

**Merchant Integration Document**

*For Payment Gateway*

**Version 1.2**

**January 2018**

*In-Solutions Global Pvt Ltd*

*601, Palm Spring, Link Road,*

*Malad (West)*

*Mumbai 400064*

*Maharashtra, India*

*Phone: +91-22-67603200*

[*www.insolutionsglobal.com*](http://www.insolutionsglobal.com/)

table of contents

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **INTRODUCTION** | | **3** |
| **2** | **TRANSACTION FIELDS** | | **4** |
|  | **2.1** | TRANSACTION REQUEST FIELDS | 4 |
| **2.2** | DETAILS OF REQUEST FIELDS | 7 |
| **2.3** | TRANSACTION RESPONSE FIELDS | 10 |
| **2.4** | DETAILS OF TRANSACTION RESPONSE FIELDS | 12 |
| **2.5** | DATA TRANSMISSION PROTOCOL | 13 |
| **2.6** | CREATING A SHA-256 SIGNATURE FOR TRANSACTIONS | 13 |
| **2.7** | ENCRYPTION | 14 |
| **2.8** | STORAGE OF SECURE HASH SECRET / ENCRYPTION KEY SECURELY | 15 |
| **2.9** | SAMPLE CODE FOR JAVA INTEGRATION | 16 |
| **3** | **TRANSACTION TYPES** | | **19** |
|  | **3.1** | PURCHASE | 19 |
|  | **3.2** | REFUND | 19 |
|  | **3.3** | STATUS | 21 |
| **4** | **RESPONSE CODES** | | **23** |

1 Introduction

Payment Gateway provides merchants a low integration and customized flow driven solution to integrate their payment enabled websites and e-commerce applications with the payment networks. It is suitable for most website hosting environments as merchants can integrate payment capabilities into their application without installing or configuring any payments software.

This guide describes how to enable payment your e-commerce application or on-line store by using the functionality of the Payment Gateway.

2 Transaction fields

This chapter covers the input fields required for basic transactions with description and sample values.

