Network›

Merchant Integration Guide — Bitmap

Version 1.0

July 2016

Issued by: TimesofMoney Ltd, Reliable Tech Park, Airoli, Mumbai, India

Contents

Copyright 5
1.0 Introduction 6
1.1 About this Document 6
1.11 Audience 6
1.12 Your Takeaways from this Document 6
1.2 System Functionality 7
2.0 Overview of Merchant Integration 8
2.1 Test Requirements 8
2.2 Integration and Testing 8
2.3 Technical Requirements 9
2.4 Requirements to Go 'Live' 9
3.0 Message Structure and its Interpretation 10
3.1 Block Existence Indicator (BEI) 11
3.2 Field Existence Indicator (FEI) 12
4.0 Transaction API 17
4.1 Overview 17
4.2 Posting URLs 17
4.3 Special Characters Allowed 17
4.4 Handling of Case Sensitivity 18
4.5 Table Structure and Transaction Request Fields 18
4.6 Transaction Response Fields 31
5.0 Formats & Examples 36
5.1 Formats & Examples for 'Transaction Requests' 36
5.11 'Transaction' Data Block Creation Method and Examples for 'Transaction Requests' 36
5.12 'Billing' Data Block Creation Method and Examples for 'Transaction Requests' 37

5.13 'Shipping' Data Block Creation Method and Examples for 'Transaction Requests' 37
5.14 'Payment' Data Block Creation Method and Examples for 'Transaction Requests' 38
5.15 'Merchant' Data Block Creation Method and Examples for 'Transaction Requests' 38
5.16 'Other' Data Block Creation Method and Examples for 'Transaction Requests' 39
5.17 'DCC' Data Block Creation Method and Examples for 'Transaction Requests' 39
5.18 Format of Entire 'Transaction Request' 40
5.19 Samples of Entire 'Transaction Request' Formation 41
5.110 Sample of Encrypted Request 42
5.2 Formats & Examples for 'Transaction Responses' 43
5.21 'Transaction' Data Block Creation Method and Examples for 'Transaction Responses' 43
5.22 'Transaction Response' Data Block Creation Method and Examples for 'Transaction Responses' 44
5.23 'Transaction Status' Data Block Creation Method and Examples for 'Transaction Responses' 44
5.24 'Merchant' Data Block Creation Method and Examples for 'Transaction Responses' 45
5.25 'Fraud' Data Block Creation Method and Examples for 'Transaction Responses' 45
5.26 'DCC' Data Block Creation Method and Examples for 'Transaction Responses' 45
5.27 'Additional' Data Block Creation Method and Examples for 'Transaction Responses' 46
5.28 Format of Entire 'Transaction Response' 47
5.29 Samples of the Entire 'Transaction Response' Formation, which you will Receive 48
5.210 Sample of Encrypted Response 48
5.3 Enabling Server-to-Server Transaction Push Responses 49
5.31 Sample of Server-to-Server Transaction Push Responses 49
6.0 Encryption Logic 51
6.1 Test Merchant Encryption and Decryption Key 51
6.2 Production Merchant Encryption and Decryption Key 51
6.3 Encryption / Decryption Method 51
6.4 How to Use 'Encryption' or 'Decryption' 52
7.0 Error Codes 53

7.1 PSP Error Codes 53
7.2 Payment Gateway Error Codes 57

Copyright

Published May, 2016

1.0 Introduction

1.1 About this Document

The ***'Merchant Integration Guide'*** provides guidance on the process of merchant integration using the ***Bitmap*** methodology. It also focuses on specific areas of integration such as transactions.

1.11 Audience

This document is intended for:

- Software Developers from the Merchant's side, who will be carrying out 'Merchant Integration'.
- Any other authorized person who is interested in, or needs to understand the Merchant Integration process.

1.12 Your Takeaways from this Document

On completion of this document, you will learn:

1. How to carry out 'Merchant Integration' in general
2. About the pre-requisites for integration
3. About the 'request' and 'response' tables required to implement integration
4. How to form 'transaction requests' and 'transaction responses'
5. How to encrypt transactions
6. About the formats and examples relevant to the integration
7. About the Error Codes relevant to integration in general

1.2 System Functionality

The ***Payment Service Provider (PSP)*** system supports all forms of payment methods for the global e-commerce business. It supports fraud monitoring, including the setting of fraud rules as well as the management of queues of suspicious transactions.

The communication between the merchant system and the PSP system is encrypted, providing security. The exchange of *requests* and *responses* occurs from the merchant's Server to the PSP Server, thus reducing the risks related to data transmission.

Figure 1 describes the standard transaction flow.

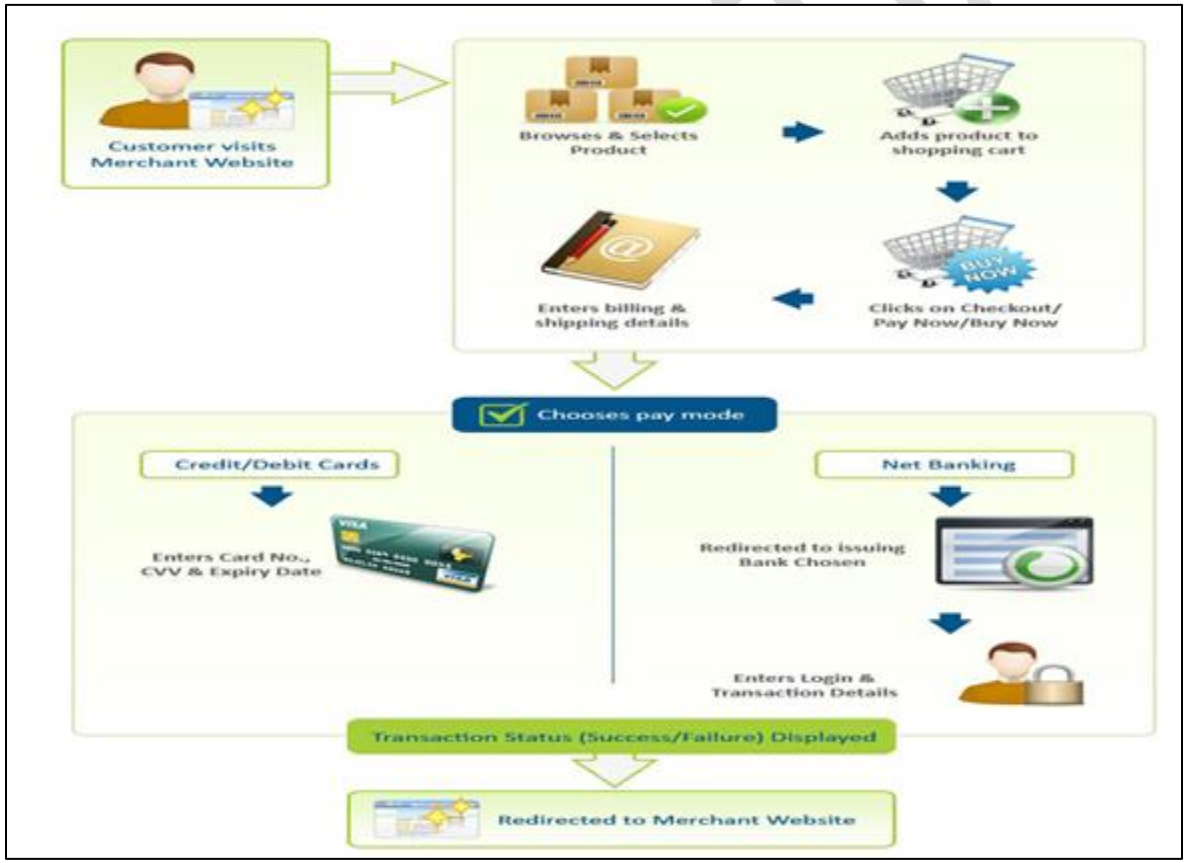

Figure 1: Standard Transaction Flow

2.0 Overview of Merchant Integration

In this integration, as a *merchant*, or a *software developer* from the merchant's side, you need to capture details of your customers on your website. You need to create your own payment page. You also need to comply with ***PCI/DSS Regulations*** for capturing credit card details in your website. You can perform transactions using this 'Merchant Integration Guide' and send the transaction details to the payment gateway.

Two options are available for integration:

1. ***PSP-Hosted Integration:*** This is used by majority of merchants. As a merchant, you can allow your customers to enter their sensitive card data on a page that is hosted on the PSP, whereby all the sensitive card information is managed by the PSP. Therefore, you do not have to comply with PCI/DSS regulations. There is no compromise on the available functionality due to this type of integration.

2. ***Merchant-Hosted Integration:*** This is used by merchants who wish to capture the card details of their customers on their merchant site. In this integration method, you need to use the 'Merchant Integration Guide' APIs but create your own payment page. If you capture credit card details, you need to comply with PCI/DSS regulations.

2.1 Test Requirements

To pass a test transaction, you need the following:

- A test merchant ID
- A merchant specific encryption key
- A posting URL for transaction

2.2 Integration and Testing

To set up a test account and begin integration, perform the following steps:

1) Create a form on your website that posts a payment request to the PSP.
   You will be redirected to the PSP secure payment page if your integration is 'PSP hosted'.
2) Enter the transaction details and click **Pay Now**.
   Once you submit the card details (either on the PSP hosted page, or your website, depending on your chosen integration option), transaction processing will proceed further.

2.3 Technical Requirements

1. Ability of the Merchant to create web pages and forms and assimilate information from the customers for payment services including fraud screening.
2. Ability of the Merchant to ensure that they process the information in the appropriate way and convey it to the customers.
3. A shopping-cart software or a website that supports shopping
4. A secure SSL order form
5. Product pages in one of the scripting languages
6. Applicable programming languages: JSP, VB, and .NET (C#), PHP

2.4 Requirements to Go 'Live'

You require a contract with the PSP service and optionally, for DCC and fraud and risk management. Please note that the lead times can take several weeks to set up live accounts, so we recommend parallel process development and testing with contracting and onboarding.

