

IoT Integration Guide

Overview

Post integration the iOS app would be able to accept payments using our iOS SDK. Our libraries PCI-DSS Compliant hence data can be directly passed to our SDK.

Prerequisites

- Please ensure Merchant Id, API key and Salt. (For these you need to register)
- Version Mapping :
 - MacOS => macOS Catalina version 10.15.6 and above
 - XCode => Version -11.7 and above
 - iOS => Version-10 and above
 - Swift => Version -4 and above

Integrating TraknPay SDK

- Import the standard TraknPay iOS SDK library.
- Navigate to TraknPay folder and add our framework into your existing XCode project.

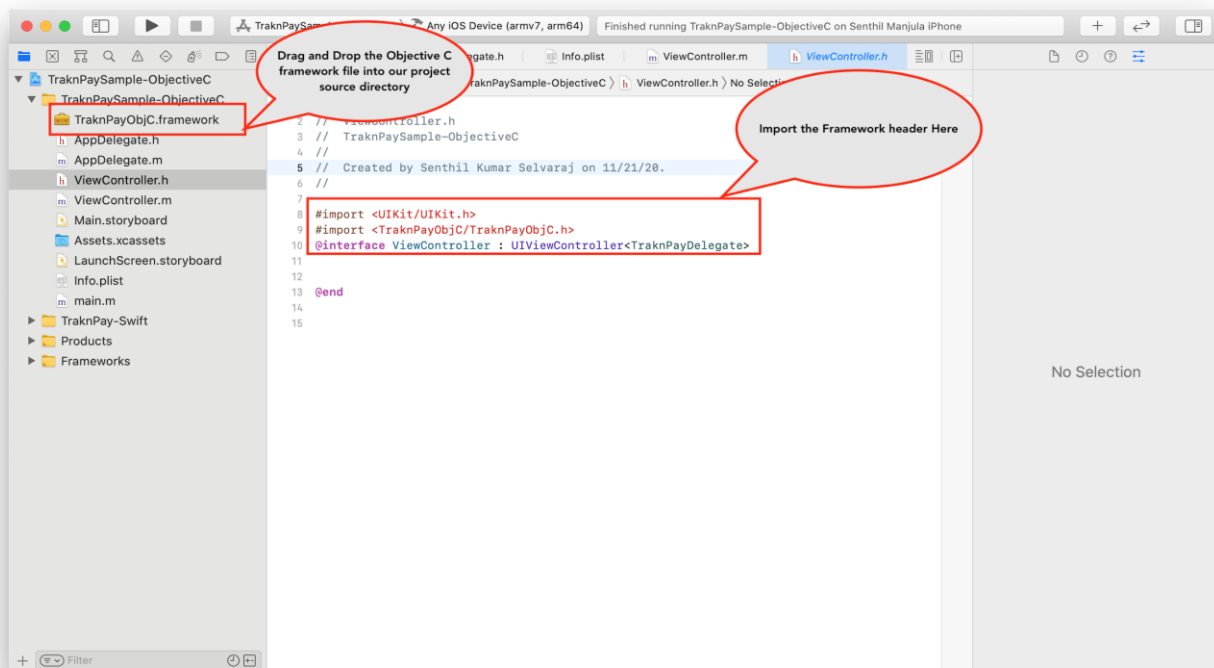
In order to embed, check if TraknPay framework is included or not, if not then please add the framework again.

