (**kwargs)

    put (**kwargs)

    update_data (**kwargs)
```

4.1.3 fakturownia.cli module

4.1.4 fakturownia.core module

class fakturownia.core.**ApiClient** (*api_token, base_url=None, request_timeout=10*)

Bases: object

Fakturownia API client

Here is an example of how to create an invoice. But first make sure you have set *FAKTUROWNIA_API_TOKEN* environment variable.

```
>>> import os
>>> os.environ.get('FAKTUROWNIA_API_TOKEN', 'Missing key')
'...'
>>> from fakturownia import get_api_client
>>> api = get_api_client()
>>> invoice = api.invoices.create(
...     seller_name='Kabaret Starszych Panów',
...     buyer_name='Odrażający drab',
...     positions=[{
...         'name': 'Smolna szczapa',
...         'quantity': 5,
...         'total_price_gross': 7.33,
...     }],
... )
```

This instance is only partially updated as create returns only subset of data properties, to get all we need to update our instance.

This shows payment_url but only if you have payments enabled.

```
>>> invoice.get()
<fakturownia.endpoints.Invoice object at 0x...>
>>> invoice.payment_url
'...'
```

We can mark this invoice as paid.

```
>>> invoice.mark_paid()
<fakturownia.endpoints.Invoice object at 0x...>
```

You can chain your calls

```
>>> invoice.put(buyer_email='kominek@niepodam.pl').send_by_email()
<fakturownia.endpoints.Invoice object at 0x...>
```

base_url

build_headers (*items=None*)

build_payload (*data=None*)

delete (*endpoint, data=None, headers=None*)

get (*endpoint, data=None, headers=None*)

post (*endpoint, data=None, headers=None*)

put (*endpoint, data=None, headers=None*)

request (*method, endpoint, params=None, payload=None, headers=None*)

request_factory (**args, **kwargs*)

validate_response (*response*)

`fakturownia.core.get_api_client()`

Factory function for Fakturownia API client with configuration options taken from environment

Returns ApiClient instance

4.1.5 fakturownia.endpoints module

class `fakturownia.endpoints.Client` (*api_client, **kwargs*)

Bases: `fakturownia.base.BaseModel`

create_invoice (***kwargs*)

class `fakturownia.endpoints.Clients` (*api_client*)

Bases: `fakturownia.base.BaseEndpoint`

model

alias of `Client`

class `fakturownia.endpoints.DateProperty` (*name*)

Bases: `object`

class `fakturownia.endpoints.Invoice` (*api_client, **kwargs*)

Bases: `fakturownia.base.BaseModel`

issue_date

mark_paid ()

normalize_vat (*default_rate=None, intra_eu_vat_rate='np'*)

This is a common business logic that maybe helpful in handling EU to EU invoicing

payment_to

sell_date

send_by_email ()

set_tax_on_positions (*rate*)

class `fakturownia.endpoints.Invoices` (*api_client*)

Bases: `fakturownia.base.BaseEndpoint`

model

alias of `Invoice`

4.1.6 fakturownia.exceptions module

exception `fakturownia.exceptions.ClientException` (*message, data=None, *args*)

Bases: `fakturownia.exceptions.FakturowniaException`

Base client exception with data attribute

exception `fakturownia.exceptions.FakturowniaException`

Bases: `Exception`

exception `fakturownia.exceptions.HttpException` (*message, response, data=None, *args*)

Bases: `fakturownia.exceptions.ClientException`

4.1.7 fakturownia.factories module

4.1.8 fakturownia.settings module

Utilities for settings loading

Since `envparse` modifies `os.environ` on while loading `.env` files we need and alternate solution that wont't touch `os.environ` by default.

`fakturownia.settings.get_default_env_file()`

Returns `secrets.env` from project root, should not be used outside testing

`fakturownia.settings.get_env_from_file(path)`

`fakturownia.settings.get_key_from_file(env_file=None)`

This is a utility to ease testing with `secrets.env` file present or not

`fakturownia.settings.parse_env(content)`

4.1.9 Module contents

Top-level package for Fakturownia.

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/wooyek/fakturownia-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.

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

Fakturownia could always use more documentation, whether as part of the official Fakturownia 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/wooyek/fakturownia-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 :)

5.2 Get Started!

Ready to contribute? Here's how to set up *fakturownia-python* for local development.

1. Fork the *fakturownia-python* repo on github.com
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/fakturownia-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 fakturownia-python
$ cd fakturownia-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 src tests
$ tox -e check
$ pytest
$ 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 versions mentioned in tox.ini file. Check https://travis-ci.org/wooyek/fakturownia-python/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_*
```


CHAPTER 6

Credits

6.1 Development Lead

- @wooyek: Janusz Skonieczny <js+pypi@bravelabs.pl>

6.2 Contributors

None yet. Why not be the first?

