Pinelabs .Net SDK Integration

This document explains the integration process of the Pinelabs .Net SDK for your .Net applications

By Pinelabs Team

Installation Process
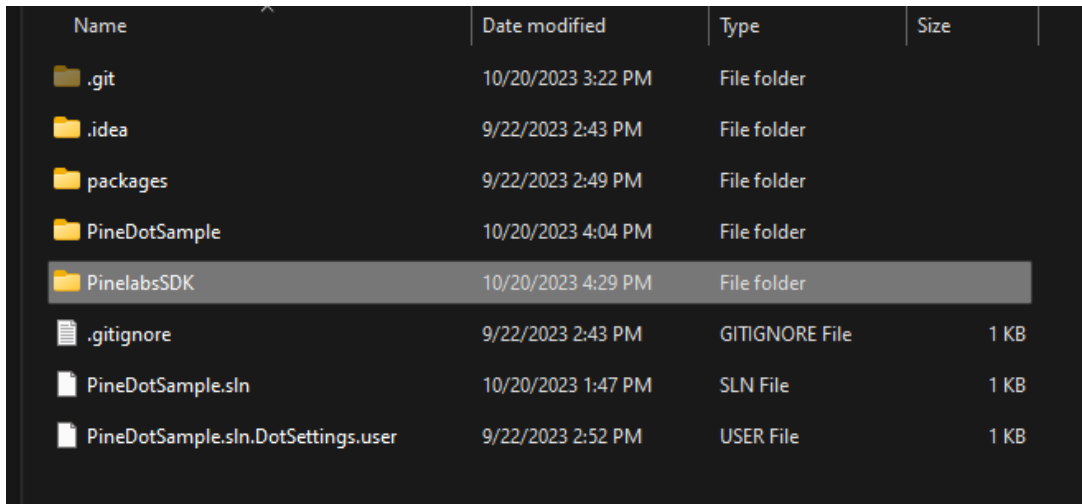
Note: By default, **.NET Framework 4.7 and later versions are configured to use TLS 1.2 and allow connections using TLS 1.1 or TLS 1.0**

This section explains the installation flow of the .Net SDK for Pinelabs.

In this example we've created a sample project named `PineDotSample` 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 to check 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 your project folder for easy use. Best to extract it in the root of the project like we've extracted it in the solutions folder.

