

Sample App Guide for Testing Python SDK (Sample App Framework: Flask)

Within this document, the step-by-step process elucidates running the Sample App to evaluate the functionality of the Python SDK

By Pinelabs Team

Prerequisites

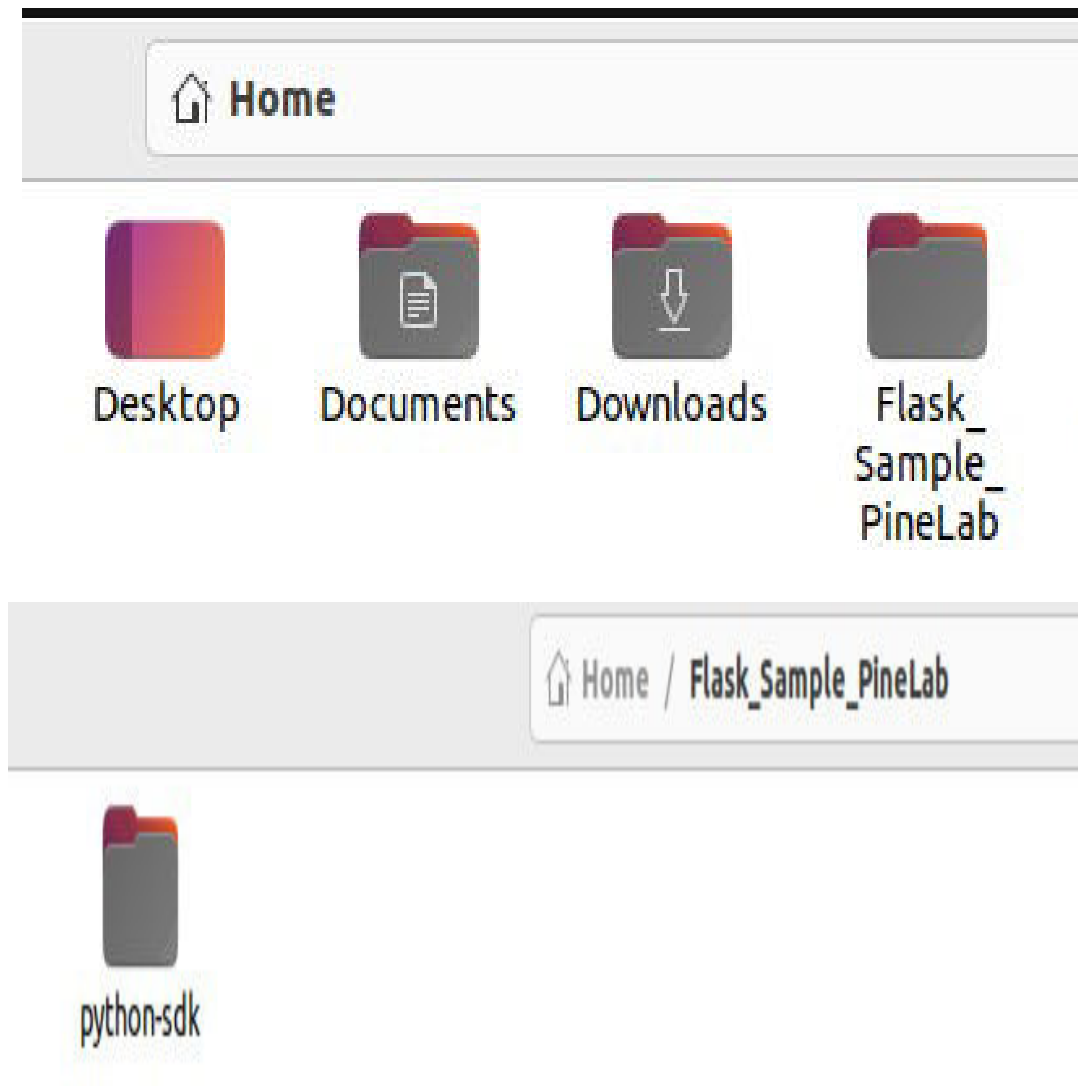
Python Version ≥ 3 (Recommended)