3.0 Message Structure and its Interpretation

As defined in Table 1, a ***message*** will have the following structure:

<Merchant ID><Collaborator ID><Encrypted String>

The ***Encrypted String*** has the following *broad-level* structure:

<Block Existence Indicator>||<Data Block 1>||<Data Block 2>||<Data Block 3>

Each ***Data Block*** has the following structure:

<Field Existence Indicator for the Block>|<Field data 1 for the Block>|<Field data 2 for the Block>|<Field data 3 for the Block…|<Field data n for the Block>

Thus, the ***overall structure*** of the *Encrypted String* is as follows:

<Block Existence Indicator>||<Field Existence Indicator for Block 1>|<Field data 1 for Block 1>|<Field data 2 for Block 1..>||<Field Existence Indicator for Block 2>|<Field data 1 for Block 2>|<Field data 2 for Block 2>|<Field data 3 for Block 2..>||<Field Existence Indicator for Block n>|<Field data 1 for Block n>|<Field data 2 for Block n>

Note 1: Each Data Block is separated by a ***double pipe*** (shown in red) and each field is separated by a ***single pipe*** (shown in red).

The structure of the ***Encrypted String*** is explained in the subsequent sections.

3.1 Block Existence Indicator (BEI)

The '***Block Existence Indicator***' indicates to the ***Message Parser*** whether a particular block of data is present in the given ***Input*** message, or ***Response*** message. As defined in the tables, each message will have the fields grouped into ***Data Blocks***. The existence, or absence of the block in a particular message will be indicated by a ***1*** or a ***0*** in the position for the block, where:

- **1 = presence of Data Block**
- **0 = absence of Data Block**

Example:

In case of ***Input/Request*** messages, a 'BEI' value of ***'1001000'*** implies the following:

1) *Transaction* Data Block is present in the request
2) *Billing* Data Block is ***not*** present in the request
3) *Shipping* Data Block is ***not*** present in the request
4) *Payment* Data Block is present in the request
5) *Merchant* Data Block is ***not*** present in the request
6) *Other Details* Data Block is ***not*** present in the request
7) *DCC* Data Block is ***not*** present in the request

Note 2:

1. ***Merchant ID*** and ***Collaborator ID*** are mandatory for all messages in the request. Valid values are as defined in Tables 1 and 2.
2. ***Transaction*** Data Block and ***Payment*** Data Block are mandatory blocks for 'Merchant-hosted' integration, where the payment details are captured at the merchants' page, and therefore any encrypted string must, at least, have the Block Existence Indicator (BEI) as ***'1001000'***.
3. For ***'PSP hosted'*** integration, the payment details are captured at the PSP's payment page, and therefore any encrypted string must mandatorily have, at least, a BEI value of ***'1000000'***.
4. The indicator for the block for which you are sending data, must be set to ***'0'***, if you choose to skip the block. For example, the ***Merchant*** Data Block is optional, and can be used by the merchants to send 'pass through' data, or any other relevant data they may need for their internal purposes, and therefore the block indicator must be set to ***'1'*** if this data is to be parsed.
5. The 'Field Existence Indicator' will be present in the request only if a Data Block is present.
6. 'BEI' is also referred to as ***'Packet FieldBitmap'***, or ***'Request Packet FieldBitmap'***, or ***'Response Packet FieldBitmap'***

3.2 Field Existence Indicator (FEI)

The ***Field Existence Indicator*** indicates to the ***Message Parser*** whether a particular ***field*** of a Data Block is present, or absent in the given ***Input*** message, or ***Response*** message, as defined in Tables 1 and 2. Each message will have the fields grouped into Data Blocks. The existence, or absence of the field in the Data Block in a particular message will be indicated by a ***1*** or a ***0*** in the position of the field for the Data Block, where:

- **1 = presence of a field in the Data Block**
- **0 = absence of a field in the Data Block**

Example:

In case of ***Input/Request*** messages, an FEI value of **'11111111'** of the ***Transaction*** Data Block implies the following:

1. *Merchant Order Number* field of the *Transaction* Data Block is present in the request
2. *Amount* field of the *Transaction* Data Block is present in the request
3. *Success URL* field of the *Transaction* Data Block is present in the request
4. *Failure URL* field of the *Transaction* Data Block is present in the request
5. *Transaction Mode* field of the *Transaction* Data Block is present in the request
6. *PayMode Type field* of the *Transaction* Data Block is present in the request
7. *Transaction Type* field of the *Transaction* Data Block is present in the request
8. *Currency* field of the *Transaction* Data Block is present in the request

And, an FEI value of ***'0000000000000'*** of the ***Billing*** Data Block implies the following:

1. *BillToFirstName* field of the *Billing* Data block is ***not*** present in the request
2. *BillToLastName* field of the *Billing* Data block is ***not*** present in the request
3. *BillToStreet1* field of the *Billing* Data block is ***not*** present in the request
4. *BillToStreet2* field of the *Billing* Data block is ***not*** present in the request
5. *BillToCity* field of the *Billing* Data block is ***not*** present in the request
6. *BillToState* field of the *Billing* Data Block is ***not*** present in the request
7. *BillToPostalCode* of the *Billing* Data Block is ***not*** present in the request
8. *BillToCountry* of the *Billing* Data Block is ***not*** present in the request
9. *BillToEmailID* of the *Billing* Data Block is ***not*** present in the request
10. *BillToMobileNumber* of the *Billing* Data Block is ***not*** present in the request
11. *BillToPhoneNumber1* of the *Billing* Data Block is ***not*** present in the request
12. *BillToPhoneNumber2* of the *Billing* Data Block is ***not*** present in the request

13. *BillToPhoneNumber3* of the *Billing* Data Block is ***not*** present in the request

An FEI value of ***'111011010001'*** of the ***Shipping*** Data block implies the following:

1. *ShipToFirstName* field of the *Shipping* Data block is present in the request
2. *ShipToLastName* field of the *Shipping* Data block is present in the request
3. *ShipToStreet1* field of the *Shipping* Data block is present in the request
4. *ShipToStreet2* field of the *Shipping* Data block is ***not*** present in the request
5. *ShipToCity* field of the *Shipping* Data block is present in the request
6. *ShipToState* field of the *Shipping* Data Block is present in the request
7. *ShipToPostalCode* of the *Shipping* Data Block is ***not*** present in the request
8. *ShipToCountry* of the *Shipping* Data Block is present in the request
9. *ShipToPhoneNumber1* of the *Shipping* Data Block is ***not*** present in the request
10. *ShipToPhoneNumber2* of the *Shipping* Data Block is ***not*** present in the request
11. *ShipToPhoneNumber3* of the *Shipping* Data Block is ***not*** present in the request
12. *ShipToMobileNumber* of the *Shipping* Data Block is present in the request

In case of ***Credit/Debit Card transaction***, an FEI value of ***'11111100000'*** of the ***Payment*** Data block implies the following:

1. *Card Number* field of the *Payment* Data Block is present in the request
2. *Expiry Month* field of the *Payment* Data Block is present in the request
3. *Expiry Year* field of the *Payment* Data Block is present in the request
4. *CVV* field of the *Payment* Data Block is present in the request
5. *Card Holder Name* field of the *Payment* Data Block is present in the request
6. *Card Type* field of the *Payment* Data Block is present in the request
7. *Customer Mobile Number* field of the *Payment* Data Block is ***not*** present in the request
8. *Payment ID* field of the *Payment* Data Block is ***not*** present in the request
9. *OTP* field of the *Payment* Data Block is ***not*** present in the request
10. *Gateway ID* field of the *Payment* Data Block is ***not*** present in the request
11. *Card Token* field of the *Payment* Data Block is ***not*** present in the request

In case of ***Net Banking transaction***, an FEI value of ***'00000000010'*** of the ***Payment*** Data Block implies the following:

1. *Card Number* field of the Payment Data Block is ***not*** present in the request
2. *Expiry Month* field of the Payment Data Block is ***not*** present in the request
3. *Expiry Year* field of the Payment Data Block is ***not*** present in the request
4. *CVV* field of the Payment Data Block is ***not*** present in the request
5. *Card Holder Name* field of the Payment Data Block is ***not*** present in the request
6. *Card Type* field of the Payment Data Block is ***not*** present in the request

7. *Customer Mobile Number* field of the Payment Data Block is ***not*** present in the request
8. *Payment ID* field of the Payment Data Block is ***not*** present in the request
9. *OTP* field of the Payment Data Block is ***not*** present in the request
10. *Gateway ID* field of the Payment Data Block is present in the request
11. *Card Token* field of the Payment Data Block is ***not*** present in the request

An FEI value of **'1000000000'** of the ***Merchant*** Data Block implies the following:

1. *UDF1* field of the *Merchant* Data Block is present in the request
2. *UDF2* field of the *Merchant* Data Block is ***not*** present in the request
3. *UDF3* field of the *Merchant* Data Block is ***not*** present in the request
4. *UDF4* field of the *Merchant* Data Block is ***not*** present in the request
5. *UDF5* field of the *Merchant* Data Block is ***not*** present in the request
6. *UDF6* field of the *Merchant* Data Block is ***not*** present in the request
7. *UDF7* field of the *Merchant* Data Block is ***not*** present in the request
8. *UDF8* field of the *Merchant* Data Block is ***not*** present in the request
9. *UDF9* field of the *Merchant* Data Block is ***not*** present in the request
10. *UDF10* field of the *Merchant* Data Block is ***not*** present in the request

An FEI value of **'1111111'** of the ***Other Details*** Data Block implies the following:

1. *Cust ID* field of the *Other Details* Data Block is present in the request
2. *Transaction Source* field of the *Other Details* Data Block is present in the request
3. *Product Info* field of the *Other Details* Data Block is present in the request
4. *Is User Logged In* field of the *Other Details* Data Block is present in the request
5. *Item Total* field of the *Other Details* Data Block is present in the request
6. *Item Category* field of the *Other Details* Data Block is present in the request
7. *Ignore Validation Result* field of the *Other Details* Data Block is present in the request

An FEI value of **'111'** of the ***DCC*** Data Block implies the following:

1. *DCC Reference Number* field of the *DCC* Data Block is present in the request
2. *Foreign Amount* field of the *DCC* Data Block is present in the request
3. *Foreign Currency* field of the *DCC* Data Block is present in the request

In case of the Transaction ***Response*** Message, the message structure would be similar to the Transaction ***Request*** Message, where a BEI and an FEI would be present.