2. Now we need to open up our .Net IDE (Integrated Development Environment) for this example we're using the JetBrains Rider IDE for .Net. You can however use any IDE; you like it doesn't matter at all. However, do note that following steps might change depending on the IDE you're using.
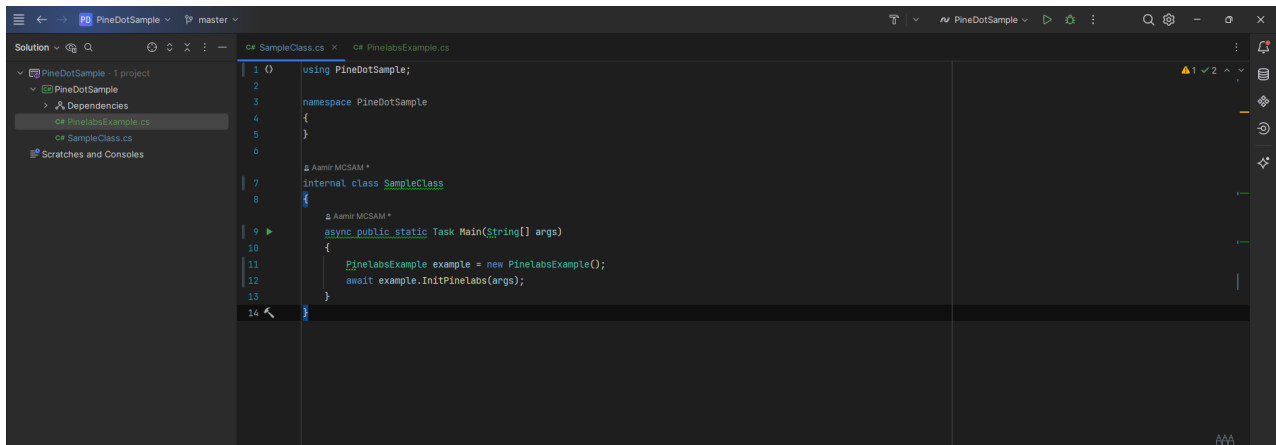
3. Now we first need to include the assembly (.dll) from the .Net SDK in our project in order for us to use it. In order to include the DLL file in the project in the Rider IDE we can right click on the dependencies section on the left-hand side menu
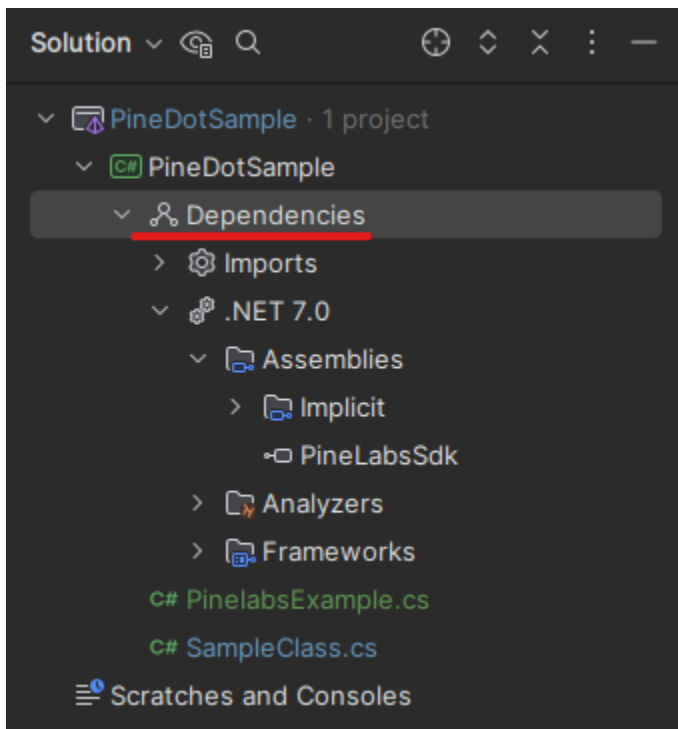
4. Click on the reference option in the menu and then a References dialog will open for you to add the Reference.