1. How to Integrate SDK

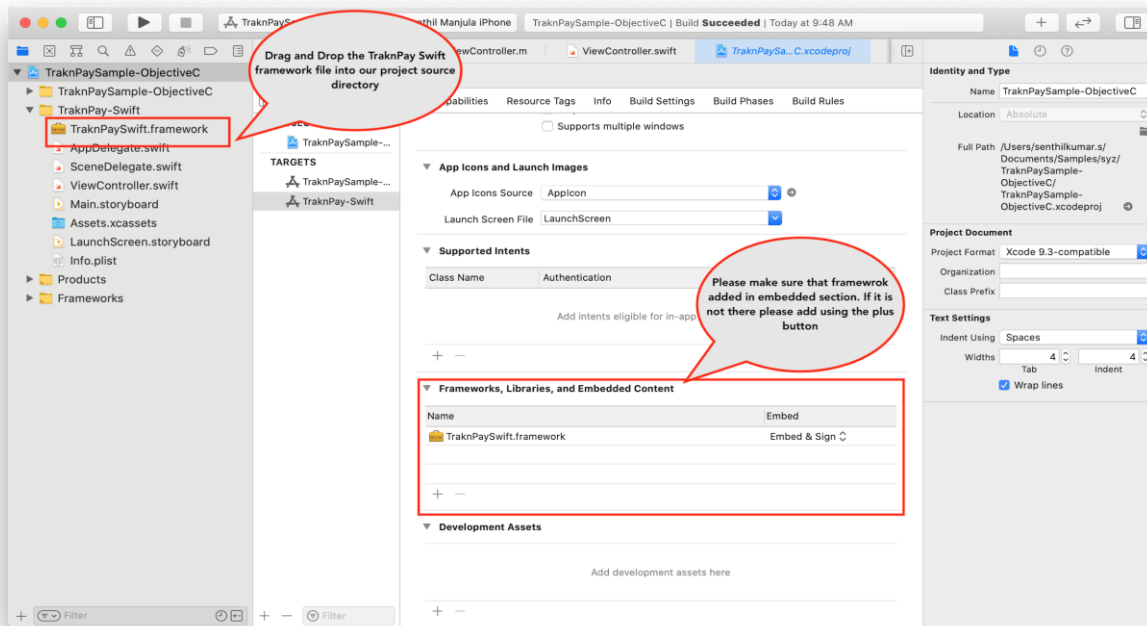
Step 1: Create a new project using XCode 11.7 and above.

Step 2: Drag and drop the TraknPay framework into your project

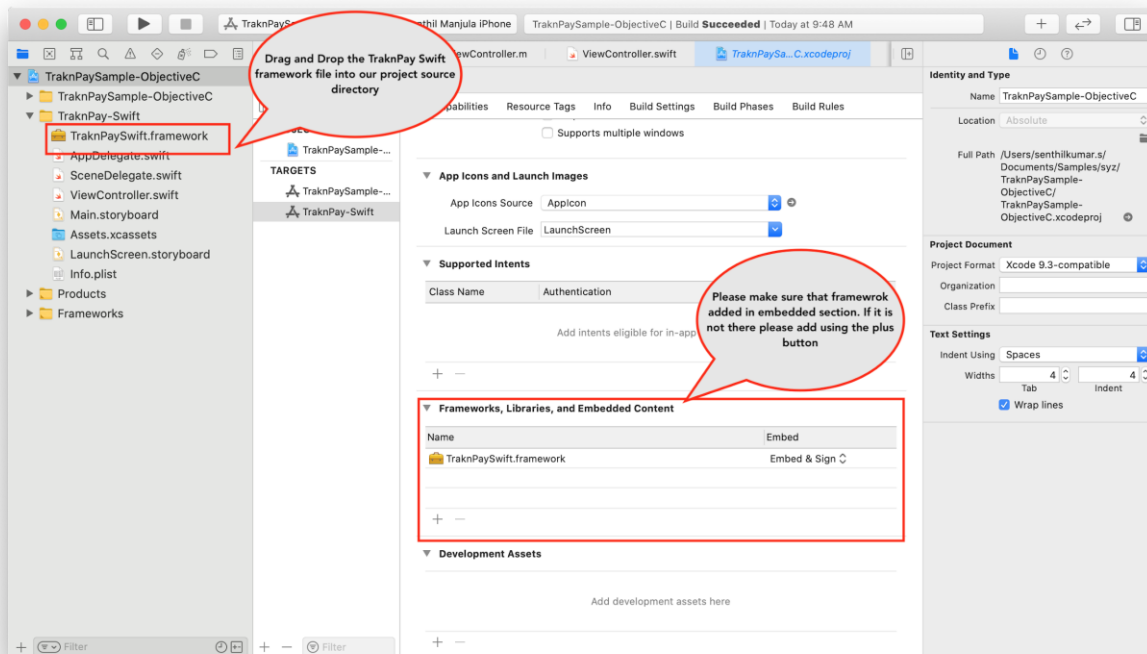
➤ FOR OBJECTIVE C



➤ FOR SWIFT



Step 3: Please make sure that TraknPay framework file added in embedded framework section

