qualpay-python Documentation

Release 1.0.0

Derek Payton

August 10, 2015

Contents

Python bindings for Qualpay.

Author Derek Payton

Version 1.0.0

License MIT

Source github.com/dmpayton/qualpay-python

Docs qualpay-python.readthedocs.org

Contents:

CHAPTER 1

Getting started

1.1 Installation

```
$ pip install qualpay
```

1.2 Authentication

```
import qualpay

qualpay.merchant_id = '<your merchant id>'
qualpay.security_key = '<your security key>'
```

1.3 Using the test gateway

To use the test payment gateway (such as during development), set `qualpay.base_endpoint` to the correct URL:

```
qualpay.base_endpoint = 'https://api-test.qualpay.com'
```

CHAPTER 2

The payment gateway

2.1 Authorization

```
qualpay.authorize(
    card_number='4111111111111111'
    exp_date='0120',
    cvv2='111',
    tran_amt=10
)
```

2.2 Capture

```
qualpay.capture(
    pg_id='8af556ae480811e484b20c4de99f0aaf'
)
```

2.3 Sale

```
qualpay.sale(
    card_number='4111111111111111'
    exp_date='0120',
    cvv2='111',
    tran_amt=10
)
```

CHAPTER 3

The card object

The `qualpay.Card` object allows you to validate and charge credit cards. It's essentially a utility class that wraps the `qualpay.PaymentGateway` and provides additional helpers for validating card information before sending it off to the API.

```
>>> card = qualpay.Card(
    number='4111 1111 1111 1111',
    exp_month=1,
    exp_year=2020,
    cvv2='111'
)
>>> card.is_expired
False
>>> card.is_valid
True
>>> card.authorize(1)
{'rcode': '000', 'rmsg': 'Approved T63362', 'pg_id': '8af556ae480811e484b20c4de99f0aaf'}
```

3.1 Gateway helpers

3.1.1 Authorization

Create an authorization of $10.00:

```
>>> card.authorize(10)
{
    'rcode': '000',
    'rmsg': 'Approved T63362',
    'pg_id': '8af556ae480811e484b20c4de99f0aaf'
}
```

3.1.2 Sale

Immediately charge $10.00:

```
>>> card.sale(10)
{
    'rcode': '000',
    'rmsg': 'Approved T42926',
```

```
    'pg_id': 'daab5fff480811e484b20c4de99f0aaf'
}
```

3.1.3 Verify

Verify the card information:

```
>>> card.verify()
{
    'rcode': '085',
    'rmsg': 'No reason to decline T19093',
    'pg_id': '299cf38f29b111e5b1460a12fcf6f1a3',
    'auth_cvv2_result': 'M',
    'auth_code': 'T19093'
}
```

3.1.4 Tokenize

Tokenize the card:

```
>>> card.tokenize()
{
    'rcode': '000',
    'rmsg': 'Token request complete',
    'pg_id': '88a4099c480711e4ac850c4de99f0aaf',
    'card_id': '88b4363 d480711e4ac850c4de99f0aaf'
}
```

CHAPTER 4

Contributing

4.1 Reporting bugs

Perhaps the easiest way to contribute to qualpay-python is to report any bugs you run into on the github issue tracker.