<MerchantID><Encrypted String>

Note 3:

1. 'Collaborator ID' is ***not*** a part of the Transaction ***Response*** message.
2. 'FEI' is also referred to as ***'FieldBitmap'***, or ***'Block FieldBitmap'***

The ***Decrypted String*** has the following structure (similar to that of the ***Encrypted*** String):

<Block Existence Indicator>||<Field Existence Indicator for Block 1>|<Field data 1 for Block 1>|<Field data 2 for Block 1..>||<Field Existence Indicator for Block 2>|<Field data 1 for Block 2>|<Field data 2 for Block 2>|<Field data 3 for Block 2..>||<Field Existence Indicator for Block n>|<Field data 1 for Block n>|<Field data 2 for Block n>

An FEI value of '**111111**' of the ***Transaction Response*** Block implies the following:

1. *Merchant Order Number* field of *Transaction Response* block is present in the response
2. *Currency* field of the *Transaction Response* block is present in the response
3. *Amount* field of the *Transaction Response* block is present in the response
4. *PayMode* field of the *Transaction Response* block is present in the response
5. *Card Type* field of the *Transaction Response* block is present in the response
6. *Transaction Type* of the *Transaction Response* block is present in the response

An FEI value of '**1100010**' of the ***Transaction Response Data*** Block implies the following:

1. *Reference Number* field of the *Transaction Response Data* Block is present in the response
2. *Transaction Date* field of the *Transaction Response Data* block is present in the response
3. *Card Enrollment Response* field of the *Transaction Response Data* block is ***not*** present in the response
4. *ECI Indicator* field of the *Transaction Response Data* Block is ***not*** present in the response
5. *Gateway Trace Number* field of the *Transaction Response Data* block is ***not*** present in the response
6. *Gateway Identifier* field of the *Transaction Response Data* Block is present in the response
7. *Auth Code* field of the *Transaction Response Data* Block is ***not*** present in the response

An FEI value of '**111**' of the ***Transaction Response Status*** Data Block implies the following:

1. *Status Flag* field of the *Transaction Response Status* Data Block is present in the response
2. *Error Code* field of the *Transaction Response Status* Block is present in the response
3. *Error Message* field of the *Transaction Response Status* Data Block is present in the response

An FEI value of **'0000000001'** of the ***Merchant*** Data Block implies the following:

1. *UDF1* field of the *Merchant* Data Block is ***not*** present in the response
2. *UDF2* field of the *Merchant* Data Block is ***not*** present in the response
3. *UDF3* field of the *Merchant* Data Block is ***not*** present in the response
4. *UDF4* field of the *Merchant* Data Block is ***not*** present in the response
5. *UDF5* field of the *Merchant* Data Block is ***not*** present in the response
6. *UDF6* field of the *Merchant* Data Block is ***not*** present in the response
7. *UDF7* field of the *Merchant* Data Block is ***not*** present in the response
8. *UDF8* field of the *Merchant* Data Block is ***not*** present in the response
9. *UDF9* field of the *Merchant* Data Block is ***not*** present in the response
10. *UDF10* field of the *Merchant* Data Block is present in the response

An FEI value of **'11'** of the ***Fraud*** Data Block implies the following:

1. *Fraud Decision* field of the *Fraud* Data Block is present in the response
2. *Fraud Reason* field of the *Fraud* Data Block is present in the response

An FEI value of **'11111'** of the ***DCC*** Data Block implies the following:

1. *DCC Converted* field of *DCC Data* block is present in the response
2. *DCC Converted Amount* field of the *DCC Data* block is present in the response
3. *DCC Currency* field of the *DCC Data* block is present in the response
4. *DCC Margin* field of the *DCC Data* block is present in the response
5. *DCC Exchange Rate* field of the *DCC Data* block is present in the response

An FEI value of **'11'** of the ***Additional Info*** Data Block implies the following:

1. *Card Token* field of the *Additional Info* Data Block is present in the response
2. *Card Number (Last Four Digits)* field of the *Additional Info* Data Block is present in the response

4.0 Transaction API

4.1 Overview

The following details will be provided:

1. **Test MID (Merchant ID):** this will be provided at the time of integration
2. **Test Encryption Key:** this will be provided at the time of integration
3. **Live MID (Merchant ID):** this will be provided once the integration is complete and the MID has been set up
4. **Live Encryption Key:** this will be provided once the integration is complete and the MID has been set up

4.2 Posting URLs

The URLs for testing and live transactions are as follows:

- TEST URL: https://uat.timesofmoney.com/direcpay/secure/PaymentTxnServlet
- LIVE URL: https://www.timesofmoney.com/direcpay/secure/PaymentTxnServlet

4.3 Special Characters Allowed

1. _ (underscore)
2. **.** (dot)
3. @ (at the rate)
4. **/** (forward slash)
5. (space)
6. **,** (comma)
7. - (hyphen)
8. **:** (colon)
9. **‘** (apostrophe)

4.4 Handling of Case Sensitivity

In general, for any input field, if a value has to be selected from a given list in the document, the values can be given in ***uppercase***, ***camel case***, ***mixed case***, or ***lower case*** format. These are converted into uppercase and stored in the system to maintain consistency while reporting.

4.5 Table Structure and Transaction Request Fields

Table 1 lists all the ***Transaction Request*** fields and the fields in this table are explained as followed:

- **Block Number:** This is the position where the *Data Block* needs to be present in the request. An ***'R'*** against a Block name means it is a ***mandatory*** field and must be passed in the Transaction Request. An ***'O'*** against a Block name means it is an ***optional*** field and you can decide if you want to pass this block as part of the request or not.

- **Field Number:** This is the position of the *field* for that particular Data Block.

The 'Transaction Request' fields can be of three types, listed as follows:

1. **Required (R)**: These fields *must* be passed to the API, or entered on the *merchant's* web page.
2. **Conditional (C)**: These fields may be required depending on the values in other fields, the specifics of which are defined in the comments column.
3. **Optional (O)**: These fields are not mandatory, but may help in reporting, analysis, and for future integrations with decision management systems, fraud detection, etc. The merchant can decide not to pass the block as part of the request.

Note 4:

1. We recommend including all of these categories of fields, if the respective information is available.
2. The merchant website URL (referred to as Merchant Transaction Referrer URL in the *Error Codes* table) from where the transaction will be initiated needs to be *whitelisted* at our end. If the URL is not whitelisted, the transaction will not go through. This whitelisting needs to be done both during 'testing' and during 'live'

Table 1: Transaction Request Fields

Transaction Request Fields									
S.N	**Block No.**	**Block Name**	**Block R/O/C**	**Block Desc**	**Field No**	**Field Name**	**Description**	**Field R/O/C**	**Field Length**
1	0		R	This needs to be passed as part of the request in clear text	1	Merchant ID	Unique merchant ID created for every merchant at the time of registration. This is mandatory for every transaction. Any character other than the Merchant ID will not be accepted. Ensure that there is no space in between, before, or after the field.	R	Numeric (15)
					2	Collaborator ID	'NI' on staging and production.	R	String (8)
2	1	Transaction Data	R	All the transaction details are passed in this block	1	Merchant Order Number	Your 'order number' for each transaction. Special characters that are defined in the document are allowed. For example, asdf123, or 1234, or ABCD-1234.	R	String (100)
					2	Amount	Price for the item. Values up to 2 decimal places are allowed. For example: 1000.02	R	Decimal (14,2)
					3	Success URL	Merchant-side URL, to which the customer will be redirected if the transaction is successful.	R	String (200)
					4	Failure URL	Merchant-side URL, to which the customer will be redirected if the transaction is a failure.	R	String (200)
					5	Transaction Mode	Type of Transaction (*Internet*, *Moto*, or *Recurring*). Default Value: ***Internet*** If the facility of *'Skip Authentication'* is offered via browser re-direct then MOTO	R	String (20)

							needs to be sent in the request. ***Note:*** To set up *'Recurring'* for a transaction, merchant needs to send the transaction mode in the request as 'Recurring'. If the transaction mode is not sent as 'Recurring' then the transaction will be processed as normal transaction. Please contact the ***'Merchant Support Helpdesk'*** to enable 'Recurring'.		
					6	PayMode Type	(CC)-Credit Card, (DC)-Debit Card, (DD)-Direct Debit, (PAYPAL)-PayPal, (NB)-Net Banking ***Note:*** *This field is required for 'Merchant Hosted' type of integration.*	C	String (10)
					7	Transaction Type	01 for SALE 02 for AUTHORIZATION	R	String (2)
					8	Currency	Used to specify the *currency* for the transaction. Currently, only ***AED*** is accepted for all transactions (as per ISO Currency Code).	R	String (3)
3	2	Billing Data	O	Billing information of the customer	1	BillToFirstName	Card Holder's first name as printed on the card. Special characters (defined earlier in this document) are allowed. ***Important:*** *This string will accept either alphabets, or a combination of alphabets and special characters (defined earlier in this document).*	O	String (50)

