
Basket Client Documentation

Release v0.3.10

Mozilla

April 04, 2016

1 About The Basket Client	3
2 Contents	5

This documentation explains how to install and use the Basket Client.

About The Basket Client

This is a client for Mozilla's email subscription service, basket. Basket is not a real subscription service, but it talks to a real one and we don't really care who/what it is.

2.1 Installation & Settings

2.1.1 Install

```
$ pip install basket-client
```

2.1.2 Settings

BASKET_URL

URL to basket server, e.g. `https://basket.mozilla.org`

Default: `http://localhost:8000`

The URL must not end with `/`. Basket-client will add `/` if needed.

BASKET_API_KEY The API Key granted to you by [the mozilla.org developers](https://mozillalabs.com/) so that you can use the `lookup_user` method with an email address.

BASKET_TIMEOUT

The number of seconds basket client should wait before giving up on the request.

Default: 10

If you're using [Django](https://www.djangoproject.com/) you can simply add these settings to your `settings.py` file. Otherwise basket-client will look for these values in an environment variable of the same name.

2.1.3 Tests

To run tests:

```
$ python setup.py test
```

2.2 Usage

Do you want to subscribe people to Mozilla's newsletters? All you need to do is:

```
import basket

basket.subscribe('<email>', '<newsletter>', <kwargs>)
```

You can pass additional fields as keyword arguments, such as format and country.

Are you checking to see if a user was successfully subscribed? You can use the `lookup_user` method like so:

```
import basket

basket.lookup_user(email='<email>', api_key='<api_key>')
```

And it will return full details about the user. `<api_key>` is a special token that grants you admin access to the data. Check with [the mozilla.org developers](https://developer.mozilla.org) to get it.

On most errors, `BasketException` will be raised. The `code` attribute on the exception object will contain a numeric code indicating the problem, and the `desc` attribute will have a short English description of it. (Use the `code` attribute to determine which error happened, but you can use `desc` in log messages etc.)

Example:

```
from basket import errors, some_basket_call

try:
    rc = some_basket_call(args)
except BasketError as e:
    if e.code == errors.BASKET_INVALID_EMAIL:
        print "That email address was not valid"
    else:
        log.exception("Some basket error (%s)" % e.desc)
```

The error codes are defined in `basket.errors`. New ones can be added anytime, but to start with, the errors are:

```
BASKET_NETWORK_FAILURE
BASKET_INVALID_EMAIL
BASKET_UNKNOWN_EMAIL
BASKET_UNKNOWN_TOKEN
BASKET_USAGE_ERROR
BASKET_EMAIL_PROVIDER_AUTH_FAILURE
BASKET_AUTH_ERROR
BASKET_SSL_REQUIRED
BASKET_INVALID_NEWSLETTER
BASKET_INVALID_LANGUAGE
BASKET_EMAIL_NOT_CHANGED
BASKET_CHANGE_REQUEST_NOT_FOUND

# If you get this, report it as a bug so we can add a more specific
# error code.
BASKET_UNKNOWN_ERROR
```

2.2.1 Usage in tests

You may wish to test your site without the overhead and fragility of calling an external service. To test your subscription forms without actually hitting the Basket service simply use a special set of email addresses:

- `success@example.com` will cause the basket client to return a success message.
- `failure@example.com` will cause the basket client to raise a `BasketError` exception.

These will allow you to test the functioning of your form handling code without making any actual network calls.

2.3 Change Log

2.3.1 v0.3.11

- Add option to send the source IP to basket service for rate limiting purposes for the `subscribe` and `send_sms` functions.

2.3.2 v0.3.10

- Set api key on subscribe call when `sync=Y`

2.3.3 v0.3.9

- Add numeric error codes.

2.3.4 v0.3.8

- Add the `start_email_change` and `confirm_email_change` functions.

2.3.5 v0.3.7

- Add the `lookup_user` function.
- Add the `BASKET_API_KEY` setting.
- Add the `BASKET_TIMEOUT` setting.

2.3.6 v0.3.6

- Add the `confirm` function.

2.3.7 v0.3.5

- Add tests

2.3.8 v0.3.4

- Fix issue with calling `subscribe` with an iterable of newsletters.
- Add `request` function to those exposed by the `basket`` module.

2.3.9 v0.3.3

- Add `get_newsletters` API method for information on currently available newsletters.
- Handle Timeout exceptions from requests.