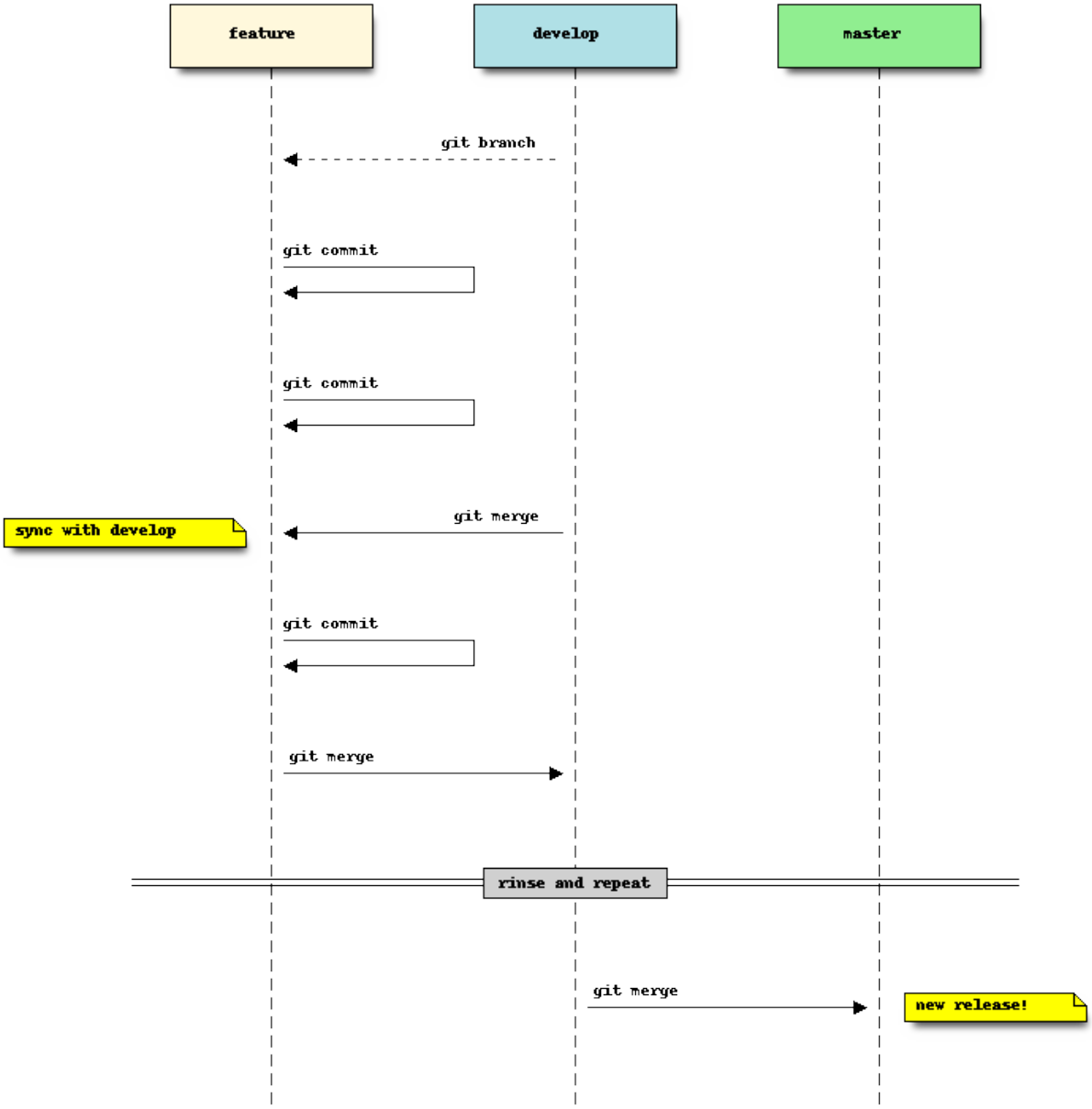
Useful bug reports are ones that get bugs fixed. A useful bug report normally has two qualities:

1. **Reproducible.** If your bug is not reproducible it will never get fixed. You should clearly mention the steps to reproduce the bug. Do not assume or skip any reproducing step. Described the issue, step-by-step, so that it is easy to reproduce and fix.
2. **Specific.** Do not write a essay about the problem. Be Specific and to the point. Try to summarize the problem in minimum words yet in effective way. Do not combine multiple problems even they seem to be similar. Write different reports for each problem.