7.1 0.1.0 (2018-01-03)

- First release on PyPI.

7.2 0.1.2 (2018-01-10)

- endpoints indexing
- Robust testing with mocked and real interchangeable backends .

7.3 0.2.1 (2018-01-12)

- EU member states VAT rate helpers for B2C transactions
- FAKTUROWNIA_TIMEOUT setting for request timeout

7.4 0.2.2 (2018-01-18)

- Payload data cleanup from readonly fields so get().put() is possible

CHAPTER 8

Indices and tables

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

b

`fakturownia.base`, [11](#)

c

`fakturownia.core`, [12](#)

e

`fakturownia.endpoints`, [13](#)

`fakturownia.exceptions`, [13](#)

f

`fakturownia`, [14](#)

s

`fakturownia.settings`, [14](#)

A

ApiClient (class in fakturownia.core), 12

B

base_url (fakturownia.core.ApiClient attribute), 12

BaseEndpoint (class in fakturownia.base), 11

BaseModel (class in fakturownia.base), 11

build_headers() (fakturownia.core.ApiClient method), 12

build_payload() (fakturownia.core.ApiClient method), 12

C

Client (class in fakturownia.endpoints), 13

ClientException, 13

Clients (class in fakturownia.endpoints), 13

create() (fakturownia.base.BaseEndpoint method), 11

create_invoice() (fakturownia.endpoints.Client method), 13

D

DateProperty (class in fakturownia.endpoints), 13

delete() (fakturownia.base.BaseModel method), 11

delete() (fakturownia.core.ApiClient method), 12

F

fakturownia (module), 14

fakturownia.base (module), 11

fakturownia.core (module), 12

fakturownia.endpoints (module), 13

fakturownia.exceptions (module), 13

fakturownia.settings (module), 14

FakturowniaException, 13

G

get() (fakturownia.base.BaseModel method), 11

get() (fakturownia.core.ApiClient method), 12

get_api_client() (in module fakturownia.core), 13

get_default_env_file() (in module fakturownia.settings), 14

get_endpoint() (fakturownia.base.BaseModel method), 11

get_env_from_file() (in module fakturownia.settings), 14

get_key_from_file() (in module fakturownia.settings), 14

get_raw_data() (fakturownia.base.BaseModel method), 11

H

HttpException, 13

I

Invoice (class in fakturownia.endpoints), 13

Invoices (class in fakturownia.endpoints), 13

issue_date (fakturownia.endpoints.Invoice attribute), 13

M

mark_paid() (fakturownia.endpoints.Invoice method), 13

model (fakturownia.endpoints.Clients attribute), 13

model (fakturownia.endpoints.Invoices attribute), 13

N

normalize_vat() (fakturownia.endpoints.Invoice method), 13

P

parse_env() (in module fakturownia.settings), 14

payment_to (fakturownia.endpoints.Invoice attribute), 13

post() (fakturownia.base.BaseModel method), 11

post() (fakturownia.core.ApiClient method), 12

prepare_post_data() (fakturownia.base.BaseModel method), 11

put() (fakturownia.base.BaseModel method), 11

put() (fakturownia.core.ApiClient method), 12

R

request() (fakturownia.core.ApiClient method), 12

request_factory() (fakturownia.core.ApiClient method), 13

S

`sell_date` (fakturownia.endpoints.Invoice attribute), [13](#)
`send_by_email()` (fakturownia.endpoints.Invoice
method), [13](#)
`set_tax_on_positions()` (fakturownia.endpoints.Invoice
method), [13](#)

U

`update_data()` (fakturownia.base.BaseModel method), [11](#)

V

`validate_response()` (fakturownia.core.ApiClient
method), [13](#)