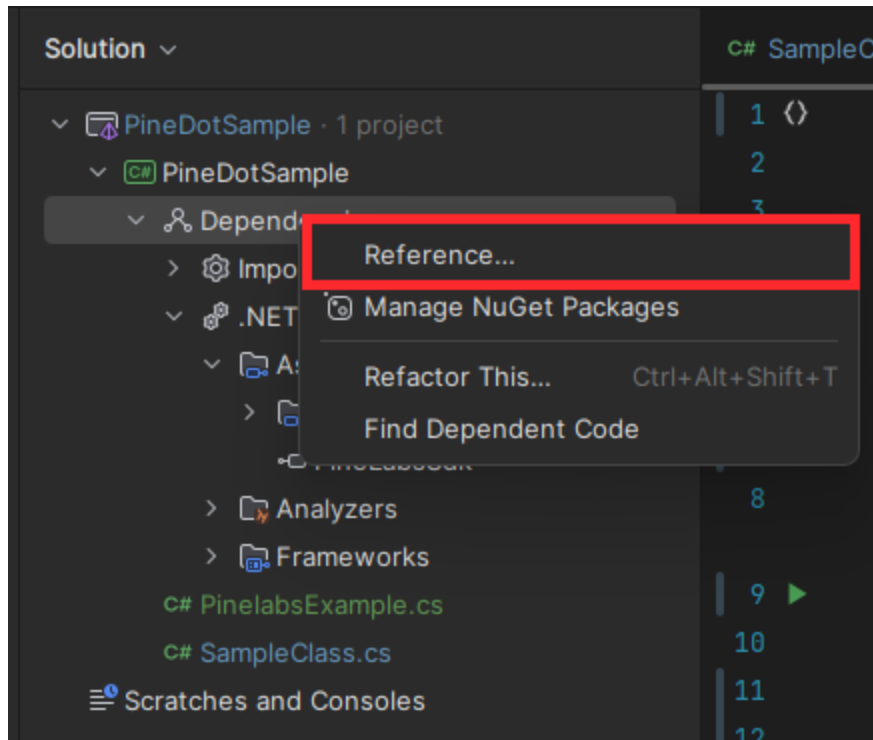
5. Click on the Add from button at the bottom of the dialog and then select the path of the DLL file from the SDK folder in your project root where we extracted the SDK from the zip.
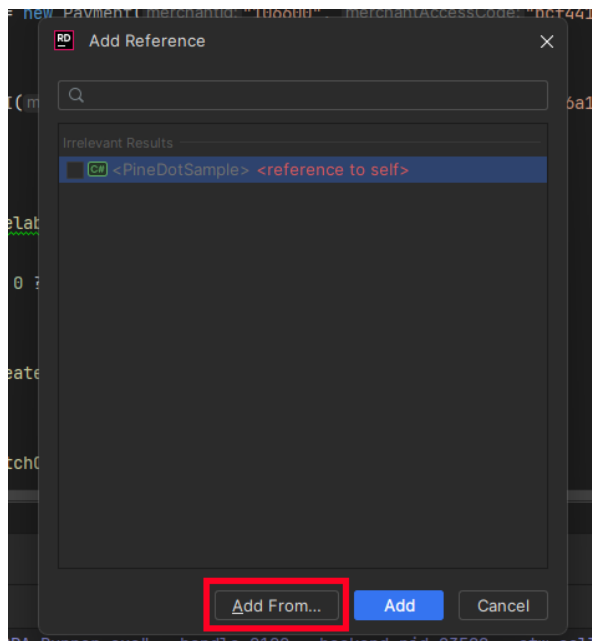
In our case that DLL file is present in "{Project Root}\PinelabsSDK\bin\Debug\net7.0" then select the DLL file and then click open
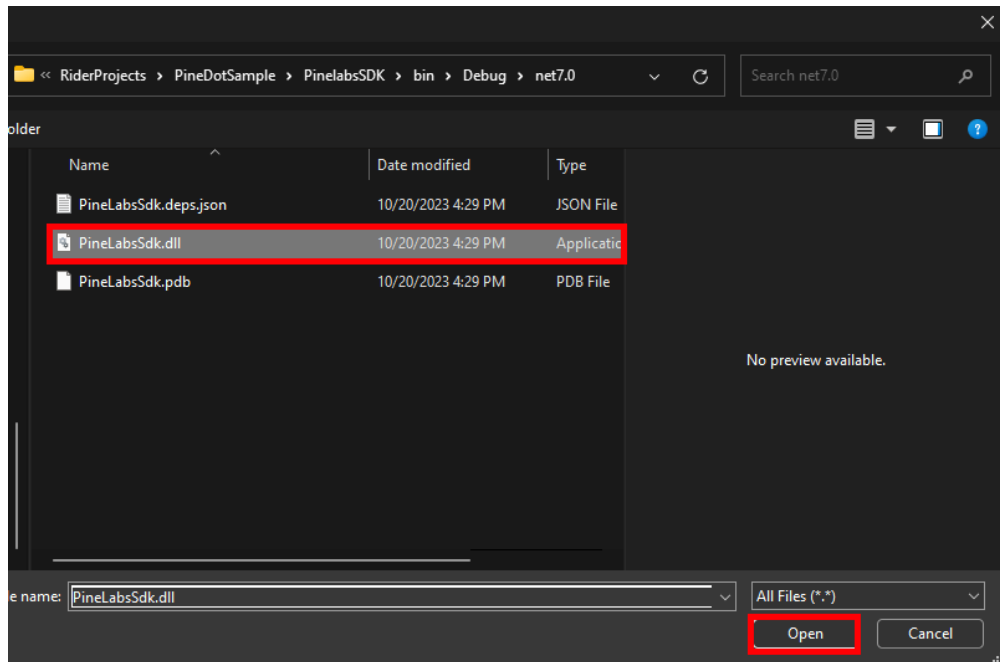
Now that we've finished installing our SDK and adding our SDK in the project it's time to use it. In the next section we'll explain how to use and call different methods available on the SDK for you to use.