					2	BillToLastName	Card Holder's last name as printed on the card. Special characters (defined earlier in this document) are allowed. ***Important:*** *This string will accept either alphabets, or a combination of alphabets and special characters (defined earlier in this document)*.	O	String (50)
					3	BillToStreet1	Customer address. Special characters (defined earlier in the document) are allowed.	O	String (40)
					4	BillToStreet2	Additional information about the address. Special characters (defined earlier in the document) are allowed.	O	String (40)
					5	BillToCity	City of the billing address. This is ***not*** a numeric field and special characters are ***not*** allowed ***except*** 'Space'.	O	String (50)
					6	BillToState	State or province of the billing address. Use the State, Province, and Territory Codes for the United States and Canada. This is ***not*** a numeric field and special characters are ***not*** allowed ***except*** 'Space'.	O	String (50)
					7	BilltoPostalCode	Postal code for the billing address. This is an *Alphanumeric* field. Special characters ***not*** allowed ***except*** space. For example, abc123, or abc 12, or 1234	O	String (10)
					8	BillToCountry	Country of the billing address. Use the two-character ISO Standard Country Codes.	O	String (2)

					9	BillToEmail	Customer email address, including the full domain name. For example, abc@gmail.com	O	String (100)
					10	BillToMobileNumber	Mobile number of the customer. If the customer enters the mobile number at your end please send the value to this particular field. Possible accepted values: +91-99999-9999 or 9999999999 Mobile number must ***not*** have a ***space***.	O	String (15)
					11	BillToPhoneNumber1	Country code of the customer's *landline phone*. If the customer does not enter the country code then set the ***default value*** as ***'99',*** or send this field as blank, because it is optional.		String (15)
					12	BillToPhoneNumber2	Area code of the customer's *landline phone*. If the customer does not enter the area code then set the ***default value*** as ***'99'***, or send this field as blank, because it is optional.		String (15)
					13	BillToPhoneNumber3	Landline number of the customer. If the customer does not enter the landline number then set the ***default value*** as ***'999999',*** or send this field as blank, because it is optional.	O	String (15)
4	3	Shipping Data	O	Shipping information of the customer	1	ShipToFirstName	*First name* of the *receiver* to whom the product will be shipped. Special characters (defined earlier in the document) are allowed. ***Important:*** This string will accept either *alphabets*, or a combination of *alphabets* and	O	String (50)

	special characters, as defined earlier in the document. For merchants opting for "Merchant Hosted" Integration, this field is not validated		
2	ShipToLastName	*Last name* of the *receiver* to whom the product will be shipped. Special characters (defined earlier in the document) are allowed. ***Important:*** This string will accept either alphabets, or a combination of alphabets and special characters as defined in the document. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (50)
3	ShipToStreet1	First line of the street address where the product needs to be delivered to. Special characters (defined earlier in the document) are allowed. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (40)
4	ShipToStreet2	Additional information about the address, for shipping the product. Special characters (defined earlier in the document) are allowed. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (40)
5	ShipToCity	City to which the product needs to be shipped. This is ***not*** a numeric field and special	O	String (50)

		characters are ***not*** allowed ***except*** space. For merchants opting for "Merchant Hosted" Integration, this field is not validated.		
6	ShipToState	*State, Province, and Territory Codes* of the 'Shipping Address'. This is ***not*** a numeric field and special characters are ***not*** allowed ***except*** space. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (50)
7	ShiptoPostalCode	*Postal code* for the shipping address. The field is an *Alphanumeric* field. Special characters are ***not*** allowed ***except*** space. For example, abc123, or abc 12, or 1234. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (10)
8	ShipToCountry	Country to which the product needs to be shipped. Use the two-character ISO Standard Country Codes. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (2)
9	ShipToPhoneNumber1	*Country code* of the receivers' landline phone number. If the customer does not enter the country code then set the default value as ***'99'***, or send this field as blank, because it is optional. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (15)

					10	ShipToPhoneNumber2	*Area code* of the receiver's landline phone number. If the customer does not enter the area code then set the default value as ***'99'***, or send this field as blank, because it is optional. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (15)
					11	ShipToPhoneNumber3	*Phone number* of the receiver's landline phone. If the customer does not enter the landline number then set the default value as ***'999999'***, or send this field as blank, because it is optional. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (15)
					12	ShipToMobileNumber	Mobile number of the receiver's mobile phone. If the customer enters the mobile number at your end please send the value in this particular field. Accepted values: +971-99999-9999 or 9999999999 Mobile number must ***not*** be sent with a ***space***. For merchants opting for "Merchant Hosted" Integration, this field is not validated.	O	String (15)
					1	Card Number	Customer Credit, or Debit card number. Minimum Length: 16 digits Max Length: 19 digits	C	Numeric (19)

5	4	Payment Data	R	Card information of the customer	2	Exp Month	The month when the card expires *(two digits)*. Format: ***MM*** (for example, 01 for Jan and 03 for March)	C	String (2)
					3	Exp Year	The year when the card expires *(four digits)*. Format: ***YYYY*** (for example, 2014, or 2016). should be >= current year	C	String (4)
					4	CVV	Card Verification Value (CVV) • *3 digits* - for all *Visa* and *MasterCard* cards • *4 digits* - for AMEX cards	C	String (4)
					5	Card Holder Name	Name of the Card Holder, as printed on the card	C	String (50)
					6	Card Type	Type of Card Accepted values: MASTERCARD, VISA, DINERS, AMEX, JCB	C	String (20)
					7	CustMobileNumber	Mobile number of the customer that is registered with his/her *Issuing Bank*	C	Numeric (10)
					8	Payment ID	The MMID that the customer has received from his/her *Issuing Bank*	C	String (15)
					9	OTP	One-time password received from the customer's *Issuing Bank*	C	Numeric (10)
					10	Gateway ID	The ID that identifies the gateway assigned to the merchant. This ID will define which bank the transaction needs to be routed to.	C	Numeric (5)
					11	Card Token	This is a unique value generated for every card that is saved in the system, which is returned in the response.	C	String (50)

							Merchants can use this value in case of *Tokenization/Stored card* feature.		
6	5	Merchant Transaction Information	O	Additional Information related to the transaction	1	UDF1	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					2	UDF2	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					3	UDF3	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					4	UDF4	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					5	UDF5	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					6	UDF6	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					7	UDF7	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					8	UDF8	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
					9	UDF9	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)

					10	UDF10	This is a 'user-defined field' that can be used to send additional information about the transaction.	O	String (250)
7	6	Other Details	O	Additional Information related to the transaction	1	Cust ID	The ID used to identify your customer when their profile was created. This field has to be passed by you only if you have opted for the *'Tokenization/Stored Card'* feature. The field is *'alphanumeric';* special characters are ***not*** allowed. ***Note:*** For *'Recurring'* transactions, the merchant needs to send the customer ID in the request for that specific customer. If the customer ID is not passed in the request, the transaction will not be set up as 'recurring transaction' and will be processed as 'normal transaction'.	O	String (50)
					2	Transaction Source	This is used to identify a channel, in case you are using multiple channels. Acceptable values are: *Web, IVR,* or *Mobile.*	O	String (20)
					3	Product Info	This field is used to send details about the product and related information.	O	String (100)
					4	Is User Logged In	Indicates whether a customer has signed in to your website, or has done a *Guest* checkout.	O	String (1)

							Values: ***'Y'*** – customer has signed in ***'N'*** – customer has done a Guest checkout		
					5	Item Total	This is a comma-separated list of the sale value of all the items in the order. Example: If the order has two items of value 500 and 1000, send the value as '500.00, 1000.00'. Up to 2 decimal places are allowed in the value.	O	String (100)
					6	Item Category	This is a comma-separated list of the categories of all the items in the order. Example: If the order is two books and one CD, send "Book, Book, CD". This field is used to determine product category: electronic, handling, physical, service, or shipping.	O	String (255)
					7	Ignore Validation Result	Indicates whether you want to proceed for Authorization irrespective of the 3DSecure/Authentication result, or not. 'TRUE' indicates you want to; 'FALSE' indicates you do not. ***Default value: FALSE***	O	String (10)
8	7	DCC Block	O	DCC required information	1	DCC Reference Number	This will be the value that is sent in the DCC response field. If the customer card is not eligible for DCC, then do not send this value in the transaction request but send this value as blank.	O	String (100)
					2	Foreign Amount	Amount that was received in the lookup response for that	O	Decimal (14,2)

							particular card. If the card is not eligible for DCC you will not receive this value, which means that it is not an international card and in that case, you will send this value as blank.		
					3	Foreign Currency	Card Currency that you have received in the DCC Lookup response. If you do not receive this value, or if the card is not eligible for DCC you will not receive this value, which means that it is not an international card and in that case, you will need to send this value as blank.	O	String (3)

4.6 Transaction Response Fields

The following table lists all the transaction response fields:

Table 2: Transaction Response Fields

Response Fields									
S.N.	**Block No.**	**Block Name**	**Block R/O/C**	**Block Desc**	**Field No.**	**Field Name**	**Description**	**Field R/O/C**	**Field Length**
1	0		R	This needs to be passed as part of the request in clear text	1	Merchant ID	Same value that is sent in the API Request. This will be sent back in plain text.	R	Numeric (15)
2	1	Transaction Response	R	Same value that is sent in the request	1	Merchant Order Number	Same value that is sent in the API Request	R	String (100)
					2	Currency	Same value that is sent in the API Request	R	String (3)
					3	Amount	Same value that is sent in the API Request	R	Decimal (14,2)
					4	PayMode	(CC)-Credit Card, (DC)-Debit Card, (DD)-Direct Debit, (PAYPAL)-PayPal, (NB)-Net Banking ***Note:*** If the value sent in the request is incorrect then the actual value will be sent back in the response.	R	String (10)
					5	Card Type	Same value that is sent in the API Request	R	String (20)
					6	Transaction Type	Same value that is sent in the API Request (01 for Sale, 02 for Authorization).	R	String (2)