Installation Process

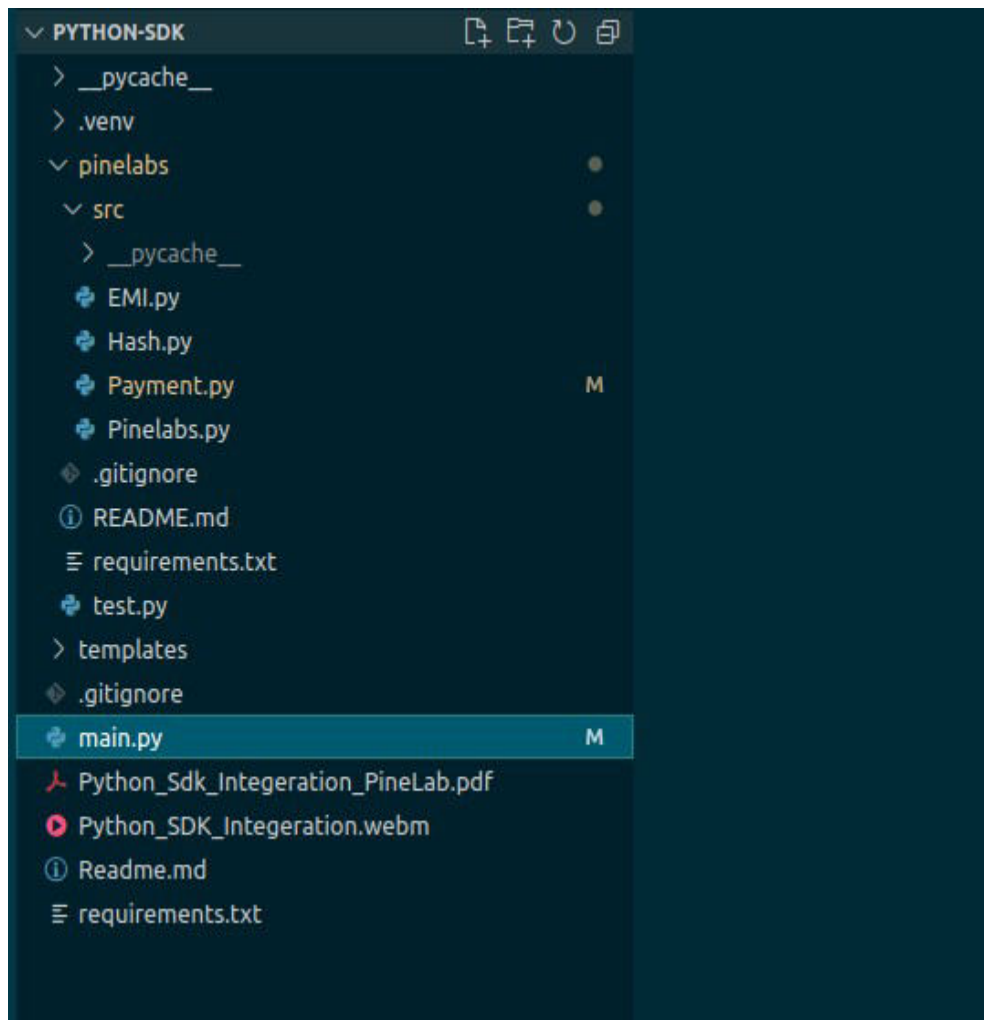
This section explains the installation flow of the Sample App

Below are the steps to extract the Sample App onto your local System and Enabling it for browser.

1. Extract the Sample App (Zip File) in any location on your system. Here we have extracted it at home path inside Flask_Sample_PineLab Folder (As shown below in image)



2. So it'll look something like this in your IDE, it'll consist of Sdk in pine_lab/api path and Controller classes will be located inside pine_labs /pine_lab path.



