# django_payu Documentation

*Release 0.3*

**Micro Pyramid**

November 10, 2016

# Contents

This package provides integration between Django and PayU Payment Gateway.

CHAPTER 1

# Installation

Install:

```
pip install django-payu
```

Next, add `payu` to your `INSTALLED_APPS` setting like this:

```
INSTALLED_APPS = [
    ....
    ....
    'payu',
]
```

Then, add the following settings in the setting file using the details from your PayU account:

```
PAYU_MERCHANT_KEY = "Your MerchantID",

PAYU_MERCHANT_SALT = "Your MerchantSALT",

# Change the PAYU_MODE to 'LIVE' for production.
PAYU_MODE = "TEST"
```

Finally, run the following commands:

```
python manage.py migrate
```

Contents:

## 1.1 Configuration Settings

In order to test your integration, first create a **PayU** Account. Once this has been created, you will be given the following...

- *Key (MerchantID)* – This ID is generated at the time of activation of your site and helps to uniquely identify you to PayU.
- *SALT* – This will be provided by PayU.in.

Add the following settings to the PAYU_INFO dictionary using the details from your PayU account:

```
PAYU_MERCHANT_KEY = "Your MerchantID",

PAYU_MERCHANT_SALT = "Your MerchantSALT",
# And add the PAYU_MODE to 'TEST' for testing and 'LIVE' for production.
PAYU_MODE = "TEST"
```

## 1.2 API Reference

*Django PayU* is an API for integrating PayU Payment Gateway and your website in respect of powering online transactions.

Django PayU supports a wide variety of transactions through the API. This page contains specific information on the API functions.

### 1.2.1 1. PayU URL (payu_url)

This function returns the *PayU* respective url based on the *mode* from settings file.

**Example**

```
from payu.gateway import payu_url

payu_url = payu_url()
```

### 1.2.2 2. Hash (Checksum) - (get_hash)

This function returns the hash value computed by the checksum formula as specified by the PayU Payment Gateway.

A checksum is generated by a mathematical function using the Merchant ID(key), message and the Salt as input. The Checksum algorithm used is SHA512.

*Hash or Checksum Formula before transaction -*

```
from hashlib import sha512

sha512(key|txnid|amount|productinfo|firstname|email|udf1|udf2|udf3|udf4|udf5||||||<SALT>)
```

**Parameters**

- *key* – Merchant ID provided by PayU.
- *txnid* – Transaction number that merchant uses to track order. (Must be unique every time if already successful, otherwise you get an error of duplicate transaction).
- *amount* – Payment amount (Type cast the amount to float).
- *productinfo* – Product Description.
- udf1 to udf10 – User defined fields.
- *SALT* – SALT will be provided by PayU.in

**Example**

```
from payu.gateway import get_hash
from uuid import uuid4

data = {
    'txnid':uuid4().hex, 'amount':10.00, 'productinfo': 'Sample Product',
    'firstname': 'test', 'email': 'test@example.com', 'udf1': 'Userdefined field',
}
hash_value = get_hash(data)
```

### 1.2.3 3. Confirm transaction (check_hash)

This function is used to compute the hash (or checksum) after the transaction. It is strongly recommended that the hash (or checksum) is computed again after the transaction and is compared with what we post in the return parameters.

*Hash or Checksum Formula after transaction* -

```
from hashlib import sha512

sha512(<SALT>|status||||||udf5|udf4|udf3|udf2|udf1|email|firstname|productinfo|amount|txnid|key)
```

**Parameters**

After the transaction has been completed, the PayU returns the data to our success url along with data sent by us in a POST method. The whole data can be passed to the check_hash() function to verify that data is sent to and from the PayU.

Same parameters as get_hash() function along with the following additional parameter -

- *status* – SUCCESS/PENDING/FAILURE sent from PayU.in in the POST request.

**Return Value**

The check_hash() function returns either True or False.

**Example**

```
from django.http import HttpResponse
from payu.gateway import check_hash
from uuid import uuid4

def success_response(request):
    hash_value = check_hash(request.POST)
    if check_hash(request.POST):
        return HttpResponse("Transaction has been Successful.")
```

### 1.2.4 4. Verify transaction (verify_payment)

This function is used to reconcile the transaction with PayU. It returns MIHPayID ( Unique ID generated for a transaction by PayU.in), amount, discount, mode and status of transaction.

**Parameters**

- *txnid* – Transaction ID generated by you to uniquely identify a transaction.