**2.1 TRANSACTION REQUEST FIELDS**

**Table 1**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Field Name** | **Description** |
| 1. | Version | The version of the PG System. |
|  |  |  |
| 2. | TxnRefNo | A unique value created by the merchant  This identifier will be displayed in the Transaction Search results in the Merchant Web Portal of the Payment Gateway |
|  |  |  |
| 3. | Amount | The amount of the transaction, expressed in the smallest currency unit.The amount must not contain any decimal points, thousands separators or currency symbols. For example, Rs. 101.20 is expressed as 10120  This value cannot be negative or zero |
|  |  |  |
| 4. | PassCode | Authenticates the merchant on the Payment Gateway |
|  |  |  |
| 5. | BankId | Unique ID used for identification of bank |
|  |  |  |
| 6. | TerminalId | Card acceptor terminal identification |
|  |  |  |
| 7. | MerchantId | The unique Merchant Id assigned to a merchant by the Payment Provider.The Merchant ID identifies the merchant account against which settlements will be made |
|  |  |  |
| 8. | MCC | MCC(Merchant Category Code) is the Code assigned to business by credit card companies.5974 is used for miscellaneous(different) and specialty retail stores. |
|  |  |  |
| 9. | Currency | Different countries have their currency code. For example,   * INDIA-356 * USA-840 * KUWAIT-414 |
|  |  |  |
| 10. | TxnType | Payment Type   * Purchase-Pay * Refund-Refund * StatusQuery-Status |
|  |  |  |
| 11. | ReturnURL | URL supplied by the merchant .It is used by the Payment Gateway to redirect the card holder's browser back to the merchant's web site.  It must be a fully qualified URL starting with HTTP:// or HTTPS:// and if  typed into a browser with Internet access, would take the browser to that  web page. |
|  |  |  |
| 12. | OrderInfo | Unique identification number of customer order |
|  |  |  |
| 13. | Email | Email Id of customer who placed order |
|  |  |  |
| 14. | Phone | Phone number of customer who placed order |
|  |  |  |
| 15. | payOpt | Payment Method for two party transactions:  cc-Credit Card  dc-Debit Card  nb-NetBanking  wt-Wallet  *For 3-party merchant, if this parameter is sent then, only respective card option will be displayed on card capture page of Payment Gateway* |
|  |  |  |
| 16. | CardNumber | The number of the card used for the transaction .The format of the Card Number is based on the Electronic Commerce Modelling Language (ECML) and , in particular, must not contain white space or formatting characters |
|  |  |  |
| 17. | ExpiryDate | The expiry date of the card in the format MMYYYY. The value must be expressed as a 6-digit number (integer) with no white space or formatting characters  For example, an expiry date of May 2017 is represented as 052017 |
|  |  |  |
| 18. | CardSecurityCode | The Card Security Code (CSC), also known as CVV (Visa), CVC2 (MasterCard) or CID/4DBC (Amex) or CVV2, which is printed, not embossed on the card.It compares the code with the records held in the card issuing institution's database |
|  |  |  |
| 19. | BankCode | Issuing Bank Id assigned by Payment System for Net-banking Transactions |
|  |  |  |
| 20. | FirstName | First name of customer who placed order |
|  |  |  |
| 21. | LastName | Last name of customer who placed order |
|  |  |  |
| 22. | Street | Street(Address) of customer who placed order |
|  |  |  |
| 23. | City | City of customer who placed order |
|  |  |  |
| 24. | ZIP | Postal Code of customer who placed order |
|  |  |  |
| 25. | State | State of customer who placed order |
|  |  |  |
| 26. | SecureHash | This is a hash of the fields sent to ensure integrity of the transaction data.  For more details refer “Creating a SHA-256 Signature for Transactions” on page 11.  **Note:** The secure secret is provided by the Payment Provider. |
|  |  |  |
| 27. | EncData | The encrypted value of request parameters is sent  For more details refer “ENCRYPTION” on page 12.  **Note:** The encryption key is provided by the Payment Provider. |
|  |  |  |
| 28. | splitPaymentType | If merchant wants to split settlement in different account for particular transaction then this parameter either set to “P” (Split amount in Percentage) or “A” (Split in Amount) |
|  |  |  |
| 29. | splitPaymentInfo | If splitPaymentType is sent in request then this parameter is mandatory. This parameter contains account number, IFSC code and amounts/percentage for split payment. |
|  |  |  |
| 30. | UDF01 | User Defined Field |
|  |  |  |
| 31. | UDF02 | User Defined Field |
|  |  |  |
| 32. | UDF03 | User Defined Field |
|  |  |  |
| 33. | UDF04 | User Defined Field |
|  |  |  |
| 34. | UDF05 | User Defined Field |
|  |  |  |
| 35. | UDF06 | User Defined Field |
|  |  |  |
| 36. | UDF07 | User Defined Field |
|  |  |  |
| 37. | UDF08 | User Defined Field |
|  |  |  |
| 38. | UDF09 | User Defined Field |
|  |  |  |
| 39. | UDF10 | User Defined Field |

A fully qualified URL (starting with HTTPS://). It must be included in the merchant's application code to send transaction information to the Payment Gateway.

**UAT URL:** https://sandbox.isgpay.com:8443/ISGPay/request.action

**Production URL:** https://isgpay.com/ISGPay/request.action

**2.2 DETAILS OF REQUEST FIELDS**

The details of input fields with sample values are provided in table 2.