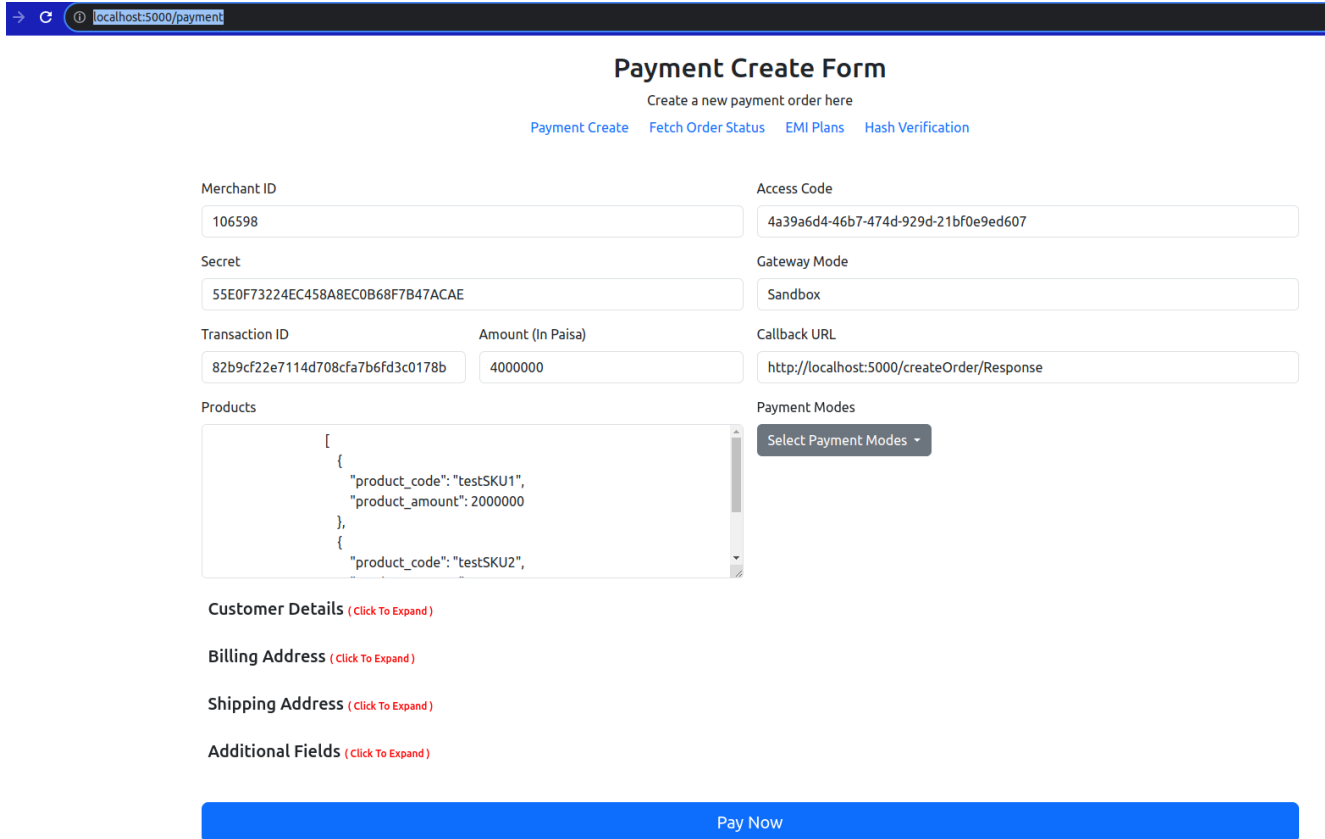
3. Now you need to run `python -m venv .venv` in your terminal in this path itself to setup virtual environment.
4. Run `./venv/Scripts/activate.bat` to activate the virtual env.
5. Run `pip install -r requirements.txt` to install all the dependencies
6. Then run `flask --app main.py run --debug` to start the development server for the application
7. Open the provided url in the console in your browser and navigate to `/payment` to start using the SDK in the sample app to create, fetch, emi, hash verify using it.
(Note- The commands may vary in other OS like Linux)

Usage Process

This section explains the usage flow of Sample App

Create Order Process

In order for you to create an order and generate a URL for payment you need to locate to url <http://localhost:5000/payment> (Below shows the image of how Create order page will look like on browser)



The screenshot shows a web browser at the URL localhost:5000/payment. The page is titled "Payment Create Form" with a subtitle "Create a new payment order here". Below the title are four links: "Payment Create", "Fetch Order Status", "EMI Plans", and "Hash Verification". The form contains several input fields and sections:

- Merchant ID:** 106598
- Access Code:** 4a39a6d4-46b7-474d-929d-21bf0e9ed607
- Secret:** 55E0F73224EC458A8EC0B68F7B47ACAE
- Gateway Mode:** Sandbox
- Transaction ID:** 82b9cf22e7114d708cfa7b6fd3c0178b
- Amount (In Paisa):** 4000000
- Callback URL:** <http://localhost:5000/createOrder/Response>
- Products:** A JSON array containing two product objects:

```
[
  {
    "product_code": "testSKU1",
    "product_amount": 2000000
  },
  {
    "product_code": "testSKU2",
    "product_amount": 2000000
  }
]
```
- Payment Modes:** A dropdown menu labeled "Select Payment Modes".
- Customer Details:** A section with a "(Click To Expand)" link.
- Billing Address:** A section with a "(Click To Expand)" link.
- Shipping Address:** A section with a "(Click To Expand)" link.
- Additional Fields:** A section with a "(Click To Expand)" link.
- Pay Now:** A large blue button at the bottom of the form.