**Example**

```
from payu.gateway import verify_payment

response = verify_payment("Your txnid")
print response

# Example Successful Response
{"status":1,"msg":"Transaction Fetched Successfully", "transaction_details": {"mihpayid": "MIHPayID",
```

### 1.2.5 5. Check transaction (check_payment)

This function is used to check payment after transaction. It returns all the parameters for a given transaction.

**Parameters**

- *mihpayid* – Payu id (mihpayid) of transaction (Unique ID generated for a transaction by PayU.in).

**Example**

```
from payu.gateway import check_payment

response = check_payment("Your mihpayid")
print response

# Example Successful Response
{"status":1,"msg":"Transaction Fetched Successfully","transaction_details": {"request_id":"RequestID"
```

### 1.2.6 6. Capture transaction (capture_transaction)

This function is used to capture an auth transaction . The return parameters are: request_id, bank_ref_num.

**Parameters**

- *mihpayid* – Payu id (mihpayid) of transaction (Unique ID generated for a transaction by PayU.in).

**Example**

```
from payu.gateway import capture_transaction

response = capture_transaction("Your mihpayid")
print response

# Example Failure Response
{"status": 0, "msg":" Capture failed", "error_code": 111}

# Example Successful Response
{"status": 1, "msg": "Capture Request Queued", "request_id": "RequestID", "bank_ref_num": "Bank Refe
```

### 1.2.7 7. Refund transaction (refund_transaction)

This function is used to refund a captured transaction. The return parameters are status, request_id,bank_ref_num. Refund can occur only after one day of capture.

**Parameters**

- *mihpayid* – Payu id (mihpayid) of transaction (Unique ID generated for a transaction by PayU.in).
- *amount* – Amount to be refunded.

**Example**

```
from payu.gateway import refund_transaction

response = refund_transaction("Your mihpayid", "Amount")
print response

# Example Successful Responses
# if capture is done on the same day
{"status": 1, "msg": "Capture is done today, please check for refund status tomorrow", "request_id":

# if request is queued for further processing.
{"status": 1, "msg": "Refund Request Queued", "request_id": "RequestID", "mihpayid": "MIHPayID", "ban

# Example Failure Response
{"status": 0, "msg":"Refund request failed"}
```

### 1.2.8 8. Cancel transaction (cancel_transaction)

This function is used to cancel an auth transaction. The return parameters are status, txn_update_id, bank_ref_num.

**Parameters**

- *mihpayid* – Payu id (mihpayid) of transaction (Unique ID generated for a transaction by PayU.in).
- *amount* – Amount.

**Example**

```
from payu.gateway import cancel_transaction

response = cancel_transaction("Your mihpayid", "Amount")
print response

# Example Successful Responses
{"status": 1, "msg": "Cancel Request Queued", "request_id": "RequestID", "mihpayid": "MIHPayID", "ban

# Example Failure Response
{"status": 0, "msg":"Cancel request failed"}
```

### 1.2.9 9. Cancel or Refund transaction (cancel_refund_transaction)

This function is used to cancel an auth transaction and refund a captured transaction. The return parameters are status, msg.

**Parameters**

- *mihpayid* – Payu id (mihpayid) of transaction (Unique ID generated for a transaction by PayU.in).
- *amount* – Amount which needs to be refunded. Please note that both partial and full refunds are allowed.

**Example**

```
from payu.gateway import cancel_refund_transaction

response = cancel_refund_transaction("Your mihpayid", "Amount")
print response

# Example Successful Responses

# On successful processing at our end
{"status": 1, "msg": "Cancel Request Queued", "request_id": "RequestID", "mihpayid": "MIHPayID", "ban

# On successful processing on our end for captured transactions
{'status': 1, 'bank_ref_num': "Bank Reference Number", 'mihpayid': MIHPayID, 'request_id': 'RequestI
```

### 1.2.10 10. Check action status (check_action_status)

This function is used to check status of refund/cancel requests. The return parameter are MIHPayID, Amount, Discount, Mode and Status of transaction.

**Parameters**

- *request_id* – Payu id (Cancel/Refund Request ID).

**Example**

```
from payu.gateway import check_action_status

response = check_action_status("Request id")
print response

# Example Successful Responses
{"status": 1, "msg": "Transaction Fetched Successfully", "transaction_details": {"status": "Request S
```

CHAPTER 2

# Indices and tables

- genindex
- modindex
- search