4.2 Writing code

Our workflow is based on Vincent Driessen's successful git branching model:

- The `master` branch is our current release
- The `develop` branch is what all pull requests should be based against
- Feature branches are where new features, both major and minor, should be developed.



git-flow is a git plugin that helps facilitate this branching strategy. It's not required, but can help make things a bit easier to manage. There is also a good write up on using git-flow.

We also request that git commit messages follow the standard format.

4.3 Testing

Continuous integration provided by Travis CI.

4.3.1 Test requirements

See *requirements.txt*:

```
pytest
pytest-cov
pytest-pep8
pytest-pythonpath
requests
responses
sphinx
sphinxcontrib-seqdiag
tox
```

4.3.2 Running the tests

Once your requirements are installed, the unit tests can be run with:

```
$ py.test tests/ --cov qualpay --cov-report term-missing --pep8 qualpay

...

==================== 15 passed in 0.15 seconds ====================
```

For testing against different Python versions, we use Tox.

```
$ tox

...

_____________ summary _____________
py27: commands succeeded
py34: commands succeeded
congratulations :)
```

4.4 Submit a pull request

You've done your hacking and are ready to submit your patch. Great! Now it's time to submit a pull request to our issue tracker on Github.

Important: Pull requests are not considered complete until they include all of the following:

- **Code** that conforms to PEP8.
- **Unit tests** that pass locally and in our CI environment.
- **Documentation** updates on an as needed basis.

CHAPTER 5

Reference

5.1 Payment gateway functions

qualpay.**authorize**(***kwargs*)
An authorization message is used to send cardholder data to the issuing bank for approval. An approved transaction will continue to be open until it expires or a capture message is received. Authorizations are automatically voided if they are not captured within 28 days, although most issuing banks will release the hold after 24 hours in retail environments or 7 days in card not present environments.

qualpay.**verify**(***kwargs*)
A verify message is used to send cardholder data to the issuing bank for validation. A verify message will return success if the cardholder information was verified by the issuer. If the AVS or CVV2 field is included in the message, then the AVS or CVV2 result code will be returned in the response message.

qualpay.**capture**(***kwargs*)
A capture message is used to capture a previously authorized transaction using the payment gateway identifier returned by the authorization message. A capture may be completed for any amount up to the authorized amount.

qualpay.**sale**(***kwargs*)
A sale message is used to perform the function of an authorization and a capture in a single message. This message is used in retail and card not present environments where no physical goods are being shipped.

qualpay.**void**(***kwargs*)
A void message is used to void a previously authorized transaction. Authorizations can be voided at any time. Captured transactions can be voided until the batch is closed. The batch close time is configurable and by default is 11 PM Eastern Time.

qualpay.**refund**(***kwargs*)
A refund message is used to issue a partial or full refund of a previously captured transaction using the payment gateway identifier. Multiple refunds are allowed per captured transaction provided that the sum of all refunds does not exceed the original captured transaction amount.

Authorizations that have not been captured are not eligible for refund.

qualpay.**credit**(***kwargs*)
A credit message is used to issue a non-referenced credit to a cardholder. A nonreferenced credit requires the cardholder data be provided in the message. The credit message is enabled during the first 30 days of production activity.

After 30 days, the credit message is disabled to prevent fraudulent use of the message. If a credit is necessary after 30 days, it is recommended that the merchant make use of the Qualpay web based business platform to issue the credit. If the merchant requires non-referenced credits to be enabled on the payment gateway beyond 30 days they can request this by contacting Qualpay.

qualpay.**tokenize**(***kwargs*)
A tokenization message is used to securely store cardholder data on the Qualpay system. Once stored, a unique card identifier is returned for use in future transactions. Optionally, tokenization can be requested in an authorization, verification or sale message by sending the tokenize field set to "true".

qualpay.**force**(***kwargs*)
A force message is used to force a declined transaction into the system. This would occur when the online authorization was declined and the merchant received an authorization from a voice or automated response (ARU) system. The required fields are the same as a sale or authorization message with the following exceptions: the cardholder expiration date (exp_date) is not required, and the 6-character authorization code received from the issuer (auth_code) is required.

