Pinelabs Python SDK Integration

This document explains the integration process of the Pinelabs Python SDK for your Python Applications

By Pinelabs Team

Prerequisites

Python Version>=3

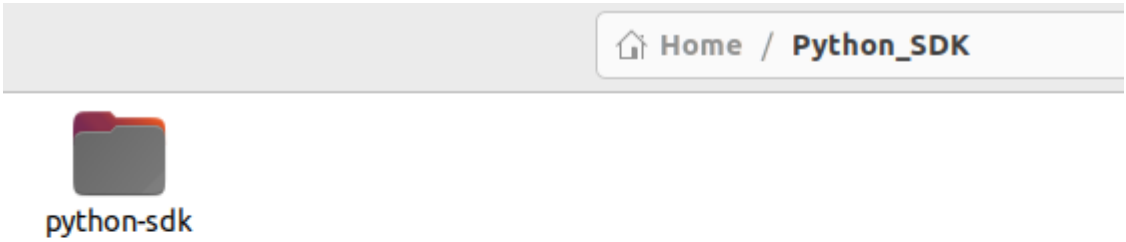
Installation Process

This section explains the installation flow of the Python SDK for dealing with Pine Lab Apis

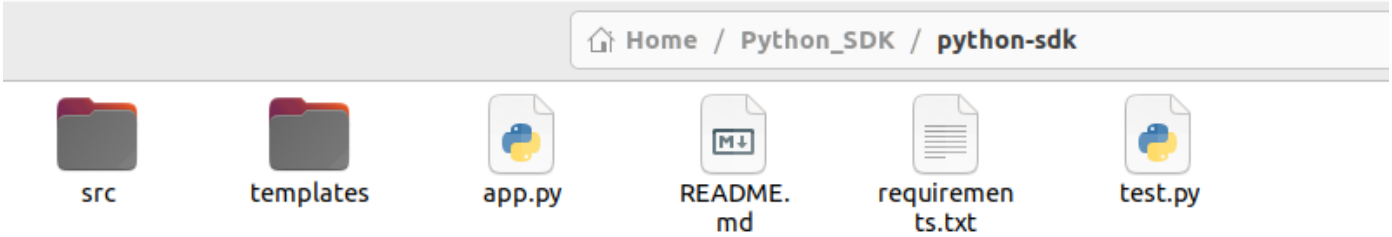
In this example we've created a sample project named `Flask_Sample_PineLab` which will be provided to you along with the SDK for you to be able to run and test out the SDK without actually integrating it. You can change and modify the code tocheck different behavior and responses we get from the SDK.

Given below are the steps for installing the SDK in your project:

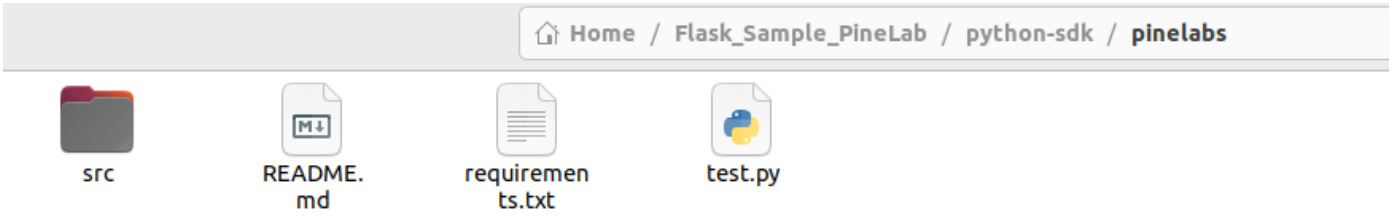
1. Extract the SDK (Zip File) in any location on your system. Here we have extracted it inside Python_SDK Folder (Sdk Folder - python-sdk)