Usage Process

This section explains the usage flow of the .Net SDK for Pinelabs.

Use the namespace "*PineLabsSdk*" in any file in which you want to use Pinelabs
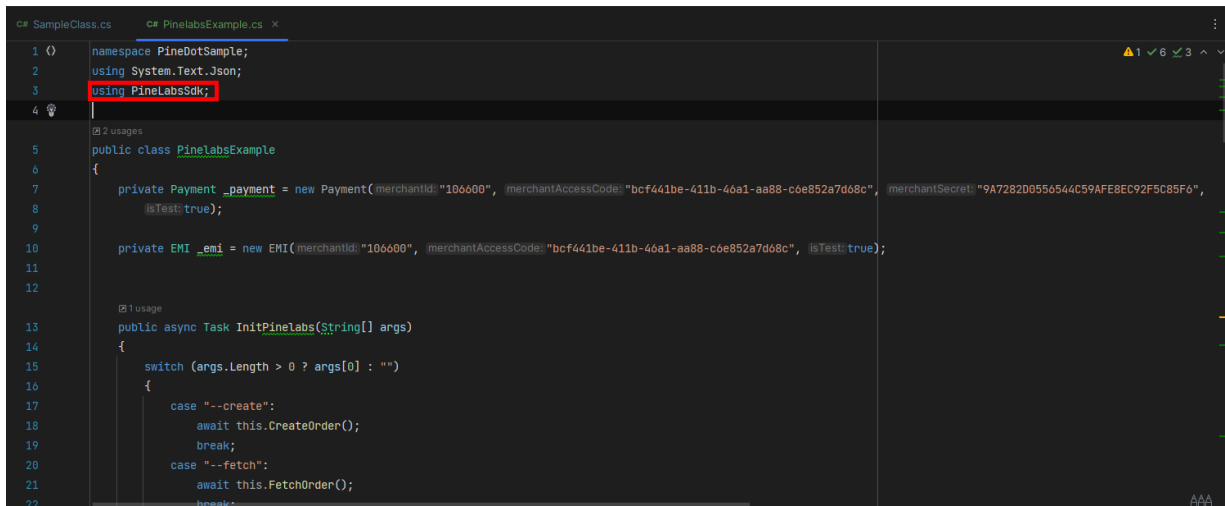
Now we have 3 classes which we can use in the Pinelabs SDK

1. Payment Class – Used for creating payment links and fetching payment details for a transaction id
2. EMI Class – Used for calculation of EMI data for a product
3. Hash Class – Used for hash verification in your application for secure development

Create Order Process

In order for us to create an order and generate a URL for payment we need to use the "Payment Class"

1. Create an instance of the Payment 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

```
private Payment _payment = new Payment(merchantId: "106600", merchantAccessCode: "bcf441be-411b-46a1-aa88-c6e852a7d68c", merchantSecret: "9A7282D0556544C59AFE8EC92F5C85F6", isTest: true);
```

2. Now call the "Create" method on the instance of the Payment class and pass the following arguments
   a. Transaction Data – Required: Details of the transactions like the amount, callback, transaction id.
   b. Customer Data – Required: Customer information including the first name, last name, mobile number, customer id, email etc.
   c. Billing Data – Optional: Billing details of the customer
   d. Shipping Data – Optional: Shipping details of the customer
   e. UDF Data – Optional: UDF data which can be anything extra you want to include in the order
   f. Payment Modes – Required: Modes of payments you want to be enabled on the checkout using the payment

   Given below is an example of how you call it with the required params passed in the

```
private async Task CreateOrder()
{
    var transactionData = new txn_data();

    transactionData.txn_id = new Random().Next(999, 999999999).ToString();
    transactionData.amount_in_paisa = "10000";
    transactionData.callback = "https://httpbin.org/post";

    var customerData = new customer_data();
    customerData.email_id = "test@pinelabs.in";
    customerData.first_name = "John";
    customerData.last_name = "Doe";
    customerData.mobile_no = "92842714583";
    customerData.customer_id = "43243242423242432";

    var paymentModes = new payment_mode();

    paymentModes.cards = true;
    paymentModes.bnpl = true;

    // Create Order
    var response:CreateOrderReturnType? = await this._payment.Create(transactionData, customerData, new billing_data(),
        new shipping_data(),
        new udf_data(),
        paymentModes); // Task<CreateOrderReturnType?>

    Console.WriteLine(JsonSerializer.Serialize(response));
}
```