**Table 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Mandatory /**  **Optional /**  **Conditional** | **Data Type** | **Length** | **Sample Data** |  |  |
| 1. | Version\*+ | M | Numeric | 1 | 1 |  |  |
|  |  |  |  |  |  |  |  |
| 2. | TxnRefNo**\*** | M | Alphanumeric | 1-40 | TEST-160816111226 |  |  |
|  |  |  |  |  |  |  |  |
| 3. | Amount**\*** | M | Numeric | 1-12 | 2995 |  |  |
|  |  |  |  |  |  |  |  |
| 4. | PassCode**\*** | M | Alphanumeric | 8 | JJRS7608 |  |  |
|  |  |  |  |  |  |  |  |
| 5. | BankId**\*+** | M | Alphanumeric | 6 | ISG027 |  |  |
|  |  |  |  |  |  |  |  |
| 6. | TerminalId**\*+** | M | Alphanumeric | 8 | 20001562 |  |  |
|  |  |  |  |  |  |  |  |
| 7. | MerchantId**\*+** | M | Alphanumeric | 15 | TESTMERCHANT001 |  |  |
|  |  |  |  |  |  |  |  |
| 8. | MCC\* | M | Numeric | 4 | 5974 |  |  |
|  |  |  |  |  |  |  |  |
| 9. | Currency**\*** | M | Numeric | 3 | 356 |  |  |
|  |  |  |  |  |  |  |  |
| 10. | TxnType**\*** | M | Alphabetic | 3-7 | Pay |  |  |
|  |  |  |  |  |  |  |  |
| 11. | ReturnURL\* | M | Alphanumeric | 1-225 | https://uat-geniusepay.in/GeniusPGRedirect/merchant/merchant\_ema\_dr.jsp |  |  |
|  |  |  |  |  |  |  |  |
| 12. | OrderInfo | O | Numeric | 1-40 | 6075000000000023 |  |  |
|  |  |  |  |  |  |  |  |
| 13. | Email | O | Alphanumeric | 1-40 | abc@xyz.com |  |  |
|  |  |  |  |  |  | | |
| 15. | Phone | O | Numeric | 12 | 919012345678 | | |
|  |  |  |  |  |  | | |
| 15. | payOpt | C | Alphabetic | 2-4 | dc | | |
|  |  |  |  |  |  | | |
| 16. | CardNumber | C | Numeric | 13-19 |  | | |
|  |  |  |  |  |  | | |
| 17. | ExpiryDate | C | Numeric | 6 | 012020 | | |
|  |  |  |  |  |  | | |
| 18. | CardSecurityCode | C | Numeric | 3 | 123 | | |
|  |  |  |  |  |  | | |
| 19. | BankCode | C | Alphanumeric | 6 | ISG001 | | |
|  |  |  |  |  |  | | |
| 20. | FirstName | C | Alphabetic | 1-30 | JOHN | | |
|  |  |  |  |  |  | | |
| 21. | LastName | C | Alphabetic | 1-30 | WILLIAMS | | |
|  |  |  |  |  |  | | |
| 22. | Street | C | Alphabetic | 1-30 | GUNBOW STREET | | |
|  |  |  |  |  |  | | |
| 23. | City | C | Alphabetic | 1-30 | MUMBAI | | |
|  |  |  |  |  |  | | |
| 24. | ZIP | C | Numeric | 6 | 400001 | | |
|  |  |  |  |  |  | | |
| 25. | State | C | Alphabetic | 1-40 | MAHARASHTRA | | |
|  |  |  |  |  |  | | |
| 26. | splitPaymentType | O | Alphabetic | 1 | P : Percentage A : Amount | | |
|  |  |  |  |  |  | | |
| 27. | splitPaymentInfo | C\* | Alphabetic | 1-3999 | When splitPaymentType=P 11110011~YBL0001~50#11110012~YBL0002~50  When splitPaymentType=A and Amount=10000 then 11110011~YBL0001~5000#11110012~YBL0002~5000  \*This parameter is mandatory when splitPaymentType parameter is sent | | |
|  |  |  |  |  |  | | |
| 28. | UDF01 | O | Alpha-Numeric | 1-500 |  | | |
| 29. | UDF02 | O | Alpha-Numeric | 1-500 |  | | |
| 30 | UDF03 | O | Alpha-Numeric | 1-500 |  | | |
| 31. | UDF04 | O | Alpha-Numeric | 1-500 |  | | |
| 32. | UDF05 | O | Alpha-Numeric | 1-500 |  | | |
| 33. | UDF06 | O | Alpha-Numeric | 1-500 |  | | |
| 34. | UDF07 | O | Alpha-Numeric | 1-500 |  | | |
| 35. | UDF08 | O | Alpha-Numeric | 1-500 |  | | |
| 36. | UDF09 | O | Alpha-Numeric | 1-500 |  | | |
| 37. | UDF10 | O | Alpha-Numeric | 1-500 |  | | |

