
daraja-mpesa Documentation

Release 2.0 stable

Tralah M Brian

Jul 17, 2021

Contents

1 Overview	1
1.1 Why this Library ?	1
2 Getting Started	3
2.1 Installation	3
3 How-to Guides	5
4 api	7
4.1 api package	7
5 Indices and tables	15
Python Module Index	17
Index	19

1.1 Why this Library ?

This project was motivated by the need and frustration faced by my colleagues and myself when developing software that interfaced and intergrated with [Mpesa's Daraja API](https://developer.safaricom.co.ke) (https://developer.safaricom.co.ke) , A popular mobile money platform that is widely used in Africa.

A lot of developers, and that may include you, as you are reading this, face some trouble when interfacing with the daraja api especially the messy parts of authenticating and obtaining an Oauth Token from the api that expires quickly even when you keep the token for 30 days.

For python developers,even when using the awesome *requests* library to do the work, you will often need to manually prepare the request headers, the payload which varies depending on the api you are interfacing with and still parse the request response to extract the information you need.

That is why i developed this library, to abstract away those messy, repetitive tasks you will often do in your code as you intergrate Mpesa into your software solution. This is to help you focus on the business logic implementation and not on the unimportant third party payment gateway.

This will result in cleaner, shorter, and beautiful code that is easy to refactor and maintain.

Now head over to the [getting started section](#) and happy coding!!

2.1 Installation

The library can be installed using *pip* or manually from the source code.

To install via *pip* open terminal and assuming you have *pip* installed run

```
$ pip install daraja-mpesa
```

To build from source, clone the github repository and run in terminal

```
$ git clone https://github.com/TralahM/pympesa.git.  
$ cd pympesa.  
$ python setup.py install
```

This avails an **mpesa** module for import in your python code.

Now head over to the [How-to section](#) for how to use the library.

CHAPTER 3

How-to Guides

4.1 api package

4.1.1 Submodules

4.1.2 api.auth module

```
class api.auth.MpesaBase (env='sandbox', app_key=None, app_secret=None,  
                          sandbox_url='https://sandbox.safaricom.co.ke',  
                          live_url='https://api.safaricom.co.ke')
```

Bases: object

authenticate ()

To make Mpesa API calls, you will need to authenticate your app. This method is used to fetch the access token required by Mpesa. Mpesa supports client_credentials grant type. To authorize your API calls to Mpesa, you will need a Basic Auth over HTTPS authorization token. The Basic Auth string is a base64 encoded string of your app's client key and client secret.

Args:

- *env* (str): Current app environment. Options: sandbox, live.
- *app_key* (str): The app key obtained from the developer portal.
- *app_secret* (str): The app key obtained from the developer portal.
- *sandbox_url* (str): Base Safaricom sandbox url.
- *live_url* (str): Base Safaricom live url.

Returns:

- *access_token* (str): This token is to be used with the Bearer header for further API calls to Mpesa.

4.1.3 api.b2b module

```
class api.b2b.B2B (env='sandbox', app_key=None, app_secret=None, sandbox_url='https://sandbox.safaricom.co.ke', live_url='https://api.safaricom.co.ke')
Bases: api.auth.MpesaBase
```

```
transact (initiator=None, security_credential=None, command_id=None, sender_identifier_type=None, receiver_identifier_type=None, amount=None, party_a=None, party_b=None, remarks=None, account_reference=None, queue_timeout_url=None, result_url=None)
```

This method uses Mpesa's B2B API to transact from one company to another.

Args:

- *initiator* (str): Username used to authenticate the transaction.
- *security_credential* (str): Generate from developer portal
- *command_id* (str): Options: BusinessPayBill, BusinessBuyGoods, DisburseFundsToBusiness, BusinessToBusinessTransfer, BusinessTransferFromMMFToUtility, BusinessTransferFromUtilityToMMF, MerchantToMerchantTransfer, MerchantTransferFromMerchantToWorking, MerchantServicesMMFAccountTransfer, AgencyFloatAdvance
- *sender_identifier_type* (str): 2 for Till Number, 4 for organization shortcode
- *receiver_identifier_type* (str): # 2 for Till Number, 4 for organization shortcode
- *amount* (str): Amount.
- *party_a* (int): Sender shortcode.
- *party_b* (int): Receiver shortcode.
- *remarks* (str): Comments that are sent along with the transaction(maximum 100 characters).
- *account_reference* (str): Use if doing paybill to banks etc.
- *queue_timeout_url* (str): The url that handles information of timed out transactions.
- *result_url* (str): The url that receives results from M-Pesa api call.