STEPS TO FOLLOW

1. Merchant Id, Access Code, Secret Key, Gateway Mode (Testing/Prod), Transaction Id, Amount in Paisa(Total amount of Products), Callback Url & Payment mode are all required fields and some of them have been pre filled with Test keys which you can change according to your requirements.
2. Product Details is Conditional (Meaning, Required in case of EMI payment mode)
3. Customer Details, Billing Address, Shipping Address and Additional Fields are Optional fields so aren't necessary for this process.
4. You need to keep Transaction Id unique every time while creating order else due to duplicate Id error you won't be redirected to anywhere.
5. If you add another product or subtract from existing products in Product Details then you need to make sure sum of total products amount is to be equal to Amount in Paisa as well if not change amount in Paisa according to total of Products in Product list, for test we have added 4000000 in amount in Paisa which equals to total amount of products in Product details you can change according to your needs.

6. After clicking on Pay Now you'll be redirected to Pinelabs Payment link page and you can select any payment mode like for testing Netbanking can be selected and after clicking on pay now you can generate success or Failure response.

The screenshot displays the Plural by Pine Labs payment interface. At the top right, there is a 'Cancel' button with a close icon. The order total is shown as ₹40000. The interface is divided into two main sections: 'Choose a payment option' on the left and 'Select Bank' on the right. Under 'Choose a payment option', there are four buttons: 'Credit Debit Cards', 'Net Banking' (which is highlighted in green), 'UPI/BHIM', and 'Wallets'. Under 'Select Bank', there is a button for 'IDBI Bank' and a 'Pay Now' button. A search bar labeled 'Select Bank' is also present. At the bottom, there are logos for 'Safe & Secure Payments', 'PCI DSS', 'Verified by VISA', 'MasterCard SecureCode', and 'SafeKey'. The footer contains the text 'Privacy Policy | Terms of Use | Copyright © 2022-2023 Pine Labs, All rights reserved.'

7. After that you'll be redirected to url provided in Callback Url field in HTML form which is part of this sample App and you'll get response something like shown in below image.

```
localhost:5000/createOrder/Response

{
  "Acquirer_Response_Code": "0300",
  "Acquirer_Response_Message": "DEFAULT",
  "acquirer_name": "BILLDESK",
  "amount_in_paisa": "4000000",
  "captured_amount_in_paisa": "4000000",
  "dia_secret": "8A3F370694CF57F7335E339F6A0C71192BDD14A54ADCB9231D73947D3DE55144",
  "dia_secret_type": "SHA256",
  "merchant_access_code": "4a39a6d4-46b7-474d-929d-21bf0e9ed607",
  "merchant_id": "106598",
  "mobile_no": "",
  "parent_txn_response_code": "",
  "parent_txn_response_message": "",
  "parent_txn_status": "",
  "payment_mode": "3",
  "pine_pg_transaction_id": "14225964",
  "pine_pg_txn_status": "4",
  "refund_amount_in_paisa": "0",
  "txn_completion_date_time": "30/11/2023 02:29:47 PM",
  "txn_response_code": "1",
  "txn_response_msg": "SUCCESS",
  "udf_field_1": "",
  "udf_field_2": "",
  "udf_field_3": "",
  "udf_field_4": "",
  "unique_merchant_txn_id": "70947dc5c6c3448b9e4e729d1d629a69"
}
```

Fetch Order Process

In order to fetch order details you need to click on link Fetch Order Status in create order page or you can navigate to it directly by using url (<http://localhost:5000/payment/fetch>)

localhost:5000/payment/fetch

Fetch Transaction Form

Check Fetch and EMI API Response

[Payment Create](#) [Fetch Order Status](#) [EMI Plans](#) [Hash Verification](#)

Merchant ID	Access Code
106598	4a39a6d4-46b7-474d-929d-21bf0e9ed607
Secret	Gateway Mode
55E0F73224EC458A8EC0B68F7B47ACAE	Sandbox
Transaction ID	
1ab05090478948ddad9f9ba456091de1	