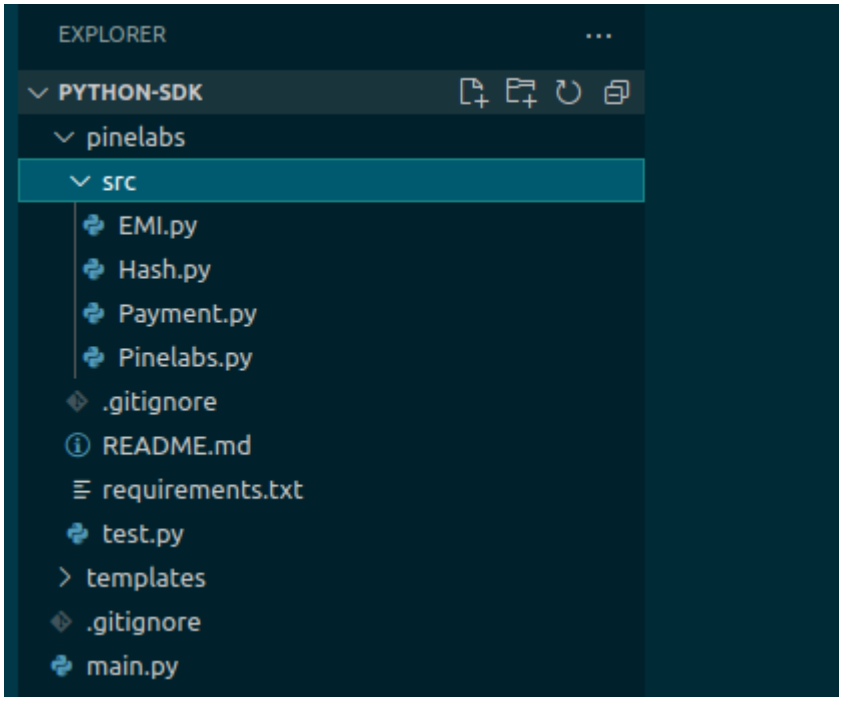
2. Now we need to copy all the files from it to to our flask Project or any Python framework you're using.



3. For sample we have imported files in our Sample Flask Project (Flask_Sample_PineLab), inside newly created folder pinelabs.In this sample you don't have to perform this step as it's already there, but in your existing project where you need to copy it & add.



4. So it'll look something like this in your IDE, where your main classes are also located in other packages.



5. After that you need to import class Pinelabs into your main class, like in our Sample Flask Project we have included classes (`from pinelabs.src.Pinelabs import Pinelabs`)

```
main.py
Aamir-MCSAM, 15 hours ago | 1 author (Aamir-MCSAM)
from flask import Flask, render_template, request, redirect
import uuid
import json
from pinelabs.src.Pinelabs import Pinelabs
```

Usage Process

This section explains the usage flow of the Python SDK

(**Note**- For testing this Flask sample app, you need to run main class and hit the url and navigate to /payment, to test run all 4 methods provided by this sdk.)

Create Order Process

In order for us to create an order and generate a URL for payment we need to use the "create Method of Pinelabs class",

1. Create an instance of the Pinelabs class first and pass the following parameters
    a. Merchant Id of the merchant provided by Pinelabs team
    b. Merchant Access Code also provided by Pinelabs team
    c. Merchant Secret also provided by Pinelabs team
    d. Pg Mode which will be of type boolean (True for Testing and False for Production)
2. Now you need to call create method, pass following data in it
    a. Transaction Data (txn_data)- Required
    b. Customer Data (customer_data)- Optional
    c. Shipping Data (shipping_data)- Optional
    d. Billing Data (billing_data)- Optional
    e. Udf Data (udf_data)- Optional
    f. Payment Modes – Required: Modes of payments you want to be enabled on the checkout using the payment (List of Strings)
    g. Product Details- Required: List of Map objects of product details.