3	2	Transaction Response	C	Transaction related information	1	Reference Number	Unique Reference ID that is used to identify a transaction and is generated at the time of transaction creation. This will be generated only when all the required fields are passed in the manner as described in the description section of Request Table (Table 1). This will not be generated in case of validation failures.	C	Numeric (16)
					2	Transaction Date	Date and time at which the transaction took place Format - DMMYY:HRMINSEC	C	String (35)
					3	Card Enrollment Response	Status of the payment. It can have the following values: *Enrolled* and *NotEnrolled*	C	String (20)
					4	ECI Indicator	Stands for *'Electronic Commerce Indicator'* and indicates whether the transaction is *3DSecure* or not. The values are: • 001: Fully Secure • 002: Partially Secure • 003: Unsecured	C	String (20)
					5	Gateway Trace Number	The reference number provided by the *Acquirer* for a transaction	C	String (35)
					6	Gateway Identifier	This is a short code, which will be sent back in the response to identify the bank, or the gateway on which the transaction was initiated.	C	String (20)
					7	Auth Code	This is the *Authorization Code* for the transaction, received from the *Acquirer/PG*.	C	String (10)
					1	Status Flag	This is the status of the payment. It can have these	R	String (7)

4	3	Transaction Response Status	R	Transaction Status information			values: SUCCESS, FAILURE, and PENDING		
					2	Error Code	Code for transaction response status. If the transaction fails, this represents the failure reason code. Refer to the 'Error Code' section (7.0) for values.	R	String (5)
					3	Error Message	This is a text description for the 'Error Code'. Refer to the Error Code section (7.0) for descriptions of each Error Code.	R	String (200)
5	4	Merchant Information	O	Additional information related to the merchant; same value that is sent in the request	1	UDF1	User defined field; same data as that sent in the request.	O	String (250)
					2	UDF2	User defined field; same data as that sent in the request.	O	String (250)
					3	UDF3	User defined field; same data as that sent in the request.	O	String (250)
					4	UDF4	User defined field; same data as that sent in the request.	O	String (250)
					5	UDF5	User defined field; same data as that sent in the request.	O	String (250)
					6	UDF6	User defined field; same data as that sent in the request.	O	String (250)
					7	UDF7	User defined field; same data as that sent in the request.	O	String (250)
					8	UDF8	User defined field; same data as that sent in the request.	O	String (250)
					9	UDF9	User defined field; same data as that sent in the request.	O	String (250)
					10	UDF10	User defined field; same data as that sent in the request.	O	String (250)
6	5	Fraud Block	O	Fraud Related Information	1	Fraud Decision	This is the result of the action performed by the FRM when it was requested.	O	String (10)

							This can have these values: ACCEPT, REJECT, and REVIEW. **Note:** If the merchant has opted for *Fraud Services* the parameters will be returned in this field.		
					2	Fraud Reason	The reason due to which the FRM module declines a transaction.	O	String (400)
7	6	DCC Block	O	DCC Related Information	1	DCC Converted	Indicates if the transaction is converted in DCC (YES|NO).	O	String (3)
					2	DCC Converted Amount	The converted amount in the customer's card currency, or the chosen currency.	O	Decimal (14,2)
					3	DCC Currency	Currency Code of the converted currency. Refer to the product specifications for possible values.	O	String (3)
					4	DCC Margin	DCC markup that is applied to the wholesale exchange rate.	O	String (10,7)
					5	DCC Exchange Rate	The exchange rate applied for the transaction.	O	String (50)
8	7	Additional	C	Additional Information	1	Card Token	This is a unique value that is generated for every card in the system. If the merchant has opted for the *'Tokenization/Stored Card'* feature then this value will be returned back.	C	String (50)
					2	Card Number	*Last four digits* of the Customer Credit, or Debit card number. Length: 16 digits First 12 digits will be ***masked*** and the last four digits will be in this field.	C	Numeric (20)

Note 5: Response would not be received in the following cases:

- No response received from PG
- Loss of Internet connectivity/browser refreshed at customer end
- Session Timeout

5.0 Formats & Examples

Each Transaction ***Request*** and ***Response*** message will be communicated in an encrypted set-up, in pipe-separated strings format, using ***Bitmap*** methodology. Examples and formats follow.

5.1 Formats & Examples for 'Transaction Requests'

Structure of ***Request parameters*** is as follows:

requestParameter = Merchant ID +"||" Collaborator ID +"||" fieldBitmap + "||" + transactionData + "||" + billingData + "||" + ShippingData + "||" + paymentData + "||" + merchantData + "||" + OtherData + "||" + DCCData

5.11 'Transaction' Data Block Creation Method and Examples for 'Transaction Requests'

transactionData is mandatory for any transaction and contains the following fields:

transactionData = FieldBitmap + "|" + MerchantOrderNumber + "|" + Amount + "|" + SuccessURL + "|" + FailureURL + "|" + TransactionMode + "|" + PayMode Type + "|" + TransactionType + "|" + Currency

Examples**:**

1. transactionData = 11111111|1427091402107|10.00|https://test.com/merchantSuccess.jsp|https://test.com/merchantFail.jsp|INTERNET|CC|02|AED
2. transactionData = 11111111|1427091402107|10.00|https://test.com/merchantSuccess.jsp|https://test.com/merchantFail.jsp|INTERNET|CC|01|AED

5.12 'Billing' Data Block Creation Method and Examples for 'Transaction Requests'

billingData is an optional block for any transaction. If it is used, then it contains the following fields:

billingData = FieldBitmap + "|" + BillToFirstName + "|" + BillToLastName + "|" + BillToStreet1 + "|" + BillToStreet2 + "|" + BillToCity + "|" + BillToState + "|" + BillToPostalCode + "|" + BillToCountry + "|" +BillToEmail + "|" + BillToMobileNumber + "|" + BillToPhoneNumber1 + "|" + BillToPhoneNumber2 + "|" + BillToPhoneNumber3

Examples:

1. billingData = 1111111111111|Soloman|Vandy|123,ParkStreet|Park Street|Mumbai|Maharashtra|400081|IN|solomanv@test.com|9820998209|91|22|39638261
2. billingData = 1110101101111|Soloman|Vandy|123,ParkStreet|Mumbai|400081|UAE|9820998209|971|04|45641321

5.13 'Shipping' Data Block Creation Method and Examples for 'Transaction Requests'

shippingData is an optional block for any transaction. If it is used, then it contains the following fields:

shippingData = FieldBitmap + "|" + ShipToFirstName + "|" + ShipToLastName + "|" + ShipToStreet1 + "|" + ShipToStreet2 + "|" + ShipToCity + "|" + ShipToState + "|" + ShipToPostalCode + "|" + ShipToCountry + "|" + ShipToPhoneNumber1 + "|" + ShipToPhoneNumber2 + "|" + ShipToPhoneNumber3 + "|" + ShipToMobileNumber

Examples:

1. shippingData = 111111111111|Soloman|Vandy|123ParkStreet|parkstreet|Mumbai|Maharashtra|400081|IN|91|22|39638261|9820998209
2. shippingData = 111000011110|Soloman|Vandy|123ParkStreet|UAE|971|04|45641321

5.14 'Payment' Data Block Creation Method and Examples for 'Transaction Requests'

paymentData is a mandatory block for any transaction and contains the following fields:

paymentData = FieldBitmap + "|" + CardNumber + "|" + ExpMonth + "|" + ExpYear + "|" + CVV + "|" + CardHolderName + "|" + CardType + "|" + CustMobileNumber + "|" + PaymentID + "|" + OTP + "|" + GatewayID + "|" + Card Token

Examples:

paymentData = 11111111111|4111111111111111|08|2022|123|Soloman|Visa|9820998209|123456|123456|1026|1202

In case of a ***credit card transaction***, the Payment Data Block formation will be as follows:

paymentData = 11111111111|4111111111111111|08|2022|123|Soloman|Visa|9820998209|123456|123456|1026|1202

In case of a ***Net Banking transaction***, the Payment Data Block formation will be as follows:

paymentData = 00000000010|1001

In case of an ***IMPS transaction***, the Payment Data Block formation will be as follows:

paymentData = 00000011100|9820998209|123456|123456

5.15 'Merchant' Data Block Creation Method and Examples for 'Transaction Requests'

merchantData is an optional Data Block. This is additional information about the transaction that the merchant may send for MIS purposes. If it is there, then it contains the following fields:

merchantData = FieldBitmap + "|" + UDF1 + "|" + UDF2 + "|" + UDF3 + "|" + UDF4 + "|" + UDF5 + "|" + UDF6 + "|" + UDF7 + "|" + UDF8 + "|" + UDF9 + "|" + UDF10

Examples:

1. If UDF1 is an *IP address*, and the other UDFs each have the value 'abc', then
   merchantData = 1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc
2. If UDF1 is an *IP address*, and the other UDFs are blank, then merchantData = 1000000000|115.121.181.112

5.16 'Other' Data Block Creation Method and Examples for 'Transaction Requests'

otherData is an optional Data Block. This is additional information related to the transaction that the merchant may send for MIS purposes. If it is there, then it contains the following fields:

otherData = FieldBitmap + "|" + CustID + "|" + TransactionSource + "|" + ProductInfo + "|" + IsUserLoggedIn + "|" + ItemTotal + "|" + ItemCategory + "|" + IgnoreValidationResult

Examples:

otherData = 1111111|12345|IVR|Book|Y|500.00, 1000.00|CD, Book|FALSE

5.17 'DCC' Data Block Creation Method and Examples for 'Transaction Requests'

DCCData is an optional Data Block. This is information required for 'DCC' only in case the merchant has opted for 'DCC'. If it is there, then it contains the following fields:

DCCData = FieldBitmap + "|" + DCCReferenceNumber + "|" + ForeignAmount + "|" + ForeignCurrency

Examples:

DCCData = 111|09898787|240.00|USD

5.18 Format of Entire 'Transaction Request'

Merchant ID + "||" + Collaborator ID + "||" + Request Packet FieldBitmap + "||" + Block 1 FieldBitmap + "|" + MerchantOrderNumber + "|" + Amount + "|" + SuccessURL + "|" + FailureURL + "|" + TransactionMode + "|" + PayMode Type + "|" + TransactionType + "|" + Currency + "||" + Block 2 FieldBitmap + "|" + BillToFirstName + "|" + BillToLastName + "|" + BillToStreet1 + "|" + BillToStreet2 + "|" + BillToCity + "|" + BillToState + "|" + BillToPostalCode + "|" + BillToCountry + "|" +BillToEmail + "|" + BillToMobileNumber + "|" + BillToPhoneNumber1 + "|" + BillToPhoneNumber2 + "|" + BillToPhoneNumber3 + "||" + Block 3 FieldBitmap + "|" + ShipToFirstName + "|" + ShipToLastName + "|" + ShipToStreet1 + "|" + ShipToStreet2 + "|" + ShipToCity + "|" + ShipToState + "|" + ShipToPostalCode + "|" + ShipToCountry + "|" + ShipToPhoneNumber1 + "|" + ShipToPhoneNumber2 + "|" + ShipToPhoneNumber3 + "|" + ShipToMobileNumber + "||" + Block 4 FieldBitmap + "|" + CardNumber + "|" + ExpMonth + "|" + ExpYear + "|" + CVV + "|" + CardHolderName + "|" + CardType + "|" + CustMobileNumber + "|" + PaymentID + "|" + OTP + "|" + GatewayID + "|" + Card Token + "||" + Block 5 FieldBitmap + "|" + UDF1 + "|" + UDF2 + "|" + UDF3 + "|" + UDF4 + "|" + UDF5 + "|" + UDF6 + "|" + UDF7 + "|" + UDF8 + "|" + UDF9 + "|" + UDF10 + "||" + Block 6 FieldBitmap + "|" + CustID + "|" + TransactionSource + "|" + ProductInfo + "|" + IsUserLoggedIn + "|" + ItemTotal + "|" + ItemCategory + "|" + IgnoreValidationResult + "||" + Block 7 FieldBitmap + "|" + DCCReferenceNumber + "|" + ForeignAmount + "|" + ForeignCurrency

Note 6: In section 5.18, ***'Request Packet FieldBitmap'*** is another term for ***'Block Existence Indicator (BEI)'*** and ***'Block FieldBitmap'*** is another term for ***'Field Existence Indicator (FEI)'***.

5.19 Samples of Entire 'Transaction Request' Formation

Here, as explained earlier, each Data Block is separated by a ***double pipe*** and each field is separated by a ***single pipe***.

Example of all Data Blocks present in the Transaction Request (Credit Card Transaction):

1111111||11111111|1427091402107|10.00|https://test.com/merchantSuccess.jsp|https://test.com/merchantFail.jsp|INTERNET|CC|01|AED||1111111111111|Soloman|Vandy|123,Parkstreet|parkstreet|Mumbai|Maharashtra|400081|IN|Test@toml.com|9820998209|971|04|45641321||111111111111|Soloman|Vandy|Parkstreet|Parkstreet|Mumbai|Maharashtra|400082|IN|971|04|45641321|565051056||11111111111|4111111111111111|08|2022|123|Soloman|Visa|9820998209|123456|123456|1001|1202||1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc||1111111|IVR|345679|Book|Y|500.00, 1000.00|CD, Book|FALSE||111|09898787|240.00|AED

Example of all Data Blocks present in the Transaction Request (Net Banking Transaction):

1110110||11111111|1427091402107|10.00|https://test.com/merchantSuccess.jsp|https://test.com/merchantFail.jsp|INTERNET|CC|01|AED||1111111111111|Soloman|Vandy|123,Parkstreet|parkstreet|Mumbai|Maharashtra|400081|IN|Test@toml.com|9820998209|971|04|45641321||111111111111|Soloman|Vandy|Parkstreet|Parkstreet|Mumbai|Maharashtra|400082|IN|971|04|45641321|565051056||00000000010|1001||1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc||1111111|IVR|345679|Book|Y|500.00, 1000.00|CD, Book|FALSE||111|09898787|240.00|AED

Example of optional Data Blocks ***not*** present in the Transaction Request (Credit Card Transaction):

1001000||11111111|1427091402107|10.00|http://test.com/merchantSuccess.jsp|http://test.com/merchantFail.jsp|INTERNET|CC|01|AED||11111111111|4111111111111111|08|2022|123|Soloman|Visa|9820998209|123456|123456|1001|1202

Example of optional Data Blocks ***not*** present in the Transaction Request (Net Banking Transaction):

1001000||11111111|1427091402107|10.00|http://test.com/merchantSuccess.jsp|http://test.com/merchantFail.jsp|INTERNET|CC|01|AED||00000000010|1001

5.110 Sample of Encrypted Request

requestParameter
=200702091000001||NI||bwjh1K9BhH9LModpp5GIaMkqp7cgy+P044Zb0n9PyHzEM95CA9WVQXM4o0QDfx0870+vtUM+KP
JE+Dge7175+vh5+rNA0pNtoaFby3NZOnQkZehdLeZ7isoNgQsoypqWQsH4vyaiwoeHW6lhhkXoKkWr/uZ+Nb/gJDlluF6Ds4
KBn7Rip2aWmU1YiYhe6VxL7Rs+bqk+7S2317ib8l4HH7GRnLzQowtoTs4kNWj4DNSpfFTAYhCZAXENXnw8OlLn/HeRqAm
5CtDSrSmHibLGLX9jYKYjl90TLCSsGgLgFSOkQR8bbfdx+Asu7B6wDCX59q7+uHinllHaZwxl3vi46FPEVxKDdvwbrHmQBq
b1GCZxb5G5Ul3AAoulnZv/J4zL5z6DthV8FkVKi81MDXpDFCiPX4qjK4Z9LZgcVNZeVU3+HQFNVsQoNQ6xwQyqOlGgdTUU
wHvYDjJBdg8P32kXjLKqqBxF6OMHp4TQqqYZLnj3p40brmwW34WUTB1qnW5E

5.2 Formats & Examples for 'Transaction Responses'

When there is a transaction decline, you will receive a response in the format given below. These are actual transaction responses after requests are submitted to the PSP.

All responses shall be posted via ***'HTTPS POST'*** to the Merchant 'Success', or 'Failure' URL. This is dependent on the status of the transaction, and mainly, on the presence of the mandatory fields, ***SuccessURL*** and ***FailureURL*** in the request, otherwise the PSP will not be able to send the response.

Note 7: The Error Codes that will be received in these responses are *PSP*, or *Payment Gateway* Error Codes. Error Codes indicating the **10000** series are *PSP* Error Codes and those with the **20000** series code are *Payment Gateway* Error Codes. Refer to the 'Error Codes' section for further details.

MerchantID will be sent in *plain text* along with the response, shown as follows:

ResponseBlockBitMap = merchantId + "||" + packetBitMap + "||" + transactionData + "||" + txnResponse + "||" + TxnResponseStatus + "||" + merchantData + "||" + fraudData + "||" + DCCData + "||" + additionalData

5.21 'Transaction' Data Block Creation Method and Examples for 'Transaction Responses'

transactionData is a mandatory block for any transaction, and contains the following fields:

transactionData = FieldBitmap + "|" + MerchantOrderNo + "|" + Currency + "|" + Amount + "|" + PayMode + "|" + CardType + "|" + TransactionType

Example:

1. transactionData = 111111|1427091402107|INR|10.00|CC|Visa|01
2. transactionData = 111101|1427091402107|INR|10.00|NB|02

5.22 'Transaction Response' Data Block Creation Method and Examples for 'Transaction Responses'

'Transaction Response' Data Block is a conditional block for the transaction. In case the transaction is posted to the gateway, or the customer has attempted the transaction i.e. in case the *Reference Number* has been generated for the transaction, you would get this Data Block back in the following format:

TxnResponse = FieldBitmap + "|" + ReferenceNumber + "|" + TxnDate + "|" + CardEnrollmentResponse + "|" + EciIndicator + "|" + GtwTraceNo + "|" + GtwIdentifier + "|" + AuthCode

Example:

TxnResponse = 1111111|1001504007769463|14-May-2015 03:09:52 PM|Enrolled|Fully Secure|12345|CitiPG|83100

5.23 'Transaction Status' Data Block Creation Method and Examples for 'Transaction Responses'

Transaction Status Data Block will always be sent back in the following format, as this will determine the *status* of the transaction:

TxnStatus = FieldBitmap + "|" + StatusFlag + "|" + ErrorCode + "|" + ErrorMessage

Example:

- In case of a 'Success' transaction, you will receive the following Error Code ***'00000'***. In case of a 'Failed' transaction, you will receive the Error Codes mapped in the Error Code table

  TxnStatus = 111|SUCCESS|00000|No Error

- In case of a 'Failed' transaction, you will receive the response, as follows:

  TxnStatus = 111|FAILURE|20003|Insufficient funds in the account.

5.24 'Merchant' Data Block Creation Method and Examples for 'Transaction Responses'

These will be the same values that you had sent in the request:

merchantData = FieldBitmap + "|" + udf1 + "|" + udf2 + "|" + udf3 + "|" + udf4 + "|" + udf5+ "|" + udf6+ "|" + udf7+ "|" + udf8+ "|" + udf9+ "|" + udf10

Example:

3. If UDF1 is an *IP address*, and the other UDFs each have the value 'abc', then
   merchantData = 1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc
4. If UDF1 is an *IP address*, and the other UDFs are blank, then merchantData = 1000000000|115.121.181.112

5.25 'Fraud' Data Block Creation Method and Examples for 'Transaction Responses'

'Fraud' Data Block is an optional block for the transaction. If the merchant has requested for *Fraud Services*, this Data Block represents the result of the action performed by the FRM.

The '*Fraud'* Data Block contains the following fields:

fraudData = FieldBitmap + "|" + frauddecision + "|" + fraudreason

Example:

fraudData = 11|REJECT|The order is rejected by Fraud Module.

5.26 'DCC' Data Block Creation Method and Examples for 'Transaction Responses'

'DCC' Data Block is an optional block for the transaction. If the merchant has requested for *'DCC'*, this Data Block provides information related to DCC.