Fetch Payment Status

STEPS TO FOLLOW

1. Merchant Id, Access Code, Secret Key, Gateway Mode (Testing/Prod), Transaction Id are required fields and some of them have been pre filled with Test keys that you can change according to your requirement.
2. You can change transaction id as well according to you.
3. Now you need to click on fetch order and it'll give you response something like shown in below image

Merchant ID

106598

Access Code

4a39a6d4-46b7-474d-929d-21bf0e9ed607

Secret

55E0F73224EC458A8EC0B68F7B47ACAE

Gateway Mode

Sandbox

Transaction ID

1ab05090478948ddad9f9ba456091de1

Fetch Payment Status

```
{
  "ppc_MerchantID": "106598",
  "ppc_MerchantAccessCode": "4a39a6d4-46b7-474d-929d-21bf0e9ed607",
  "ppc_PinePGTxnStatus": "7",
  "ppc_TransactionCompletionDateTime": "24/11/2023 11:48:19 AM",
  "ppc_UniqueMerchantTxnID": "1ab05090478948ddad9f9ba456091de1",
  "ppc_Amount": "4000000",
  "ppc_TxnResponseCode": "1",
  "ppc_TxnResponseMessage": "SUCCESS",
  "ppc_PinePGTransactionID": "14215875",
  "ppc_CapturedAmount": "3880908",
  "ppc_RefundedAmount": "0",
  "ppc_AcquirerName": "HDFC",
  "ppc_DIA_SECRET": "CFD90BE4DC85E4688E73E61185F44389B5B917F46C1308885A9016F7ED23A94F",
  "ppc_DIA_SECRET_TYPE": "SHA256",
  "ppc_EMITenureMonth": "6",
  "ppc_EMIIInterestRatePercent": "10.00",
  "ppc_EMIPrincipalAmount": "3880908",
  "ppc_EMIAmountPayableEachMonth": "653371",
  "ppc_Is_BankEMITransaction": "1",
  "ppc_EMICashBackType": "0",
  "ppc_PaymentMode": "4",
  "ppc_Parent_TxnStatus": "4",
  "ppc_ParentTxnResponseCode": "1",
}
```

EMI Details Process

In order to calculate EMI we need to click on EMI Plans or we can navigate to the path (<http://localhost:5000/payment/emi>)

EMI Calculation Form

Check Fetch and EMI API Response

[Payment Create](#) [Fetch Order Status](#) [EMI Plans](#) [Hash Verification](#)

Merchant ID	Access Code
<input type="text" value="106598"/>	<input type="text" value="4a39a6d4-46b7-474d-929d-21bf0e9ed607"/>
Secret	Gateway Mode
<input type="text" value="55E0F73224EC458A8EC0B68F7B47ACAE"/>	<input type="text" value="Sandbox"/>
Products	Product Amount
<pre>[{ "product_code": "testSKU1", "product_amount": 2000000 }, { "product_code": "testSKU2", "product_amount": 2000000 }]</pre>	<input type="text" value="4000000"/>

Fetch EMI Options

STEPS TO FOLLOW

1. Merchant Id, Access Code, Secret Key, Gateway Mode (Testing/Prod), Amount in Paisa, Product Details are required fields and they have been pre filled with Test keys you can change according to your requirement.
2. If you add another product or subtract from existing products in Product Details then you need to make sure sum of total products amount is to be equal to Amount in Paisa as well if not change amount in Paisa according to total of Products in Product list, for test we have added 4000000 in amount in Paisa which equals to total amount of products in Product details you can change according to your needs.
3. After clicking on Calculate Fetch EMI Options button you'll be able to see response as shown in below image.

Products

```
[
  {
    "product_code": "testSKU1",
    "product_amount": 2000000
  },
  {
    "product_code": "testSKU2",
    "product_amount": 2000000
  }
]
```

Product Amount

Fetch EMI Options

```
{
  "issuer": [
    {
      "list_emi_tenure": [
        {
          "offer_scheme": {
            "product_details": [
              {
                "schemes": [
                  {
                    "scheme_id": 48150,
                    "program_type": 112,
                    "is_scheme_valid": true
                  }
                ]
              }
            ],
            "product_code": "testSKU1",
            "product_amount": 2000000
          }
        }
      ]
    }
  ]
}
```

Hash Verification Process

In order for us to verify hash we need to click on Hash Verification link or navigate to this path (<http://localhost:5000/payment/hash>)

Hash Verification Form

[Payment Create](#) [Fetch Order Status](#) [EMI Plans](#) [Hash Verification](#)

Merchant ID

106598

Access Code

4a39a6d4-46b7-474d-929d-21bf0e9ed607

Secret

55E0F73224EC458A8EC0B68F7B47ACAE

Gateway Mode

Sandbox

Response Data

enter response data

Hash Verify

STEPS TO FOLLOW

1. Merchant Id, Access Code, Secret Key, Gateway Mode (Testing/Prod), Response Data are required fields and some of them have been pre filled with Test keys you can change according to your requirement.
2. You need to paste response data that you got while creating order or while fetching details of order in Response Data as shown below.