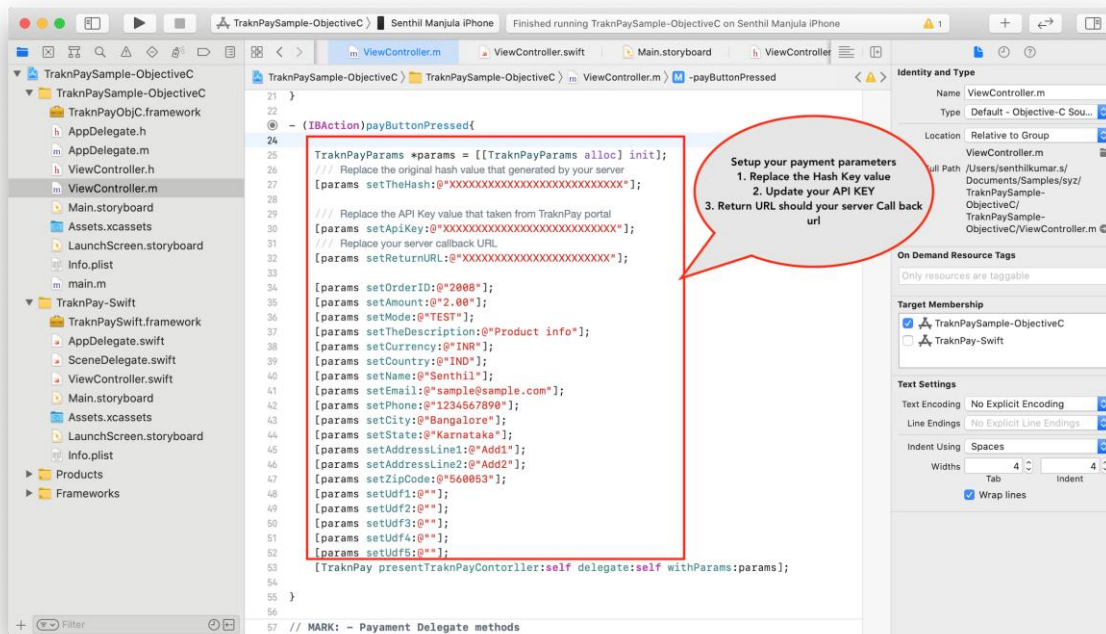


Step 4: Then set up your Payment parameters. Please make sure following parameters updated through your dashboard

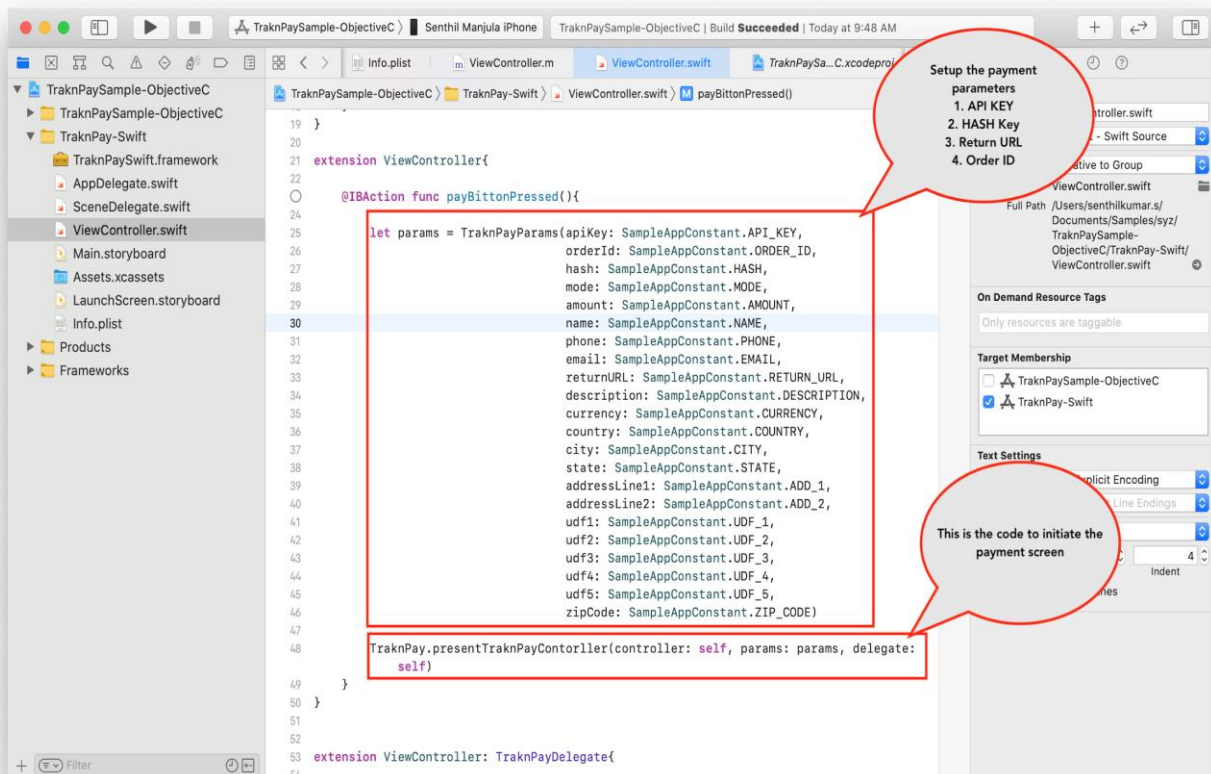
1. **API KEY** - Take this one from the client TraknPay portal.

