
Python Braspag Lib Documentation

Release 0.2

Sérgio Oliveira

November 19, 2012

CONTENTS

Contents:

INSTALLATION

Dependencies:

- Jinja2

Stable version (pypi):

```
$ pip install braspag
```

Development version (github):

```
$ git clone https://github.com/TracyWebTech/braspag.git
$ cd braspag
$ python setup.py install
```


PYTHON BRASPAG LIB

2.1 Request object

`class braspag.BraspagRequest` (*merchant_id=None, homologation=False*)
Implements Braspag Pagador API (manual version 1.9).

Boleto generation is not yet implemented.

`authorize` (*order_id, customer_id, customer_name, customer_email, amount, card_holder, card_number, card_security_code, card_exp_date, save_card, card_token, number_of_payments, payment_method, payment_plan, transaction_type=2, currency='BRL', country='BRA'*)

All arguments supplied to this method must be keyword arguments.

Parameters

- **order_id** – Order id. It will be used to indentify the order later in Braspag.
- **customer_id** – Must be user's CPF/CNPJ.
- **customer_name** – User's full name.
- **customer_email** – User's email address.
- **amount** – Amount to charge.
- **card_holder** – Name printed on card.
- **card_number** – Card number.
- **card_security_code** – Card security code.
- **card_exp_date** – Card expiration date.
- **save_card** – Flag that tell to Braspag to store card number. If set to True Reponse will return a card token. *Default: False.*
- **card_token** – Card token returned by Braspag. When used it should replace *card_holder, card_exp_date, card_number* and *card_security_code*.
- **number_of_payments** – Number of payments that the amount will be devided (number of months). *Default: 1.*
- **currency** – Currency of the given amount. *Default: BRL.*
- **country** – User's country. *Default: BRA.*
- **transaction_type** – An integer representing one of the *Transaction Types*. *Default: 2.*

- **payment_plan** – An integer representing how multiple payments should be handled. *Default: 0*. See *Payment Plans*.
- **payment_method** – Integer representing one of the available *Payment Methods*.

Returns `BraspagResponse`

capture (*transaction_id*, *amount=0*)

Capture the given *amount* from the given *transaction_id*.

This method should only be called after pre-authorizing the transaction by calling `authorize()` with *transaction_types* 1 or 3.

Returns `BraspagResponse`

refund (*transaction_id*, *amount=0*)

Refund the given amount for the given *transaction_id*.

This method should be used to return funds to customers for transactions that happened at least 24 hours ago. For transactions that happened within 24 hours use `void()`.

If the amount is 0 (zero) the full transaction will be refunded.

Returns `BraspagResponse`

void (*transaction_id*, *amount=0*)

Void the given amount for the given *transaction_id*.

This method should be used to return funds to customers for transactions that happened within less than 23h and 59 minutes ago. For other transactions use `refund()`.

If the amount is 0 (zero) the full transaction will be voided.

Returns `BraspagResponse`

2.1.1 Examples

Creditcard example

The example bellow shows how to authorize a credit card transation:

```
from braspag import BraspagRequest

ORDER_ID = u'84765421-5435-85C2-3F41-85A72BE2433E'
MERCHANT_ID = u'12345678-1234-1234-1234-1234567890AB'

request = BraspagRequest(merchant_id=MERCHANT_ID, homologation=True)
response = request.authorize(
    order_id=ORDER_ID,
    customer_id=u'12345678900',
    customer_name=u'José da Silva',
    customer_email='jose123@dasilva.com.br',
    payment_method=997, #simulated BRL
    amount=10000,
    card_holder=u'Jose da Silva',
    card_number=u'0000000000000001',
    card_security_code='123',
    card_exp_date='05/2018',
)
```

The argument *order_id* should be an GUID valid string which indentifies this transaction on your system. The argument *customer_id* must be a valid CPF or CNPJ for the user.

The *amount* given should be an integer where the 2 last digits will represent the decimal places.

2.2 Response object

```
class braspag.BraspagResponse(http_reponse)
```

```
    TODO:
```

```
        correlation_id
```

```
        errors
```

```
        success
```

```
        order_id
```

```
        braspag_order_id
```

```
        transaction_id
```

```
        payment_method
```

```
        amount
```

```
        acquirer_transaction_id
```

```
        authorization_code
```

```
        return_code
```

```
        return_message
```

```
        card_token
```

```
        status
```

2.3 Exceptions

```
exception braspag.exceptions.BraspagException
```

```
    Bases: exceptions.Exception
```

```
    Custom exception
```

```
exception braspag.exceptions.BraspagHttpResponseException(code, msg)
```

```
    Bases: braspag.exceptions.BraspagException
```

```
    Status code Exception
```

2.4 Utils

```
braspag.utils.is_valid_guid(guid)
```

```
braspag.utils.spaceless(xml_str)
```


CONSTANTS

The following tables contains integer codes that will be needed by requests and/or returned by responses.

3.1 Transaction Types

Table 3.1: Transaction Types

Code	Transaction Type
1	Pre-authorization
2	Automatic capture
3	Pre-authorization with authentication
4	Automatic capture with authentication

3.2 Payment Plans

Table 3.2: Payment Plans

Code	Payment Plan
0	One time payment
1	Many payments - interests will be charged from customer
2	Many payments - interests will be charged from establishment

3.3 Payment Methods

Table 3.3: Payment Methods

Code	Payment Method
123	Cielo Visa Electron
500	Cielo Visa
501	Cielo MasterCard
502	Cielo Amex
503	Cielo Diners
504	Cielo Elo
505	Banorte Visa
Continued on next page	

Table 3.3 – continued from previous page

Code	Payment Method
506	Banorte MasterCard
507	Banorte Diners
508	Banorte Amex
509	Redecard Visa
510	Redecard MasterCard
511	Redecard Diners
512	PagosOnLine Visa
513	PagosOnLine MasterCard
514	PagosOnLine Amex
515	PagosOnLine Diners
516	Payvision Visa
517	Payvision MasterCard
518	Payvision Diners
519	Payvision Amex
520	Banorte Cargos Automaticos Visa
521	Banorte Cargos Automaticos MasterCard
522	Banorte Cargos Automaticos Diners
523	Amex 2P
524	SITEF Visa
525	SITEF MasterCard
526	SITEF Amex
527	SITEF Diners
528	SITEF HiperCard
529	SITEF Leader
530	SITEF Aura
531	SITEF Santander Visa
532	SITEF Santander MasterCard
995	Simulated USD
996	Simulated EUR
997	Simulated BRL

GET THE SOURCE

Source: <https://github.com/TracyWebTech/braspag>

Travis:

INDICES AND TABLES

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

PYTHON MODULE INDEX

b

`braspag.exceptions, ??`

`braspag.utils, ??`