python-leetchi Documentation

Release dev

2012, Florent Messa and contributors

May 02, 2013

CONTENTS

python-leetchi is a client library to work with mangopay api (formely leetchi).

You can report bugs and discuss features on the issues page.

CHAPTER

ONE

REFERENCE

For further details see the reference documentation:

1.1 Installation

1. Either check out python-leetchi from GitHub or to pull a release off PyPI

   ```
   pip install python-leetchi
   ```

1.1.1 Dependencies

python-leetchi requires requests, M2Crypto and blinker to work.

If you are installing it with pip, all dependencies will be installed for you.

1.2 Usage

1.2.1 Creating a handler

To manipulate resources (Users, Wallets, etc.) from this api you will have to instanciate a new handler which is basically a connection authentification.

To create a new handler, you have to provide several parameters.

API_PARTNER_ID

This is the partner identifier used by mangopay to identify you.

API_PRIVATE_KEY

This is the certificat used in each requests.

API_PRIVATE_KEY_PASSWORD

This is the password linked to the certificat.

API_HOST

The host used to call the API. We will see later when you are creating a new handler you can choose between multiple environment hosts already registered.

Let's get to work, we will create our first handler with the sandbox host

```
private_key = '/path/to/the/private/key/file'
partner_id = 'dummy'
private_key_password = '$ecret'

from leetchi.api import LeetchiAPI

handler = LeetchiAPI(partner_id,
                     private_key,
                     private_key_password,
                     sandbox=True)
```

Now we have a new handler which is using the **'sanbox host'_**.

If you are not specifying that you are using the sandbox host nor an existing host, it will use the production host.

Specific host for mangopay endpoint

```
handler = LeetchiAPI(partner_id,
                     private_key,
                     private_key_password,
                     host='http://dummy.api.prod.leetchi.com')
```

1.2.2 Using resources

To manipulate resources, this library is heavily inspired from peewee, so every operations will be like manipulating a ORM.

For required parameters you have to refer to the reference api.

Users

Creating a new user

```
from leetchi.resources import User
from datetime import date

user = User(first_name='Florent',
            last_name='Messa',
            email='florent@dummy.host',
            ip_address='127.0.0.1',
            tag='new user',
            birthday=date.today(),
            nationality='FR')

user.save(handler) # save the new user

print user.get_pk() # retrieve the primary key
```

Retrieving an existing user

```
user = User.get(1)

print user.first_name # Florent
```

Detecting user that does not exist

```
try:
    user = User.get(2, handler)
except User.DoesNotExist:
    print 'The user 2 does not exist'
```

Wallets

Affecting a wallet to an existing user

```
user = User.get(1, handler)

from leetchi.resources import Wallet

wallet = Wallet(tag='wallet for user n.1',
                name='Florent Messa wallet',
                description='A new wallet for Florent Messa',
                raising_goal_amount=1200,
                users=[user])
wallet.save(handler) # save the new wallet

print wallet.get_pk() # 1
```

Retrieving all wallets for an existing user

```
user = User.get(1, handler)

wallet_list = user.wallet_set
```

By default all amount are in centimes but this library provides an helper to quickly convert an amount to a readable one

```
print wallet.raising_goal_amount # 1200
print wallet.raising_goal_amount_converted # 12.00
```

Contributions

A contribution a the only way to put money on a wallet, with the mangopay API you can also put money a user wallet.

Creating a new contribution for a dedicated wallet

```
from leetchi.resources import Contribution, Wallet, User

user = User.get(1, handler)
wallet = Wallet.get(1, handler)

contribution = Contribution(user=user,
                            wallet=wallet,
                            amount=1000,
                            return_url='http://my-website/back-url',
                            client_fee_amount=0)
contribution.save(handler)
```

```
print contribution.is_success() # False
print contribution.is_succeeded # False
print contribution.is_completed # False
```

Creating a new contribution for a personal wallet

```
contribution = Contribution(user=user,
                            wallet=0,
                            amount=1000,
                            return_url='http://my-website/back-url',
                            client_fee_amount=0)
contribution.save(handler)
```

Transfers

Creating a transfer from a personal wallet to another wallet

```
from leetchi.resources import User, Transfer, Wallet

user = User.get(1, handler)

beneficiary = User.get(2, handler)

beneficiary_wallet = Wallet.get(2, handler)

transfer = Transfer(payer=user,
                    beneficiary=beneficiary,
                    payer_wallet_id=0,
                    beneficiary_wallet=beneficiary_wallet,
                    amount=1000)
transfer.save(handler)

print transfer.get_pk() # 1

beneficiary_wallet = Wallet.get(2, handler)

print beneficiary_wallet.collected_amount # 1000
```

Transfer refunds

If you want to cancel a transfer and move back the money from one wallet to another

```
from leetchi.resources import TransferRefund, Transfer, User

user = User.get(1, handler)
transfer = Transfer.get(1, handler)

transfer_refund = TransferRefund(user=user, transfer=transfer)

wallet = transfer.beneficiary_wallet

print wallet.collected_amount # 1000
print wallet.remaining_amount # 0

print user.personal_wallet_amount # 1000
```

Refunds

If you want to refund a contribution and move back the money from a wallet to a credit card account

```
from leetchi.resources import Contribution, User, Refund

user = User.get(1, handler)
contribution = Contribution.get(1, handler)

refund = Refund(contribution=contribution,
                user=user)
refund.save(handler)
```

Operations

Retrieving all operations for a dedicated user

```
from leetchi.resources import User

user = User.get(1, handler)

operation_list = user.operation_set
```

1.3 Signals

1.4 Indices and tables

- *genindex*
- *modindex*
- *search*

CHAPTER

TWO

ISSUES

For any bug reports and feature requests, please use the Github issue tracker.