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.

```
{"token":"S01%2fGkV5TCwlCfCrlw4qej%2f8aLDtoQhzGS%2b%2fiNar3AXEbQ%3d","url":"https://uat.pinepg.in/pinepg/v2/process/paym
ent?token=S01%2fGkV5TCwlCfCrlw4qej%2f8aLDtoQhzGS%2b%2fiNar3AXEbQ%3d","status":true}
```

Fetch Order Process

In order for us to fetch an order detail we need to use the "Payment Class"

1. We'll call the "Fetch" method and pass it the following
    a. Transaction ID: Transaction id which was used during the order creation process explained at the top
    b. Transaction Type: Type of the transaction which was passed at the order creation process. This needs to be the same otherwise it won't work.

```
↗ 1 usage
private async Task FetchOrder()
{
    // Fetch Order
    var response = await this._payment.Fetch(txnId: "650acb67d3752", txnType: 3);
    Console.WriteLine(JsonSerializer.Serialize(response));
}
```

2. It will give you the following response when executed

```
{"ppc_MerchantID":"106600","ppc_MerchantAccessCode":"bcf441be-411b-46a1-aa88-c6e852a7d68c","ppc_PinePGTxnStatus":"7","pp
c_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":"A36BFCDE7745E7D889C6B57FD8FC4F5861DC74805
B4C42A7414A9BE3A928543D","ppc_DIA_SECRET_TYPE":"SHA256","ppc_PaymentMode":"3","ppc_Parent_TxnStatus":"4","ppc_ParentTxnR
esponseCode":"1","ppc_ParentTxnResponseMessage":"SUCCESS","ppc_CustomerMobile":"7737291210","ppc_UdfField1":"","ppc_UdfF
ield2":"","ppc_UdfField3":"","ppc_UdfField4":"","ppc_AcquirerResponseCode":"0300","ppc_AcquirerResponseMessage":"DEFAULT
"}
```

EMI Details Process

In order for us to fetch EMI details for any product we need to use the "EMI Class"

1. First create an instance of the class like shown below and pass it 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

```
private EMI _emi = new EMI(merchantId: "106600", merchantAccessCode: "bcf441be-411b-46a1-aa88-c6e852a7d68c", isTest: true);
```

2. Now that we have the instance of the class we can use it like the following
    a. It takes the total amount of the products in the array of product details
    b. It also takes an array of all the product details which are just product code, product amount

```csharp
🖉 1 usage
private async Task EmiCalculator()
{
    // Emi Calculator
    var product1 = new product_details();
    product1.product_code = "testSKU1";
    product1.product_amount = 500000;

    product_details[] details = { product1 };
    var response:RootObject? = await this._emi.CalculateEmi(amountInPaisa: "500000", details);

    Console.WriteLine(JsonSerializer.Serialize(response));
}
```