Create Instance of PineLabs SDK

Import pinelabs sdk and create object of Pinelabs class. It takes 4 parameters which are as follows:

1. merchant_id (string) : Merchant ID provided by PineLabs
2. access_code (string) : Merchant Access Code Provided by PineLabs
3. secret (string) : Merchant Secret
4. is_test (boolean) : If using test mode then set this to `true`

```
const pinelabs = Pinelabs("{merchant_id}", "{access_code}", "{secret}", is_test)
```

```
// Transaction Data ( Mandatory )
const txn_data = {
    txn_id: "", // String
    callback: "", // String
    amount_in_paisa: "1000", // String
}
```

```
// Customer Data ( Optional )
const customer_data = {
    email_id: "", // String
    first_name: "", // String
    last_name: "", // String
    mobile_no: "", // String
    customer_id: "", // String
}
```

```
// Billing Data ( Optional )
const billing_data = {
    address1: "", // String
    address2: "", // String
    address3: "", // String
    pincode: "", // String
    city: "", // String
    state: "", // String
    country: "", // String
}
```

```
// Shipping Data ( Optional )
const shipping_data = {
    first_name: "", // String
    last_name: "", // String
    mobile_no: "", // String
    address1: "", // String
    address2: "", // String
    address3: "", // String
    pincode: "", // String
    city: "", // String
    state: "", // String
    country: "", // String
}
```

```
// Payment Modes That Needs To Be Shown ( Mandatory )
const payment_mode = {
    netbanking: true, // Boolean
    cards: true, // Boolean
    emi: true, // Boolean
    upi: true, // Boolean
    cardless_emi: true, // Boolean
    wallet: true, // Boolean
    debit_emi: true, // Boolean
    prebooking: true, // Boolean
    bnpl: true, // Boolean
    paybypoints: false, // Boolean
}
```

```
// Product Details ( Optional, Required For Multicart )
const product_details = [
    {
        "product_code": "testSKU1", // String
        "product_amount": 500000 // Integer
    },
    {
        "product_code": "testSKU1", // String
        "product_amount": 500000 // Integer
    }
]
```

```
// Create Order
try :
    orderCreateResponse = pinelabs.payment.create(txn_data, customer_data, billing_data, shipping_data,
    print(orderCreateResponse)
except Exception as e:
    print("Exception : " , e)
```

```
billing_data, shipping_data, udf_data, payment_mode, products_details)
```

3. It will return the following once executed
    a. **Token**: Payment order token for processing of order
    b. **URL**: URL using which the user can be directly redirected to the payment page where he/she can make the payment.
    c. **Status:** Status of the order create api status if it worked successfully or not.

Success Response


```
{
  "status": true,
  "redirect_url": "https://uat.pinepg.in/pinepg/v2/process/payment?token=S01wPSlIH%2bopelRVif7m7e4SgrTRIcKYx25YD
  "token": "S01wPSlIH%2bopelRVif7m7e4SgrTRIcKYx25YDYfmgtbPOE%3d"
}
```


Failure Response

```
Exception : DUPLICATE TRANSACTION ID RECEIVED FROM MERCHANT
```

Fetch Order Process

In order for us to fetch an order detail we need to use the "fetch Method of Pinelabs class"

1. Create an instance of the Pinelabs class first and pass the following parameters
    a. Merchant Id of the merchant provided by Pinelabs team
    b. Merchant Access Code also provided by Pinelabs team
    c. Merchant Secret also provided by Pinelabs team
    d. Pg Mode which will be of type boolean (True for Testing and False for Production)

2. Now call fetch function of Pinelabs class's object and pass below parameters.
    a. Transaction ID: Transaction id which was used during the order creation process

```
// Fetch Order
try :
    orderFetchResponse = pinelabs.payment.fetch("650acb67d3752")
    print({"response":orderFetchResponse})
except Exception as e:
    print("Exception : " , e)
```