**Note:**

* The fields marked with asterisk (\*) are only mandatory.
* Fields marked with Plus (+) are to be send externally along with EncData.

**2.3** **TRANSACTION RESPONSE FIELDS**

The details of input fields with sample values are provided in table 3.

**Table 3**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Field Name** | **Description** |
| 1. | BankId | Unique ID used for identification of bank |
|  |  |  |
| 2. | TxnRefNo | A unique value created by the merchant |
|  |  |  |
| 3. | MerchantId | The unique Merchant Id assigned to a merchant by the Payment Provider. The Merchant ID identifies the merchant account against which settlements will be made |
|  |  |  |
| 4. | Amount | The amount of the transaction, expressed in the smallest currency unit. |
|  |  |  |
| 5. | TerminalId | Card acceptor terminal identification |
|  |  |  |
| 6. | ResponseCode | A response code that is generated by the Payment Server to indicate the status of the transaction. A ResponseCode of “00” (Double zero) indicates that the transaction was processed successfully and approved by the acquiring bank. Please refer the sheet i.e. “RESPONSE CODES”from page **23** |
|  |  |  |
| 7. | Message | This is a message to indicate what sort of error, if any, the transaction encountered |
|  |  |  |
| 8. | RetRefNo | RetRefNo(Reference Retrieval Number or RRN)is a unique identifier that is passed back to the cardholder for their records if the merchant application does not generate its own receipt number. |
|  |  |  |
| 9. | AuthCode | Authorization Code assigned by Card Issuer upon approval of transaction |
|  |  |  |
| 10. | FirstName | First name of the customer |
|  |  |  |
| 11. | LastName | Last name of the customer |
|  |  |  |
| 12. | AddressZip | Address of customer |
| 13. | UDF01 | User Defined Field |
| 14. | UDF02 | User Defined Field |
| 15. | UDF03 | User Defined Field |
| 16. | UDF04 | User Defined Field |
| 17. | UDF05 | User Defined Field |
| 18. | UDF06 | User Defined Field |
| 19. | UDF07 | User Defined Field |
| 20. | UDF08 | User Defined Field |
| 21. | UDF09 | User Defined Field |
| 22. | UDF10 | User Defined Field |

**2.4** **DETAILS OF TRANSACTION RESPONSE FIELDS**

**Table 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Required / Optional**  **/ Conditional** | **Data Type** | **Length** | **Sample Data** |
| 1. | BankId | M | Alphanumeric | 6 | ISG027 |
|  |  |  |  |  |  |
| 2. | TxnRefNo | M | Alphanumeric | 1-40 | TEST-160816111226 |
|  |  |  |  |  |  |
| 3. | MerchantId | M | Alphanumeric | 1-16 | TESTMERCHANT001 |
|  |  |  |  |  |  |
| 4. | Amount | M | Numeric | 1-12 | 2995 |
|  |  |  |  |  |  |
| 5. | TerminalId | M | Alphanumeric | 8 | 20001562 |
|  |  |  |  |  |  |
| 6. | ResponseCode | M | Numeric | 2 | 00 |
|  |  |  |  |  |  |
| 7. | Message | M | Alphanumeric | 1-255 | SUCCESS |
|  |  |  |  |  |  |
| 8. | RetRefNo | C | Numeric | 12 | 622966334257 |
|  |  |  |  |  |  |
| 9. | AuthCode | C | Alphanumeric | 6 | 123456 |
|  |  |  |  |  |  |
| 10. | FirstName | C | Alpha | 1-25 | JOHN |
|  |  |  |  |  |  |
| 11. | LastName | C | Alpha | 1-25 | WILLIAMS |
|  |  |  |  |  |  |
| 12. | AddressZip | C | Alphanumeric | 1-25 | 400001 |