2. **HASH KEY** - Client server should generate that unique key based on SALT KEY and payment parameters
3. **Return URL** - this is client server call back URL
4. **ORDER ID** - This is unique and it should be generate by client server

➤ For Objective C

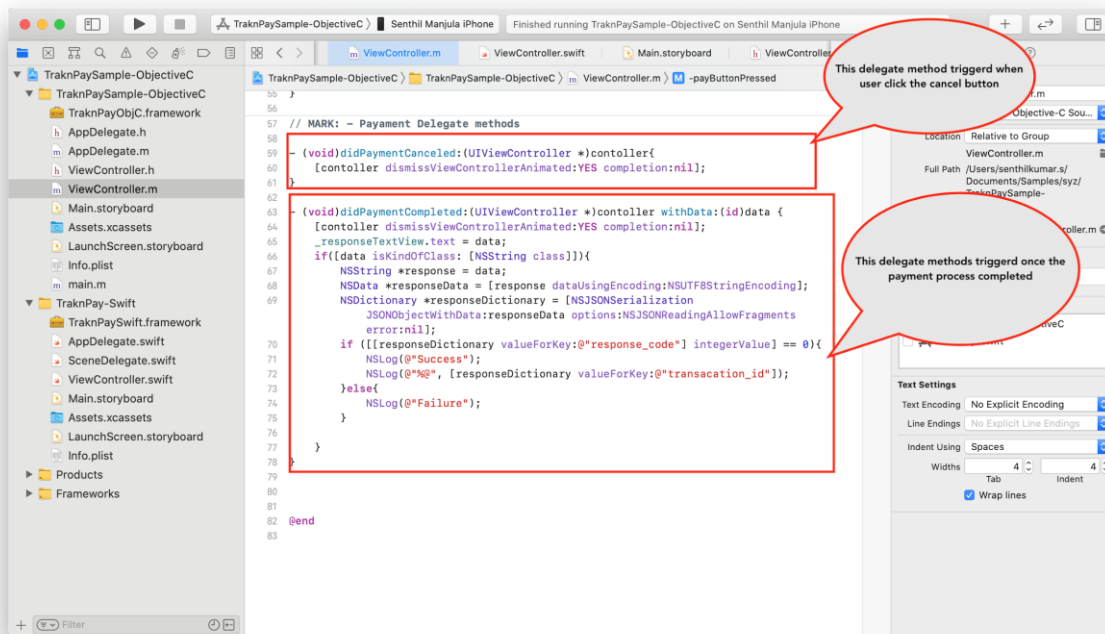


➤ For SWIFT

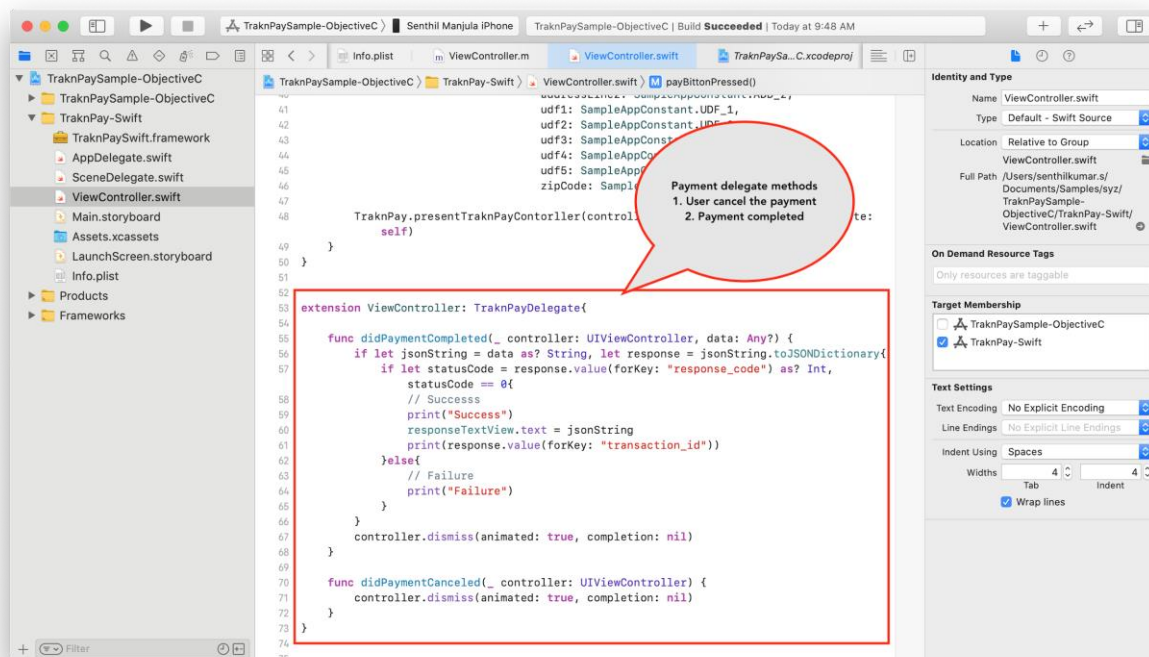


Step 5: Make sure that you have placed the payment delegate methods

➤ For Objective C



➤ For Swift



Step 6: Please make sure follow things enabled:

1. App Transport Security field updated in Info.plist file
2. Please add WebKit.framework if you get any crash related to WebKit framework

2. Code Integration

➤ Using Objective C

❖ Payment Trigger Function

```
/// This line trigger the payment screen
[TraknPay presentTraknPayContorller:self delegate:self withParams:params];
/// This is ONLY for merchant

[TraknPay presentTraknPayContorller:self url:@"https://biz.aggrepaypayments.com"
delegate:self withParams:params];
```

❖ Delegate Methods

```
/// Trigger while user cancel the payment
- (void)didPaymentCanceled:(UIViewController *)controller{
    [controller dismissViewControllerAnimated:YES completion:nil];
}

/// This method triggered once the payment completed