Returns:

- *OriginatorConversionID* (str): The unique request ID for tracking a transaction.
- *ConversationID* (str): The unique request ID returned by mpesa for each request made
- *ResponseDescription* (str): Response Description message

4.1.4 api.b2c module

```
class api.b2c.B2C (env='sandbox', app_key=None, app_secret=None, sandbox_url='https://sandbox.safaricom.co.ke', live_url='https://api.safaricom.co.ke')
Bases: api.auth.MpesaBase
```

```
transact (initiator_name=None, security_credential=None, command_id=None, amount=None, party_a=None, party_b=None, remarks=None, queue_timeout_url=None, result_url=None, occasion=None)
```

This method uses Mpesa's B2C API to transact between an M-Pesa short code to a phone number registered on M-Pesa..

Args:

- *initiator_name* (str): Username used to authenticate the transaction.

- *security_credential* (str): Generate from developer portal
- *command_id* (str): Options: SalaryPayment, BusinessPayment, PromotionPayment
- *amount* (str): Amount.
- *party_a* (int): Organization/MSISDN making the transaction - Shortcode (6 digits) - MSISDN (12 digits).
- *party_b* (int): MSISDN receiving the transaction (12 digits).
- *remarks* (str): Comments that are sent along with the transaction(maximum 100 characters).
- *account_reference* (str): Use if doing paybill to banks etc.
- *queue_timeout_url* (str): The url that handles information of timed out transactions.
- *result_url* (str): The url that receives results from M-Pesa api call.
- *ocassion* (str):

Returns:

- *OriginatorConversionID* (str): The unique request ID for tracking a transaction.
- *ConversationID* (str): The unique request ID returned by mpesa for each request made
- *ResponseDescription* (str): Response Description message

4.1.5 api.balance module

```
class api.balance.Balance (env='sandbox', app_key=None, app_secret=None,
                             sandbox_url='https://sandbox.safaricom.co.ke',
                             live_url='https://api.safaricom.co.ke')
```

Bases: *api.auth.MpesaBase*

```
get_balance (initiator=None, security_credential=None, command_id=None, party_a=None, identifier_type=None, remarks=None, queue_timeout_url=None, result_url=None)
```

This method uses Mpesa's Account Balance API to to enquire the balance on an M-Pesa BuyGoods (Till Number).

Args:

- *initiator* (str): Username used to authenticate the transaction.
- *security_credential* (str): Generate from developer portal.
- *command_id* (str): AccountBalance.
- *party_a* (int): Till number being queried.
- *identifier_type* (int): Type of organization receiving the transaction. Options: 1 - MSISDN 2 - Till Number 4 - Organization short code
- *remarks* (str): Comments that are sent along with the transaction(maximum 100 characters).
- *queue_timeout_url* (str): The url that handles information of timed out transactions.
- *result_url* (str): The url that receives results from M-Pesa api call.

Returns:

- *OriginatorConversionID* (str): The unique request ID for tracking a transaction.
- *ConversationID* (str): The unique request ID returned by mpesa for each request made
- *ResponseDescription* (str): Response Description message

4.1.6 api.c2b module

```
class api.c2b.C2B (env='sandbox', app_key=None, app_secret=None, sandbox_url='https://sandbox.safaricom.co.ke', live_url='https://api.safaricom.co.ke')
```

Bases: `api.auth.MpesaBase`

```
register (shortcode=None, response_type=None, confirmation_url=None, validation_url=None)
```

This method uses Mpesa's C2B API to register validation and confirmation URLs on M-Pesa.