**2.5** **DATA TRANSMISSION PROTOCOL**

* Data is sent via a form POST to https://URL/
* Does not support GET data transfer. The request will be rejected

**2.6** **CREATING AN SHA-256 SIGNATURE FOR TRANSACTIONS**

The merchant code creates the SHA-256 Secure Hash value on the Transaction Request data. The Payment Gateway creates another SHA-256 Secure Hash value and sends it back to the merchant in the Transaction Response.

The Secure Hash is a Hex encoded SHA-256 output of a concatenation of all the data parameters. The order that the data parameters are hashed in is extremely important as different transactions contain different data fields so rather than giving the explicit order for each parameter, the order that parameters are hashed in should follow the following rules:

* The Secure Hash Secret is always first.
* Then all parameters are concatenated to the secret in alphabetical order of the parameter name. More specifically, the data sort should be in ascending order of the ASCII value of each parameter's name, for example, 'Card' comes before 'card'. Where one string is an exact substring of another, the smaller string should be ordered before the longer, for example, 'Card' should come before 'CardNum'.
* Fields must not have any separators between them and must not include any null terminating characters or the like. For example, if the secret is DEC2BEE8967AF2911BE26727A3C6D69B, and the Transaction Request includes only the following parameters:

|  |  |
| --- | --- |
| **Field Name** | **Example Value** |
| MerchantId | TESTMERCHANT001 |
| TxnRefNo | TEST-160816111226 |
| Amount | 2995 |

In ascending alphabetical order, the input to the SHA-256 Secure Hash creation routine would be:

**DEC2BEE8967AF2911BE26727A3C6D69B2995TESTMERCHANT001TEST-160816111226**

This string is then Hex encoded and then passed through the merchant's SHA-256 Secure Hash generator in the programming language the merchant is using. This output (for example, a value of 4ec7fdf28fb8da74bebb2edf8f79920e12302d56b3e0a8a2c4fc2a2a53d0901a) is then included in the Transaction Request using the SecureHash field.

The Virtual Payment Client also includes the SecureHash in the Transaction Response so the merchant can check the security of the receipt data. This is performed by first stripping off the SecureHash, and then performing the same steps as creating an SHA-256 Secure Hash for the Transaction Request, but using the received Transaction Response data fields instead. The received SecureHash is then compared with the SHA-256 Secure Hash calculated from the Transaction Response data.

If both SHA-256 signatures are the same, the data has not been changed in transit. If they are different the data needs to be doubled checked.

**2.7** **CREATING AN ENCRYPTED VALUE FOR TRANSACTIONS**

The merchant code encrypts the Transaction Request data. The Payment Gateway also sends back the Transaction Response to the merchant in encrypted form. The following request parameters are encrypted using the industry approved AES-256 algorithm.

* Amount
* PassCode
* CardNumber
* ExpiryDate
* CardSecurityCode

The encrypted value is generated creating the string of the above the request parameters with the parameters separated by “ :: (double colon)” .The Parameter name and value separated are by “||(double pipes)” .Fields must not have any separators between them and must not include any null terminating characters. For example, consider the following parameters:

|  |  |
| --- | --- |
| **Field Name** | **Example Value** |
| PassCode | JJRS7608 |
| Amount | 2995 |
| CardNumber | 4012001037141112 |
| ExpiryDate | 012020 |
| CardSecurityCode | 123 |

the input to the AES-256 Encryption would be:

**PassCode||JJRS7608::Amount||100::CardNumber||4012001037141112::ExpiryDate||012020::CardSecurityCode||123**

This string is then encrypted using AES-256 Encryption algorithm in the programming language the merchant is using. The output (for example, a value of **KtEzii5R2N0SKG991tKoNSlHGpi80r21VQzhwFeiJc1lkujkUztxRZcfiO53S7OEuqJXXRzBM**

**pk4p9hvges8SewP+47NIZiFJBG0AkOPMVF3JkrU40bw9Ivw4FOF+3MF** is then included in the Transaction Request using the “EncData“ field.