- (void)didPaymentCompleted:(UIViewController *)controller withData:(id)data {
    [controller dismissViewControllerAnimated:YES completion:nil];
    _responseTextView.text = data;
    if([data isKindOfClass: [NSString class]]){
        NSString *response = data;
        NSData *responseData = [response dataUsingEncoding:NSUTF8StringEncoding];
        NSDictionary *responseDictionary = [NSJSONSerialization JSONObjectWithData:responseData
options:NSJSONReadingAllowFragments error:nil];
        if ([[responseDictionary valueForKey:@"response_code"] integerValue] == 0){
            NSLog(@"Success");
            NSLog(@"%@", [responseDictionary valueForKey:@"transacation_id"]);
        }else{
            NSLog(@"Failure");
        }
    }
}
```

➤ Using Swift

❖ Payment Trigger Function:

```
/// This line trigger the payment screen
TraknPay.presentTraknPayContorller(controller: self, params: params, delegate: self)

/// This is ONLY for merchant
TraknPay.presentTraknPayContorller(controller: self, url: " https://biz.aggrepaypayments.com", params:
params, delegate: self)
```

❖ Delegate Methods

/// Trigger while user cancel the payment

```
func didPaymentCanceled(_ controller: UIViewController) {  
    controller.dismiss(animated: true, completion: nil)  
}
```