Hash Verification Form

[Payment Create](#) [Fetch Order Status](#) [EMI Plans](#) [Hash Verification](#)

Merchant ID

106598

Access Code

4a39a6d4-46b7-474d-929d-21bf0e9ed607

Secret

55E0F73224EC458A8EC0B68F7B47ACAE

Gateway Mode

Sandbox

Response Data

```
{
  "ppc_MerchantID": "106598",
  "ppc_MerchantAccessCode": "4a39a6d4-46b7-474d-929d-21bf0e9ed607",
  "ppc_PinePGTxnStatus": "7",
  "ppc_TransactionCompletionDateTime": "24/11/2023 11:49:19 AM",
  "ppc_UniqueMerchantTxnID": "1ab05090478948ddad9f9ba456091de1",
  "ppc_Amount": "4000000",
  "ppc_TxnResponseCode": "1",
  "ppc_TxnResponseMessage": "SUCCESS",
  "ppc_PinePGTransactionID": "14215875",
```

Hash Verify

- Now after clicking on Hash Verify button you'll be get Response like True or False whether hash has been verified or not.

Response Data

```
MSWIC3BIYZINDP9VZMZICJYXAN0YMFjayibkio0CBEAXINj03VUDCBVZIBSCYAXIMDAUMDAgoz4g0GvZ0RNLVTEgiWITMFua19pbnKicmVZ0R9YTXKIA3BICMNI0BNKNZZU0UJEZIMDAW
MCwiYmFua19pbnRlcmVZdF9yYXRljo4OTY2MCwicHJvZHVjdF9kaXNwbGF5X25hbWUiOiJ0ZXN0U0tVMsJ9LHsic2NoZW1lcyl6W3sic2NoZW1lX2lkjo0ODEyNCwicHJvZ3JhbV
90eXBlljoXMTIsImZlX3NjaGVtZV92YWxpZCI6dHJ1ZX1dLCJwcm9kdWN0X2NvZGU0iJ0ZXN0U0tVMslnByb2R1Y3RfYW1vdW50ljoyMDAwMDAwLCJzdWJ2ZW50aW9uX2Nhc
2hiYWNRX2Rpc2NvdW50ljo4OTU1MCwicHJvZHVjdF9kaXNjb3VudCI6MTAwMDAslnN1YnZlbnRpb25FY2FzaGJhY2tfZGlzY291bnRfcGVyY2VudGFnZSI6NDUwMDAslnByb2R1Y3
RfZGlzY291bnRfcGVyY2VudGFnZSI6MCwic3VidmVudGlubl90eXBlljoXLCJzcGVjaWFsX29mZmVyX2Nhc2hiYWNRljoIR2V0IERpc2NvdW50IG9mFjZlDEwMC4wMCRvbiB0ZXN0U0
tVMiAiLCJiYW5rX2ludGVyZXN0X3JhdGVfcGVyY2VudGFnZSI6MTYwMDAwLCJiYW5rX2ludGVyZXN0X3JhdGVhZGUiOjg5NjYwLCJwcm9kdWN0X2Rpc3BsYXlfbmFtZSI6InRlc3RTS1U
yln1dLCJlbWlfc2NoZW1lIjp7InNjaGVtZV9pZCI6NDczNTMslnByb2dyYW1fdHlwZSI6MTA1LCJpc19zY2hlcWVfdmFsaWQiOnRydWV9FQ==",
"ppc_AcquirerResponseCode": "CAPTURED",
"ppc_AcquirerResponseMessage": "CAPTURED",
"ppc_AuthId": "999999"
}
```

Hash Verify

Hash verification successful

TDR/GST Information:

Please note no additional charges like TDR, GST, etc are handled in our Plugins and the same needs to be manually handled at the merchant end.

TLS 1.2 information:

- Python 2.7.9 and Python 3.3 and above support TLS 1.2.

Note: The availability may depend on the OpenSSL version linked with Python.