The '*DCC'* Data Block contains the following fields:

DCCData = FieldBitmap + "|" + DCCConverted + "|" + DCCConverted Amount + "|" + DCC Currency + "|" + DCCMargin + "|" + DCCExchangeRate

Example:

DCCData = 11111|YES|600.00|USD|3.75|0.7178203

5.27 'Additional' Data Block Creation Method and Examples for 'Transaction Responses'

'Additional' Data Block is an optional block for the transaction. It provides information about a card in the system. If the merchant has opted for the feature called '***Tokenization'***, then this Data Block is present and contains the following fields:

additionalData = FieldBitmap + "|" + CardToken + "|" + CardNumber

Example:

additionalData = 11|1202|XXXXXXXXXXXX1234

Note 8: *"Tokenization"* is a feature offered by the system wherein, whenever a customer opts to ***save***, or ***store*** his/her card number in the merchant's website, the saved/stored card is assigned a unique ID by the system, called the ***Card Token***. This token is used to identify the card for all future transactions carried out using the card.

5.28 Format of Entire 'Transaction Response'

Merchant ID + "||" + Response Packet FieldBitMap + "||" + Block 1 FieldBitmap + "|" + MerchantOrderNo + "|" + Currency + "|" + Amount + "|" + PayMode + "|" + CardType + "|" + TransactionType + "||" + Block 2 FieldBitmap + "|" + ReferenceNumber + "|" + TxnDate + "|" + CardEnrollmentResponse + "|" + EciIndicator + "|" + GtwTraceNo + "|" + GtwIdentifier + "|" + AuthCode + "||" + Block 3 FieldBitmap + "|" + StatusFlag + "|" + ErrorCode + "|" + ErrorMessage + "||" + Block 4 FieldBitmap + "|" + udf1 + "|" + udf2 + "|" + udf3 + "|" + udf4 + "|" + udf5+ "|" + udf6+ "|" + udf7+ "|" + udf8+ "|" + udf9+ "|" + udf10+ "||" + Block 5 FieldBitmap + "|" + frauddecision + "|" + fraudreason + "||" + Block 6 FieldBitmap + "|" + DCCConverted + "|" + DCCConverted Amount + "|" + DCC Currency + "|" + DCCMargin + "|" + DCCExchangeRate + "||" + Block 7 FieldBitmap + "|" + CardToken + "|" + CardNumber

Note 9: In section 5.28, ***'Response Packet FieldBitmap'*** is another term for ***'Block Existence Indicator (BEI)'*** and ***'Block FieldBitmap'*** is another term for ***'Field Existence Indicator (FEI)'***.

5.29 Samples of the Entire 'Transaction Response' Formation, which you will Receive

- **For a 'Success' Transaction:**

200702091000001||1111111||111111|1427091402107|INR|10.00|CC|Visa|01||1111111|1001504007769463|14-May-2015 03:09:52 PM|Enrolled|Fully Secure|12345|CitiPG|83100||111|SUCCESS|00000|No Error||1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc||11|REJECT|The order is rejected by Fraud Module||11111|Y|240.00|AED|0.75|0.7178203||11|1202|XXXXXXXXXXXX1234

- **For a 'Failure' Transaction:**

200702091000001||1111111||111111|1427091402107|INR|10.00|CC|Visa|01||1111111|1001504007769463|14-May-2015 03:09:52 PM|Enrolled|Fully Secure|12345|CitiPG|83100||111|FAILURE|20003|Insufficient funds in the account||1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc||11|REJECT|The order is rejected by Fraud Module||11111|Y|240.00|AED|0.75|0.7178203||11|1202|XXXXXXXXXXXX1234

- **In case where the 'Merchant' Data Block is not Received in the Request:**

200702091000001||1110111||111111|1427091402107|INR|10.00|CC|Visa|01||1111111|1001504007769463|14-May-2015 03:09:52 PM|Enrolled|Fully Secure|12345|CitiPG|83100||111|SUCCESS|00000|No Error||11|REJECT|The order is rejected by Fraud Module||11111|Y|240.00|AED|0.75|0.7178203||11|1202|XXXXXXXXXXXX1234

5.210 Sample of Encrypted Response

Response:

200702091000001||bwjh1K9BhH9LModpp5GIaMkqp7cgy+P044Zb0n9PyHzEM95CA9WVQXM4o0QDfx0870+vtUM+KPJE+Dge7175+vh5+rNA0pNtoaFby3NZOnQkZehdLeZ7isoNgQsoypqWQsH4vyaiwoeHW6IhhkXoKkWr/uZ+Nb/gJDIluF6Ds4KBn7Rip2aWmU1YiYhe6VxL7Rs+bqk+7S2317ib8I4HH7GRnLzQowtoTs4kNWj4DNSpfFTAYhCZAXENXnw8OlLn/HeRqAm5CtDSrSmHibLGLX9jYKYjl90TLCSsGgLgFSOkQR8bbfdx+Asu7B6wDCX59q7+uHinlIHaZwxI3vi46FPEVxKDdvwbrHmQBqb1GCZxb5G5Ul3AAoulnZv/J4zL5z6DthV8FkVKi81MDXpDFCiPX4qjK4Z9LZgcVNZeVU3+HQFNVsQoNQ6xwQyqOlGgdTUUwHvYDjJBdg8P32kXjLKqqBxF6OMHp4TQqqYZLnj3p40brmwW34WUTB1qnW5E

5.3 Enabling Server-to-Server Transaction Push Responses

When there is Authorization/Sale success/failure, you will receive an additional server-to-server response in the format given below in this page, if you enable the same. These are actual transaction responses after Authorizations/Sale requests have been submitted to the Gateway.

All responses shall be posted via URL connection to the URL provided by you at for Server-to-Server response. Kindly ensure that you share this URL with the PSP team before moving to production for white-listing at our end. If this URL is not whitelisted at our end then the server-to-server responses will not be sent to you. You will only receive the browser re-direct responses.

Note 10: The Error Codes that will be received in these responses are *PSP*, or *Payment Gateway* Error Codes. Error Codes indicating the **10000** series are *PSP* Error Codes and those with the **20000** series code are 'Payment Gateway' Error Codes. Refer to the 'Error Codes' section in this document for further details.

5.31 Sample of Server-to-Server Transaction Push Responses

responseParameter = <Encrypted String>

This message will be encrypted using the ***AES 256-bit*** encryption and sent to the URL whitelisted at our end. You will receive an encrypted response, which you can decrypt. You can also save the transaction status in your database.

Note 11: Before decrypting the encrypted response you will have to ***decode*** the packet using the following ***URL Decoder***.

URL Decoder for JAVA: cipherPacket = URLDecoder.decode(cipherPacket);

URL Decoder for DotNet (VB & C#): cipherPacket = Server.UrlDecode(cipherPacket);

URL Decoder PHP: $cipherPacket=urldecode($cipherPacket);

For JAVA: Import java.net.*:

For dot net: HttpContext.Current.Server.UrlDecode

For PHP: Inbuilt library in PHP

Code sample for Java programming language, applicable in JSP/Servlet:

```
String decKey = "KtKU+n34+ml3ErXW9+ioPw/ypPiUuce1wunTnAv8e84=";
String encryptedMessage = request.getParameter("responseParameter");
// Decrypt the message using AES 256 bit decryption
EncDec aesDecrypt = new EncDec(decKey, encryptedMessage);
String message = aesDecrypt.decrypt();
```

6.0 Encryption Logic

6.1 Test Merchant Encryption and Decryption Key

[Will be provided at the time of Integration]

6.2 Production Merchant Encryption and Decryption Key

[Will be provided when the Integration is complete and the 'live' MID is created]

6.3 Encryption / Decryption Method

We use ***AES 256-bit*** encryption to ensure secure message transmission.

6.31 Pre-requisites:

1. Java 6 (1.6)
   Note: *If you are using Java 7 then JCE jars will be deployed accordingly.*

2. JCE (Java Cryptography Extension) library *[to be provided as part of the 'Merchant Integration Kit']*
3. NI Encryption / Decryption library *[to be provided as part of the 'Merchant Integration Kit']*
4. Apache Commons Codec library *[to be provided as part of the 'Merchant Integration Kit']*
5. Encryption / Decryption Key *[for **test** environment, use the key given below; for **production** environment, you will get a separate key]*
6. The ***.jar*** files will be provided depending on the platform that is used.

6.32 Set-up Steps:

1. Copy the ***EncDec.jar*** and ***commons-codec-1.8.jar*** files from ***lib*** folder into the classpath.

2. Take backup and replace ***local_policy.jar*** and ***US_export_policy.jar*** files in the ***jre/lib/security*** folder of your installed JRE. Please refer to the README.txt file in the jce folder for detailed description.

6.4 How to Use 'Encryption' or 'Decryption'

Please refer to the following sample java code:

```
package com.sample;
import com.ni.ecom.util.EncDec;
public class TestEncDec {
	public static void main(String[] args) {
String key = "KtKU+n34+ml3ErXW9+ioPw/ypPiUuce1wunTnAv8e84=";
String str
="1111111||11111111|1427091402107|10.00|https://test.com/merchantSuccess.jsp|https://test.com/merchantFail.jsp|INTERNET|CC|01|AED||1111111111111|Soloman|Vandy|123,Parkstreet|parkstreet|Mumbai|Maharashtra|400081|IN|Test@toml.com|9820998209|971|04|45641321||111111111111|Soloman|Vandy|Parkstreet|Parkstreet|Mumbai|Maharashtra|400082|IN|971|04|45641321|565051056||1111111111|4111111111111111|08|2022|123|Soloman|Visa|9820998209|123456|123456|1001|1202||1111111111|115.121.181.112|abc|abc|abc|abc|abc|abc|abc|abc|abc||1111111|IVR|345679|Book|Y|500.00, 1000.00|CD, Book|FALSE||111|09898787|240.00|AED";
EncDec aesEncrypt = new EncDec(key);
String encryptedStr = aesEncrypt.encrypt(str);
			System.out.println("Encrypted String: " + encryptedStr);
			EncDec aesDecrypt = new EncDec(key, encryptedStr);
			String decryptedStr = aesDecrypt.decrypt();
			System.out.println("Decrypted String: " + decryptedStr);
	}
}
```

7.0 Error Codes

If there is any error in the inputs, or in the processing of the inputs given by you (merchant), the application returns an *'Error Code'*. These can be classified as follows:

7.1 PSP Error Codes

The *PSP* errors occur due to invalid user inputs as indicated in Table 3.