/// This method triggered once the payment completed

```
func didPaymentCompleted(_ controller: UIViewController, data: Any?) {  
    if let jsonString = data as? String, let response = jsonString.toJSONDictionary{  
        if let statusCode = response.value(forKey: "response_code") as? Int, statusCode == 0{  
            // Success  
            print("Success")  
            responseTextView.text = jsonString  
            print(response.value(forKey: "transaction_id"))  
        }else{  
            // Failure  
            print("Failure")  
        }  
    }  
    controller.dismiss(animated: true, completion: nil)  
}
```

Request Parameters

PARAMETER NAME	DESCRIPTION	REQUIRED	DATATYPE
api_key	Traknpay would assign a unique 40-digit merchant key to you. This key is exclusive to your business/login account.If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	Mandatory	String - Max:40.
order_id	This is your (merchant) reference number. It must be unique for every transaction. We do perform a validation at our end and do not allow duplicate order_ids for the same merchant.	Mandatory	String - Max:30.

PARAMETER NAME	DESCRIPTION	REQUIRED	DATATYPE
mode	This is the payment mode ("TEST" or "LIVE" are valid values). "LIVE" is the default value when not specified.	Optional	String - Max:4.
amount	This is the payment amount.	Mandatory	Decimal - Max Digits Before Decimal:15, Max Digits after Decimal:2.
currency	This is the 3-digit currency code (INR).	Mandatory	String - Max:3.
description	Brief description of product or service that the customer is being charged for.	Mandatory	String - Max:200.
name	Name of customer.	Mandatory	String - Max:200.
email	Customer email address.	Mandatory	String - Max:200.
phone	Customer phone number.	Mandatory	String - Max:30.
address_line_1	Customer's address line 1.	Optional	String - Max:255.
address_line_2	Customer's address line 2.	Optional	String - Max:255.
city	Customer City.	Mandatory	String - Max:50.
state	Customer State.	Optional	String - Max:50.
country	Customer Country.	Mandatory	String - Max:50.
zip_code	Customer Zipcode.	Mandatory	String - Max:10.
timeout_duration	Timeout duration (in seconds).	Optional	Integer - Min:0,Max:1000.
udf1	User defined field 1.	Optional	String - Max:300.
udf2	User defined field 2.	Optional	String - Max:300.
udf3	User defined field 3.	Optional	String - Max:300.
udf4	User defined field 4.	Optional	String - Max:300.
udf5	User defined field 5.	Optional	String - Max:300.
return_url	TThis API needs to be created by the Client using their web	Mandatory	String - Max:200.

PARAMETER NAME	DESCRIPTION	REQUIRED	DATATYPE
	server to which, Traknpay will automatically send all the response after a payment as a POST request after a payment is completed. Client is required to verify the hash and store the data to their database.This must be HTTPS, not HTTP to ensure a secured line.		
return_url_failure	Traknpay will send all failed transaction response parameters to this URL if specified, else, it will send the failed response to the "return_url" parameter	Optional	String - Max:200.
return_url_cancel	Traknpay will send all cancelled transaction response parameters to this URL if specified, else, it will send the cancelled response to the "return_url" parameter.	Optional	String - Max:200.
percent_tdr_by_user	Percent of tdr amount paid by user.	Optional	Integer - Min:0,Max:100.
flatfee_tdr_by_user	Fixed fee paid by user.	Optional	Integer - Min:0,Max:99999999.
show_convenience_fee	Controls whether the convenience fee amount (for surcharge merchants) is displayed to the customer (on the payment page) or not.	Optional	String - Max:1.
split_enforce_strict	Controls whether payment is required to be split before settlement. By default it is set to 'n', If this is set to 'y' then settlement will be on HOLD until splitsettlement api is called to provide split information.	Optional	String - Max:1.
payment_options	payment options to be displayed such credit card(cc), netbanking(nb), wallet(w) and	Optional	String - Max:20.

PARAMETER NAME	DESCRIPTION	REQUIRED	DATATYPE
	atm card(atm).Tabs will be displayed by order in which values are sent. Values accepted are: cc,nb,w,atm (comma separated string).		
payment_page_display_text	This text will be displayed below the logo on payment page.	Optional	String - Max:200.

Response Codes

Below are the response codes that comes in the payment response post payment from our server, that must be handled by the client.