3. Response

Success Response


```
{
  "ppc_MerchantID": "106600",
  "ppc_MerchantAccessCode": "bcf441be-411b-46a1-aa88-c6e852a7d68c",
  "ppc_PinePGTxnStatus": "7",
  "ppc_TransactionCompletionDateTime": "20\/09\/2023 04:07:52 PM",
  "ppc_UniqueMerchantTxnID": "650acb67d3752",
  "ppc_Amount": "1000",
  "ppc_TxnResponseCode": "1",
  "ppc_TxnResponseMessage": "SUCCESS",
  "ppc_PinePGTransactionID": "12069839",
  "ppc_CapturedAmount": "1000",
  "ppc_RefundedAmount": "0",
  "ppc_AcquirerName": "BILLDESK",
  "ppc_DIA_SECRET": "D640CFF0FCB8D42B74B1AFD19D97A375DAF174CCBE9555E40CC6236964928896",
  "ppc_DIA_SECRET_TYPE": "SHA256",
  "ppc_PaymentMode": "3",
  "ppc_Parent_TxnStatus": "4",
  "ppc_ParentTxnResponseCode": "1",
  "ppc_ParentTxnResponseMessage": "SUCCESS",
  "ppc_CustomerMobile": "7737291210",
  "ppc_UdfField1": "",
  "ppc_UdfField2": "",
  "ppc_UdfField3": "",
  "ppc_UdfField4": "",
  "ppc_AcquirerResponseCode": "0300",
  "ppc_AcquirerResponseMessage": "NA"
}
```

EMI Details Process

In order for us to fetch EMI details for any product we need to use the "calculate method of emi class using Pinelabs Class"

1. Create an instance of the Pinelabs class first and pass the following parameters
    a. Merchant Id of the merchant provided by Pinelabs team
    b. Merchant Access Code also provided by Pinelabs team
    c. Merchant Secret also provided by Pinelabs team
    d. Pg Mode which will be of type boolean (True for Testing and False for Production)

2. You need to pass txn data containing amount_in_paisa or total amount, product details list of objects if multicart else list of single object, and you need to pass it into calculate function.

```
// Emi Calculation
const txn_data = {
    amount_in_paisa: "1000",
}

const products_details = [
    {
        "product_code": "testproduct02",
        "product_amount": "10000"
    }
];

try :
    orderEmiResponse = pinelabs.emi.calculate(txn_data, products_details)
    print(orderEmiResponse)
except Exception as e:
    print("Exception : " , e)
```

3. Once executed it will return the following response

Success Response


```
{
  "issuer": [
    {
      "list_emi_tenure": [
        {
          "offer_scheme": {
            "product_details": [
              {
                "schemes": [],
                "product_code": "testproduct02",
                "product_amount": 10000,
                "subvention_cashback_discount": 0,
                "product_discount": 0,
                "subvention_cashback_discount_percentage": 0,
                "product_discount_percentage": 0,
                "subvention_type": 3,
                "bank_interest_rate_percentage": 150000,
                "bank_interest_rate": 251
              }
            ],
            "emi_scheme": {
              "scheme_id": 48040,
              "program_type": 105,
              "is_scheme_valid": true
            }
```


Hash Verification Process

In order for us to verify hash of all the responses we get from the SDK to make sure there was no tampering with the response we need to use the "verify method of hash class using Pinelabs Class"

1. Create an instance of the Pinelabs class first and pass the following parameters
    a. Merchant Id of the merchant provided by Pinelabs team
    b. Merchant Access Code also provided by Pinelabs team
    c. Merchant Secret also provided by Pinelabs team
    d. Pg Mode which will be of type boolean (True for Testing and False for Production)
2. You need to pass response data along with hash you want to verify, but you need to remove hash and hash type from response data and then pass it into method.

```
// Verify Hash
try :
    hash = orderResponse["ppc_DIA_SECRET"]

    keys_to_remove = ["ppc_DIA_SECRET", "ppc_DIA_SECRET_TYPE"]

    for key in keys_to_remove:
        orderResponse.pop(key, None)

    isVerified = pinelabs.hash.verify(hash, orderResponse)
    print(isVerified)
except Exception as e:
    print("Exception : " , e)
```

4. Once Executed it'll return boolean response whether it was verified or not (True/False)

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.