Args:

- *shortcode* (int): The short code of the organization.
- *response_type* (str): Default response type for timeout. In case a transaction times out, Mpesa will by default Complete or Cancel the transaction.
- *confirmation_url* (str): Confirmation URL for the client.
- *validation_url* (str): Validation URL for the client.

Returns:

- *OriginatorConversationID* (str): The unique request ID for tracking a transaction.
- *ConversationID* (str): The unique request ID returned by mpesa for each request made
- *ResponseDescription* (str): Response Description message

```
simulate (shortcode=None, command_id=None, amount=None, msisdn=None, bill_ref_number=None)
```

This method uses Mpesa's C2B API to simulate a C2B transaction.

Args:

- *shortcode* (int): The short code of the organization.
- *command_id* (str): Unique command for each transaction type. - CustomerPayBillOnline - Customer-BuyGoodsOnline.
- *amount* (int): The amount being transacted
- *msisdn* (int): Phone number (msisdn) initiating the transaction MSISDN(12 digits)
- *bill_ref_number*: Optional

Returns:

- *OriginatorConverstionID* (str): The unique request ID for tracking a transaction.
- *ConversationID* (str): The unique request ID returned by mpesa for each request made
- *ResponseDescription* (str): Response Description message

4.1.7 api.mpesa_express module

```
class api.mpesa_express.MpesaExpress (env='sandbox', app_key=None, app_secret=None, sandbox_url='https://sandbox.safaricom.co.ke', live_url='https://api.safaricom.co.ke')
```

Bases: `api.auth.MpesaBase`

```
query (business_shortcode=None, checkout_request_id=None, passcode=None)
```

This method uses Mpesa's Express API to check the status of a Lipa Na M-Pesa Online Payment..

Args:

- *business_shortcode* (int): This is organizations shortcode (Paybill or Buygoods - A 5 to 6 digit account number) used to identify an organization and receive the transaction.
- *checkout_request_id* (str): This is a global unique identifier of the processed checkout transaction request.
- *passcode* (str): Get from developer portal

Returns:

- *CustomerMessage* (str):
- *CheckoutRequestID* (str):
- *ResponseDescription* (str):
- *MerchantRequestID* (str):
- *ResponseCode* (str):

stk_push (*business_shortcode=None, passcode=None, amount=None, callback_url=None, reference_code=None, phone_number=None, description=None*)

This method uses Mpesa's Express API to initiate online payment on behalf of a customer..

Args:

- *business_shortcode* (int): The short code of the organization.
- *passcode* (str): Get from developer portal
- *amount* (int): The amount being transacted
- *callback_url* (str): A CallBack URL is a valid secure URL that is used to receive notifications from M-Pesa API.
- *reference_code*: Account Reference: This is an Alpha-Numeric parameter that is defined by your system as an Identifier of the transaction for CustomerPayBillOnline transaction type.
- *phone_number*: The Mobile Number to receive the STK Pin Prompt.
- *description*: This is any additional information/comment that can be sent along with the request from your system. MAX 13 characters

Returns:

- *CustomerMessage* (str):
- *CheckoutRequestID* (str):
- *ResponseDescription* (str):
- *MerchantRequestID* (str):
- *ResponseCode* (str):

4.1.8 api.reversal module

```
class api.reversal.Reversal (env='sandbox', app_key=None, app_secret=None,
                             sandbox_url='https://sandbox.safaricom.co.ke',
                             live_url='https://api.safaricom.co.ke')
```

Bases: *api.auth.MpesaBase*

```
reverse (initiator=None, security_credential=None, command_id='TransactionReversal', transaction_id=None, amount=None, receiver_party=None, receiver_identifier_type=None, queue_timeout_url=None, result_url=None, remarks=None, occasion=None)
```

This method uses Mpesa's Transaction Reversal API to reverse a M-Pesa transaction.