Code	Response Message	Description
0	SUCCESS	Transaction successful
1000	FAILED	Transaction failed
1001	INVALID-API-KEY	The api key field is incorrect
1002	INVALID-LIVE-MODE-ACCESS	The live mode access is not allowed
1003	INVALID-ORDER-ID-FIELD	The order id field should to be unique
1004	ORDER-ID-FIELD-NOT-FOUND	The order id field is not found
1005	INVALID-AUTHENTICATION	Invalid authentication at bank
1006	WAITING-BANK-RESPONSE	Waiting for the response from bank
1007	INVALID-INPUT-REQUEST	Invalid input in the request message
1008	TRANSACTION-TAMPERED	Transaction tampered
1009	DECLINED-BY-BANK	Bank Declined Transaction
1010	INVALID-AMOUNT	Amount cannot be less than 1
1011	AUTHORIZATION-REFUSED	Authorization refused
1012	INVALID-CARD	Invalid Card/Member Name data
1013	INVALID-EXPIRY-DATE	Invalid expiry date
1014	DENIED-BY-RISK	Transaction denied by risk

Code	Response Message	Description
1015	INSUFFICIENT-FUND	Insufficient Fund
1016	INVALID-AMOUNT-LIMIT	Total Amount limit set for the terminal for transactions has been crossed
1017	INVALID-TRANSACTION-LIMIT	Total transaction limit set for the terminal has been crossed
1018	INVALID-DEBIT-AMOUNT-LIMIT	Maximum debit amount limit set for the terminal for a day has been crossed
1019	INVALID-CREDIT-AMOUNT-LIMIT	Maximum credit amount limit set for the terminal for a day has been crossed
1020	MAXIMUM-DEBIT-AMOUNT-CROSS	Maximum debit amount set for per card for rolling 24 hrs has been crossed
1021	MAXIMUM-CREDIT-AMOUNT-CROSS	Maximum credit amount set for per card for rolling 24 hrs has been crossed
1022	MAXIMUM-TRANSACTION-CROSS	Maximum transaction set for per card for rolling 24 hrs has been crossed
1023	HASH-MISMATCH	Hash Mismatch
1024	INVALID-PARAMS	Invalid parameters
1025	INVALID-BANK-CODE	Invalid bank code
1026	INVALID-MERCHANT	Merchant is not active
1027	INVALID-TRANSACTION	Invalid transaction
1028	TRANSACTION-NOT-FOUND	Transaction not found
1029	TRANSACTION-TERMINATED	Transaction terminated
1030	TRANSACTION-INCOMPLETE	Transaction incomplete
1031	AUTO-REFUNDED	Transaction auto refunded
1032	REFUNDED	Transaction refunded
1033	SINGLE-TRANSACTION-LOWER-LIMIT-CROSS	The amount provided is less than transaction lower limit
1034	SINGLE-TRANSACTION-UPPER-LIMIT-CROSS	The amount provided is more than transaction upper limit
1035	TRANSACTION-DAILY-LIMIT-	The daily transaction limit is exceeded for the

Code	Response Message	Description
	CROSS	merchant
1036	TRANSACTION-MONTHLY-LIMIT-CROSS	The monthly transaction limit is exceeded for the merchant
1037	DAILY-TRANSACTION-NUMBER-CROSS	The daily transaction number is exceeded for the merchant
1038	MONTHLY-TRANSACTION-NUMBER-CROSS	The monthly transaction number is exceeded for the merchant
1039	INVALID-REFUND-AMOUNT	The refund amount is greater than transaction amount
1040	INVALID-CVV	Invalid Card Verification Code
1041	AUTO-REFUNDED-TNP	Transaction is auto refunded by TnP
1042	FAILED-NO-RESPONSE	Transaction failed as there was no response from bank
1043	TRANSACTION-CANCELLED	Transaction cancelled
1044	UNAUTHORIZED	Unauthorized
1045	FORBIDDEN	Forbidden Access
1046	TRANSACTION-ALREADY-CAPTURED	Transaction already captured
1047	AUTHORIZED	Transaction authorized
1048	CAPTURED	Transaction captured
1049	VOIDED	Transaction voided
1050	NO-RECORD-FOUND	No data record found for the given input
1051	ACQUIRER-ERROR	Error occurred at the bank end
1052	INVALID-EMAIL	Invalid Email ID
1053	INVALID-PHONE	Invalid phone number
9999	UNKNOWN-ERROR	Unknown error occurred
997	-	These are unhandled errors coming from banks directly.