5.2 qualpay.Card

class qualpay.**Card**(*number*, *exp_month*, *exp_year*, *cvv2*)
A credit card that may be valid or invalid.

brand
Returns the brand of the card, if applicable, else an "unknown" brand.

is_expired
Returns whether or not the card is expired.

is_luhn_valid
Returns whether or not the card's number validates against the luhn algorithm, automatically returning False on an empty value.

is_valid
Returns whether or not the card is a valid card for making payments.

mask
Returns the credit card number with each of the number's digits but the first six and the last four digits replaced by an X, formatted the way they appear on their respective brands' cards.

5.3 qualpay.PaymentGateway

class qualpay.**PaymentGateway**(*merchant_id=None*, *security_key=None*, *base_endpoint=None*)

authorize(***kwargs*)
An authorization message is used to send cardholder data to the issuing bank for approval. An approved transaction will continue to be open until it expires or a capture message is received. Authorizations are automatically voided if they are not captured within 28 days, although most issuing banks will release the hold after 24 hours in retail environments or 7 days in card not present environments.

capture(*pg_id*, ***kwargs*)
A capture message is used to capture a previously authorized transaction using the payment gateway identifier returned by the authorization message. A capture may be completed for any amount up to the authorized amount.

credit(***kwargs*)
A credit message is used to issue a non-referenced credit to a cardholder. A nonreferenced credit requires the cardholder data be provided in the message. The credit message is enabled during the first 30 days of production activity.

After 30 days, the credit message is disabled to prevent fraudulent use of the message. If a credit is necessary after 30 days, it is recommended that the merchant make use of the Qualpay web based business platform to issue the credit. If the merchant requires non-referenced credits to be enabled on the payment gateway beyond 30 days they can request this by contacting Qualpay.

force(***kwargs*)
A force message is used to force a declined transaction into the system. This would occur when the online authorization was declined and the merchant received an authorization from a voice or automated response (ARU) system. The required fields are the same as a sale or authorization message with the following exceptions: the cardholder expiration date (exp_date) is not required, and the 6-character authorization code received from the issuer (auth_code) is required.

refund(*pg_id*, ***kwargs*)
A refund message is used to issue a partial or full refund of a previously captured transaction using the payment gateway identifier. Multiple refunds are allowed per captured transaction provided that the sum of all refunds does not exceed the original captured transaction amount.

Authorizations that have not been captured are not eligible for refund.

sale(***kwargs*)
A sale message is used to perform the function of an authorization and a capture in a single message. This message is used in retail and card not present environments where no physical goods are being shipped.

tokenize(***kwargs*)
A tokenization message is used to securely store cardholder data on the Qualpay system. Once stored, a unique card identifier is returned for use in future transactions. Optionally, tokenization can be requested in an authorization, verification or sale message by sending the tokenize field set to "true".

verify(***kwargs*)
A verify message is used to send cardholder data to the issuing bank for validation. A verify message will return success if the cardholder information was verified by the issuer. If the AVS or CVV2 field is included in the message, then the AVS or CVV2 result code will be returned in the response message.

void(*pg_id*, ***kwargs*)
A void message is used to void a previously authorized transaction. Authorizations can be voided at any time. Captured transactions can be voided until the batch is closed. The batch close time is configurable and by default is 11 PM Eastern Time.

5.4 Exceptions

exception qualpay.**APIError**
Base exception from which all other Qualpay exceptions inherit.

exception qualpay.**GatewayError**(*code*, *response*)
Raised when the payment gateway returns a non-success code.

exception qualpay.**HttpError**(*code*, *response*)
Raised when the payment gateway returns a non-200 response.

```
Payment Gateway Specification v1.2
```

CHAPTER 6

Indices and tables

- genindex
- modindex
- search