Args:

- *initiator* (str): Username used to authenticate the transaction.
- *security_credential* (str): Generate from developer portal
- *command_id* (str): TransactionReversal
- *transaction_id* (str): Unique identifier to identify a transaction on M-Pesa.
- *amount* (int): The amount being transacted
- *receiver_party* (int): Organization/MSISDN making the transaction - Shortcode (6 digits) - MSISDN (12 digits).
- *receiver_identifier_type* (int): MSISDN receiving the transaction (12 digits).
- *queue_timeout_url* (str): The url that handles information of timed out transactions.
- *result_url* (str): The url that receives results from M-Pesa api call.
- *remarks* (str): Comments that are sent along with the transaction(maximum 100 characters)
- *occassion* (str):

Returns:

- *OriginatorConversionID* (str): The unique request ID for tracking a transaction.
- *ConversationID* (str): The unique request ID returned by mpesa for each request made
- *ResponseDescription* (str): Response Description message

4.1.9 api.transaction_status module

```
class api.transaction_status.TransactionStatus (env='sandbox', app_key=None,
app_secret=None, sandbox_url='https://sandbox.safaricom.co.ke',
live_url='https://api.safaricom.co.ke')
```

Bases: *api.auth.MpesaBase*

```
check_transaction_status (party_a=None, identifier_type=None, remarks=None,
initiator=None, passcode=None, result_url=None,
queue_timeout_url=None, transaction_id=None, occassion=None,
shortcode=None)
```

This method uses Mpesa's Transaction Status API to check the status of a transaction.

Args:

- *party_a* (str): Organization/MSISDN receiving the transaction - MSISDN or shortcode.
- *identifier_type* (str): Type of organization receiving the transaction 1-MSISDN. 2-Till Number, 3-Shortcode.
- *remarks* (str): Comments that are sent along with the transaction(maximum 100 characters).
- *initiator* (str): This is the credential/username used to authenticate the transaction request.
- *passcode* (str): Get from developer portal
- *result_url* (str): The url that handles information from the mpesa API call.
- *transaction_id* (str): Unique identifier to identify a transaction on M-Pesa.
- *queue_timeout_url* (str): The url that stores information of timed out transactions.

- *result_url* (str): The url that receives results from M-Pesa api call.
- *shortcode* (int): The short code of the organization.
- *occassion* (str):

Returns:

- *ResultDesc*: ,
- *CheckoutRequestID*: ,
- *ResponseDescription*: ,
- *MerchantRequestID*: ,
- *ResponseCode*: ,
- *ResultCode*:

4.1.10 Module contents

CHAPTER 5

Indices and tables

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

a

api, 13
api.auth, 7
api.b2b, 8
api.b2c, 8
api.balance, 9
api.c2b, 10
api.mpesa_express, 10
api.reversal, 11
api.transaction_status, 12

A

`api` (module), 13
`api.auth` (module), 7
`api.b2b` (module), 8
`api.b2c` (module), 8
`api.balance` (module), 9
`api.c2b` (module), 10
`api.mpesa_express` (module), 10
`api.reversal` (module), 11
`api.transaction_status` (module), 12
`authenticate()` (*api.auth.MpesaBase* method), 7

B

`B2B` (class in *api.b2b*), 8
`B2C` (class in *api.b2c*), 8
`Balance` (class in *api.balance*), 9

C

`C2B` (class in *api.c2b*), 10
`check_transaction_status()`
(*api.transaction_status.TransactionStatus*
method), 12

G

`get_balance()` (*api.balance.Balance* method), 9

M

`MpesaBase` (class in *api.auth*), 7
`MpesaExpress` (class in *api.mpesa_express*), 10

Q

`query()` (*api.mpesa_express.MpesaExpress* method),
10

R

`register()` (*api.c2b.C2B* method), 10
`Reversal` (class in *api.reversal*), 11
`reverse()` (*api.reversal.Reversal* method), 11

S

`simulate()` (*api.c2b.C2B* method), 10
`stk_push()` (*api.mpesa_express.MpesaExpress*
method), 11

T

`transact()` (*api.b2b.B2B* method), 8
`transact()` (*api.b2c.B2C* method), 8
`TransactionStatus` (class in
api.transaction_status), 12