3. Once executed it will return the following response

{"issuer":[{"list_emi_tenure":[{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":3,"bank_interest_rate":12550}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate":0}],"tenure_id":"3","tenure_in_month":"3","monthly_installment":170850,"bank_interest_rate":150000,"interest_pay_to_bank":12550,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"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":22096}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate_percentage":0,"bank_interest_rate":0}],"tenure_id":"6","tenure_in_month":"6","monthly_installment":87016,"bank_interest_rate":150000,"interest_pay_to_bank":22096,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"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":31765}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate_percentage":0,"bank_interest_rate":0}],"tenure_id":"9","tenure_in_month":"9","monthly_installment":59085,"bank_interest_rate":150000,"interest_pay_to_bank":31765,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate":0}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate":0}],"tenure_id":"96","tenure_in_month":"1","monthly_installment":0,"bank_interest_rate":0,"interest_pay_to_bank":0,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null}],"issuer_name":"HDFC","is_debit_emi_issuer":false},{"list_emi_tenure":[{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":3,"bank_interest_rate_percentage":140000,"bank_interest_rate":13718}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate_percentage":0,"bank_interest_rate":0}],"tenure_id":"3","tenure_in_month":"3","monthly_installment":170570,"bank_interest_rate":140000,"interest_pay_to_bank":11710,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"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":22096}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate_percentage":0,"bank_interest_rate":0}],"tenure_id":"6","tenure_in_month":"6","monthly_installment":87016,"bank_interest_rate":150000,"interest_pay_to_bank":22096,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":3,"bank_interest_rate_percentage":140000,"bank_interest_rate":29614}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate_percentage":0,"bank_interest_rate":0}],"tenure_id":"9","tenure_in_month":"9","monthly_installment":58846,"bank_interest_rate":140000,"interest_pay_to_bank":29614,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate":41540}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate":0}],"tenure_id":"12","tenure_in_month":"12","monthly_installment":45129,"bank_interest_rate":150000,"interest_pay_to_bank":41540,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null},{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate":0}],"tenure_id":"96","tenure_in_month":"1","monthly_installment":0,"bank_interest_rate":0,"interest_pay_to_bank":0,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null}],"issuer_name":"ICICI","is_debit_emi_issuer":false},{"list_emi_tenure":[{"offer_scheme":{"product_details":[{"schemes":[],"product_code":"testSKU1","product_amount":500000,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"additional_cashback":"10% upto 1000","bank_interest_rate_percentage":0,"bank_interest_rate":0}],"product_code":null,"product_amount":0,"subvention_cashback_discount":0,"product_discount":0,"subvention_cashback_discount_percentage":0,"product_discount_percentage":0,"subvention_type":0,"bank_interest_rate_percentage":0,"bank_interest_rate":0}],"tenure_id":"96","tenure_in_month":"1","monthly_installment":0,"bank_interest_rate":0,"interest_pay_to_bank":0,"total_offerred_discount_cashback_amount":0,"loan_amount":500000,"auth_amount":500000,"emi_scheme":null}],"issuer_name":"Kotak Debit","is_debit_emi_issuer":true}],"response_code":1,"response_message":"SUCCESS"}

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 "Hash Class"

1. We don't need to create instance of the hash class as it provides a single static method "VerifyHash" which will return a Boolean value depending if the hash is matched or not In the response
2. Pass the following params to the method
   a. **PCC_DIA_SECRET:** Pass the hash received in the response of any of the API's
   b. **Response Body**: Pass the entire response body **without the hash and hash type**
   c. **Merchant Secret:** Pass it the merchant secret which again is provided by the Pinelabs team

```
⊠ 1 usage
private static void VerifyHash()
{
    // Verify Hash
    var isVerified = Hash.VerifyHash("D640CFF0FCB8D42B74B1AFD19D97A375DAF174CCBE9555E40CC6236964928896", request: new
    {
        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_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"
    }, merchantSecret: "9A7282D0556544C59AFE8EC92F5C85F6");
    Console.WriteLine(isVerified);
}
```

Doing the following this will return a Boolean value (true/false) based on if the hash matches or not.

Sample Project

You can also check, change and run the sample project provided to you with the SDK itself using the following steps

1. Open the sample project in the IDE of your choice and then click on the Run / Execute button at the top and change code in the "PinelabsExample" class
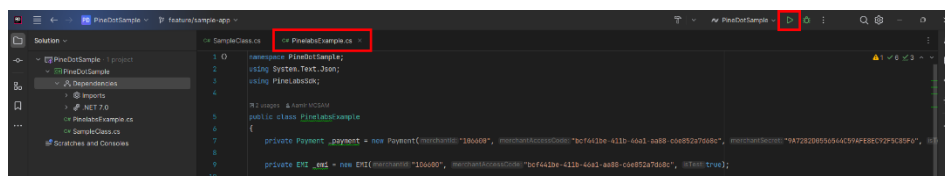
Note:

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.