

# Sample App Guide for Testing React JS SDK

Within this document explains the step-by-step process to run the sample app to evaluate the functionality of the React JS SDK

By Pinelabs Team

## Prerequisites

1. NODE version 18.17.1 or higher
2. NPM version 9.6.7 or higher

## Installation

This section explains the installation flow of the Sample App .

1. Extract the Sample App (Zip File) in any location on your system . Open in code editor
  2. In order to run the sample app you'll need to run the following commands.
    - `cd .\sample-app\`
    - `npm link "../React SDK"`
    - `npm install "../React SDK"`
    - `npm run dev`
- 

## Usage Process

This section explains the usage flow of Sample App .

### Create Order Process

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.

## Test Form

Create Order

[Create Order](#) | [Fetch Order details](#) | [Fetch EMI Offers](#) | [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"/>	
Transaction ID	Amount (In Paisa)	Callback URL
<input type="text" value="Transaction ID"/>	<input type="text" value="4000000"/>	<input type="text" value="http://localhost:3000/callback"/>
Product Details		Payment Mode
<pre>{   {     "product_code": "testSKU1",     "product_amount": "2000000"   },   {     "product_code": "testSKU2",     "product_amount": "2000000"   } }</pre>		<div>Select Payment Mode ▾</div>
<div> <div>▼ Customer Details</div> <div>▼ Billing Address</div> <div>▼ Shipping Address</div> <div>▼ Additional Fields</div> </div>		
<div>Pay Now</div>		

- 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.

plurad by pine labs

Order Total: ₹40000

Choose a payment option

- Credit Debit Cards
- Net Banking**
- UPI/BHIM
- Wallets

Select Bank

IDBI Bank

or

IDBI Bank

Pay Now

Safe & Secure Payments

PCI DSS Verified by VISA MasterCard SecureCode SafeKey

- 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 .

## Fetch Order Process

### Test Form

Fetch API Response

[Create Order](#) | [Fetch Order details](#) | [Fetch EMI Offers](#) | [Hash Verification](#)

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

Fetch Details

- 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.
- You need enter Transaction Id of which you previously created Payment or if having txn id for

testing purpose you can enter that as well.

3. Now you need to click on fetch order and it'll give you response something like this

### 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": "D640CFF0FCB8D42B74B1AFD19D97A375DAF174CCBE9555E4
0CC6236964928896",
  "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"
}
```

### Failure Response

```
{
  "ppc_MerchantID": "106600",
  "ppc_MerchantAccessCode": "bcf441be-411b-46a1-aa88-c6e852a7d68c",
  "ppc_PinePGTxnStatus": "-6",
  "ppc_TransactionCompletionDateTime": "21\09\2023 11:29:48 PM",
  "ppc_UniqueMerchantTxnID": "106600_2109202323294890763",
  "ppc_TxnResponseCode": "-40",
  "ppc_TxnResponseMessage": "INVALID DATA",
  "ppc_CapturedAmount": "0",
  "ppc_RefundedAmount": "0",
  "ppc_DIA_SECRET": "4B9DD62C1CE94C354E368A2DA1C51C2E8ED16ABDC46414B8AAA60F37
8CDCE390",
}
```

```
} "ppc_DIA_SECRET_TYPE": "SHA256"
```

Incorrect Merchant Details

"IP Access Denied"

---

## EMI Detail Process

### Test Form

EMI API Response

[Create Order](#) | [Fetch Order details](#) | [Fetch EMI Offers](#) | [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"/>
Amount (In Paisa)	
<input type="text" value="4000000"/>	
Product Details	
<pre>[   {     "product_code": "testSKU1",     "product_amount": "2000000"   },   {     "product_code": "testSKU2",     "product_amount": "2000000"   } ]</pre>	

XAMPP Control Panel v3.3.0 | Compiled: Apr 6th 2021 |

Fetch EMI Options

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 EMI button you'll be able to see response like this

## 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
            }
          },
          "tenure_id": "3",
          "tenure_in_month": "3",
          "monthly_installment": 3417,
          "bank_interest_rate": 150000,
          "interest_pay_to_bank": 251,
          "total_offered_discount_cashback_amount": 0,
          "loan_amount": 10000,
          "auth_amount": 10000
        },
        {
          "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": 440
    },
    ],
    "emi_scheme": {
        "scheme_id": 48040,
        "program_type": 105,
        "is_scheme_valid": true
    }
},
"tenure_id": "6",
"tenure_in_month": "6",
"monthly_installment": 1740,
"bank_interest_rate": 150000,
"interest_pay_to_bank": 440,
"total_offered_discount_cashback_amount": 0,
"loan_amount": 10000,
"auth_amount": 10000
}
],
"issuer_name": "HDFC",
"is_debit_emi_issuer": false
}
],
"response_code": 1,
"response_message": "SUCCESS"
}

```

### Failure Response

Fatal error: Uncaught Exception: INVALID DATA,MISMATCH\_IN\_TOTAL\_CART\_AMOUNT\_AND\_TOTAL\_PRODUCT\_AMOUNT

---

### Hash Verification Process

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

[Create Order](#) | [Fetch Order details](#) | [Fetch EMI Offers](#) | [Hash Verification](#)

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

Response Data

```
ppc_ParentTxnStatus : 4 ,
"ppc_ParentTxnResponseCode": "1",
"ppc_ParentTxnResponseMessage": "SUCCESS",
"ppc_CustomerMobile": "",
"ppc_UdfField1": "",
"ppc_UdfField2": "",
"ppc_UdfField3": "",
"ppc_UdfField4": "",
"ppc_AcquirerResponseCode": "0300",
"ppc_AcquirerResponseMessage": "DEFAULT"
```

Hash Verify

XAMPP Control Panel v3.3.0 | Compiled: Apr 6th 2021 |

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

Hash verification successful

### React Js :-

React.js itself does not handle TLS directly, as it's a JavaScript library for building user Interfaces. The TLS support is typically handled by the underlying server or service your React.js application communicates with.

### Note :-

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

### Note – Browser Compatibility :-

This React SDK is Compatible for non-Chromium-based browsers like Mozilla Firefox. However, because Chromium-based browsers, such as Google Chrome, include the following headers:

- Sec-Ch-Ua: "Google Chrome";v="119", "Chromium";v="119", "Not?A\_Brand";v="24"
- Sec-Ch-Ua-Mobile: ?0
- Sec-Ch-Ua-Platform: "Windows"

Due to the nature of React.js, a JavaScript library for building user interfaces, it's not feasible to remove or modify these headers.