Table 3: PSP Error Codes

ERROR CODES			
S.N.	**Field Name/Error Type**	**Error Code**	**Error Description**
1.	MerchantID	**10001**	Invalid Merchant ID
2.	Merchant Transaction Referrer URL	**10002**	Invalid Referral URL
3.	Amount	**10003**	Invalid Amount
4.	ExpiryMonth	**10004**	Invalid Expiry Month
5.	ExpiryYear	**10005**	Invalid Expiry Year
6.	Amount	**10006**	Refund amount greater than transaction amount.
7.	MerchantOrderNumber, CurrencyCode, Amount, Success Url, Failure Url, TransactionType , Transaction Mode, CreditCardNumber , CVV, ExpiryYear, ExpiryMonth, CardType, PayModeType	**10007**	The request is missing one, or more required fields.
8.	TransactionSource, ProductInfo, IsUserLoggedIn,	**10008**	One or more fields in the request contain invalid data.

ERROR CODES			
	ItemTotal, ItemCategory, IgnoreValidationResult, udf1, udf2, udf3, udf4, udf5		
9.	NIOnlineRefID	**10009**	Invalid NIOnlineRefID
10.	Merchant Status	**10010**	Merchant is inactive
11.	Validity	**10011**	Merchant is not in valid period
12.	MerchantOrderNumber	**10012**	Invalid Merchant Order Number If the order number is duplicate and successfully completed (Success transaction), you will receive the following error message: **<Invalid Merchant Order Number. Orderno already processed>** If the order number is duplicate and the transaction is pending for completion you will receive the following error message: **<Invalid Merchant Order Number. Orderno is in pending status>** If the order number is duplicate and the previous attempt was a fail, the system will perform a retry.
13.	Currency	**10013**	Invalid Currency
14.	Currency	**10014**	Currency not mapped to the merchant
15.	Transaction Slab	**10015**	Transaction amount is greater than transaction defined slab.
16.	SuccessURL	**10016**	Please provide a valid success URL.
17.	FailureURL	**10017**	Please provide a valid failure URL.
18.	ResponseURL	**10018**	Please provide a valid response URL.
19.	TransactionType	**10019**	Invalid Transaction Type
20.	TransactionMode	**10020**	Invalid Transaction Mode
21.	PayModeType	**10021**	Invalid Paymode Type

ERROR CODES			
22.	CreditCardNumber	**10022**	Invalid Credit or Debit Card Number
23.	CVV	**10023**	Invalid Card Verification Value
24.	CardType	**10024**	Invalid Card Type
25.	GatewayID	**10025**	Invalid Gateway ID
26.	TransactionSource	**10027**	Invalid Transaction Source
27.	IsUserLoggedIn	**10028**	Invalid user logged in value
28.	Capture Date	**10029**	Invalid Capture Date
29.	Transaction Request Encryption	**10030**	Error while decrypting transaction request
30.	System Mapping	**10031**	One or more system mapping is missing
31.	Customer IP Address	**10032**	Invalid IP address. Transaction not permitted.
32.	Encryption Key	**10033**	Security Key Not Found
33.	Invalid Data (either blanks or incorrect format) in: BillToFirstName, BillToLastName, BillToStreet1, BillToStreet2, BillToCity, BillToState, BillToPostalCode, BillToCountry, BillToEmail, BillToPhoneNumber1, BillToPhoneNumber2, BillToPhoneNumber3, BillToMobileNumber, ShipToFirstName, ShipToLastName, ShipToStreet1, ShipToStreet2, ShipToCity, ShipToState, ShipToCountry, ShipToPostalCode, ShipToPhoneNo1, ShipToPhoneNo2, ShipToPhoneNo3, ShipToMobileNo	**10034**	Invalid Billing/Shipping Information. The response will have details pertaining to the missing field.

ERROR CODES			
34.	CustomerID	**10035**	Invalid Customer ID.
35.	Refund	**10036**	Only Partial Refund or Settled Transactions can be Reversed.
36.	Transaction Type	**10037**	Only Capture and Credit transactions can be Voided.
37.	Transaction Type	**10038**	Only Authorization Transactions can be Reversed.
38.	Refund	**10040**	Insufficient fund in merchant account to process refund
39.	DCC	**10041**	Invalid DCC Reference No.
40.	DCC	**10042**	Invalid DCC Foreign Amount
41.	DCC	**10043**	Invalid DCC Foreign Currency
42.	DCC	**10044**	DCC is not enabled
43.	DCC	**10045**	DCC Reference No. is already utilized
44.	Card Validation	**10048**	Invalid Bin Number
45.	Capture API	**10050**	Only Authorization, or FRM Review Transactions can be captured
46.	Any RunTime Technical Exceptions * Example: - Null Pointer Exceptions - DB Failure Exceptions - Exceptions due to Parameters' Mismanagement	**19999**	Technical Error
47.	Collaborator ID	**10053**	Invalid Collaborator ID

Note 12: If Error Code '19999' is received, please contact the '***Merchant Support Helpdesk***'.

7.2 Payment Gateway Error Codes

When there are errors encountered by the 'Payment Gateway' in the transactions received by them, 'Payment Gateway Error Codes' are generated. These are related to transaction types, as indicated in Table 4.

Table 4: Payment Gateway Error Codes

ERROR CODES			
S.N.	**Field Name/Error Type**	**Error Code**	**Error Description**
1	General Card Credential Error	**20001**	Expired card. You might also receive this if the expiration date you provided does not match the date the issuing bank has on file.
2	General Card Credential Error	**20002**	General decline of the card. No other information was provided by the issuing bank.
3	General Card Credential Error	**20003**	Insufficient funds in the account.
4	General Card Credential Error	**20004**	Stolen or lost card
5	General Card Credential Error	**20005**	Issuing bank unavailable
6	General Card Credential Error	**20006**	Inactive card or card not authorized for card-not-present transactions.
7	General Card Credential Error	**20007**	American Express Card Identification Digits (CID) did not match.
8	General Card Credential Error	**20008**	The card has reached the credit limit.
9	General Card Credential Error	**20009**	Invalid CVN
10	General Card Credential Error	**20010**	Invalid account number
11	General Card Credential Error	**20011**	The card type is not accepted by the payment processor.
12	Capture Amount	**20012**	The requested capture amount exceeds the originally authorized amount.
13	General Card Credential Error	**20013**	This transaction has been declined. Kindly contact credit card issuer bank
14		**20014**	The authorization has already been reversed.
15		**20015**	The authorization has already been captured.
16		**20016**	The requested transaction amount must match the previous transaction amount.
17		**20017**	You requested a capture, but there is no corresponding, unused authorization record.
18		**20018**	The order is rejected by Fraud Module.
19		**20019**	The transaction has already been settled or reversed.
20		**20020**	This transaction failed due to invalid 3D Secure authentication
21		**20021**	Server Timeout
22		**20022**	The request was received but there was a server timeout. This error does not include timeouts between the client and the server.
23		**20023**	The issuing bank has questions about the request.
24		**20024**	The customer matched an entry on the processor's negative file.

ERROR CODES			
25		**20025**	The capture or credit is not voidable because the capture or credit information has already been submitted to your processor.
26		**20026**	You requested a credit for a capture that was previously voided.
27		**20027**	Stand-alone credits are not allowed.
28	Invalid Request *	**20028**	Request data posted is Invalid
29		**20030**	Invalid Card Number
30		**20031**	The same billing address has been used several times with multiple customer identities.
31		**20032**	The same account number has been used several times with multiple customer identities.
32		**20033**	The same email address has been used several times with multiple customer identities.
33		**20034**	The same IP address has been used several times with multiple customer identities.
34		**20035**	The same phone number has been used several times with multiple customer identities.
35		**20036**	The same shipping address has been used several times with multiple customer identities.
36		**20037**	Account is locked or inactive
37		**20038**	Transaction Processing Error at PayPal
38		**20042**	Account is Locked or inactive
39		**20043**	API access is disabled for this account
40		**20044**	The API call has been denied as it has exceeded the permissible call rate limit.
41		**20048**	Receiving Limit exceeded
42		**20055**	Invalid Tax Total for the Transaction
43		**20056**	Authorization has expired
44		**20057**	Authorization feature is not enabled for the merchant. Contact ‘Customer Service’.
45		**20058**	Maximum number of reauthorizations allowed for the ‘auth’ has been reached.
46		**20059**	Transaction has already voided or expired.
47		**20060**	Order has already been voided, expired, or completed.
48		**20061**	Username/Password is incorrect
49		**20062**	You do not have permission to make this API call.
50	Query API	**20063**	The Customer did not go through with the transaction.
51	Query API	**20064**	This transaction has been cancelled, as user did not complete the 3D Secure authentication.
52	Query API	**20065**	The authorization was not complete for the transaction.
53	Query API	**20066**	The transaction was reversed successfully.
54	Query API	**20067**	The transaction was refunded successfully.
55	Query API	**20068**	The transaction was voided successfully.

ERROR CODES			
56	Query API	**20069**	The transaction is successfully settled.
57	Query API	**20070**	The transaction is marked for fraud review.
58	Query API	**20071**	Transaction type mismatch with current status of transaction
59	Issuer not available for authentication	**20072**	Transaction Declined; Issuer unable to authenticate

Note 13: If Error Code '20028' is received, please contact the '***Merchant Support Helpdesk***'.