**Note:**

* The encryption key is provided by the Payment Provider.
* Encryption is implemented only for purchase transactions.
* ***You need to encrypt all the request parameter.***

**2.8 STORAGE OF SECURE HASH SECRET/ENCRYPTION KEY SECURELY**

You must keep your Secure Hash Secret/Encryption Key stored securely. Do not store it within the source code of an ASP, JSP, or other website page as it is common for web server vulnerabilities to be discovered where source code of such pages can be viewed.

You should store it in a secured database, or in a file that is not directly accessible by your web server and has suitable system security permissions.

You should change your Secure Hash Secret/Encryption Key regularly in accordance with your company's security policy, and any time when you believe that its security may have been compromised.

**2.9 SAMPLE CODE FOR JAVA INTEGRATION**

For Java-JSP integration, you will need ISGPayEncDec\_v1.0.jar file. This jar will do all encryption / decryption as well as hashing for you. There are different overloaded methods provided for encryption & decryption. They are as follow:

1. ENCRYPTION FOR REQUEST FROM MERCHANT TO ISGPay PAYMENT GATEWAY:

For sending request to PG, Jar provides ISGPayEncryption class. This class’s method will do all your encryption & hashing as per ISGPay PG accepts. There are 3-ways you can achieve encryption 1st you need to create object of ISGPayEncryption class:

* + 1. Method 1st: *encrypt(HttpServletRequest, String, String)* method.
       1. 1st parameter is HttpServletRequest object.
       2. 2nd parameter is Encryption key provided by ISG for doing encryption.
       3. 3rd parameter is SALT(SecureSecret) provided by ISG for hash value generation.

Here is sample code for this method:  
  
*ISGPayEncryption encObj = new ISGPayEncryption();*

*encObj.encrypt(request, sEncryptionKey, sSecureSecret);*

* + 1. Method 2nd: e*ncrypt(LinkedHashMap<String, String>, String, String)* method.
       1. 1st parameter is Map object which have all your request parameter name as key and request values as value along with Merchant ID, Terminal ID, Bank ID, and Version Number.
       2. 2nd parameter is Encryption key provided by ISG for doing encryption.
       3. 3rd parameter is SALT(SecureSecret) provided by ISG for hash value generation.

Here is sample code for this method:

*ISGPayEncryption encObj = new ISGPayEncryption();*

*LinkedHashMap<String, String> hmReqFields = new LinkedHashMap<String, String>();*

*...*

*...*

*...*

*...*

*encObj.encrypt(hmReqFields, sEncryptionKey, sSecureSecret);*

* + 1. Method 3rd: *encrypt(LinkedHashMap<String,String>, String, String, String, String, String, String)* method:
       1. 1st parameter is Map object which have all your request parameter name as key and request values as value along with Merchant ID, Terminal ID, Bank ID, and Version Number.
       2. 2nd parameter is your Merchant ID.
       3. 3rd parameter is your Terminal ID.
       4. 4th parameter is your Bank ID.
       5. 5th parameter is kit version number which you are using.
       6. 6th parameter is Encryption key provided by ISG for doing encryption.
       7. 7th parameter is SALT(SecureSecret) provided by ISG for hash value generation.

Here is sample code for this method:

*ISGPayEncryption encObj = new ISGPayEncryption();*

*LinkedHashMap<String, String> hmReqFields = new LinkedHashMap<String, String>();*

*...*

*...*

*...*

*...*

*encObj.encrypt(hmReqFields, sMerchantId, sTerminalID, sBankID, sVersion, sEncryptionKey, sSecureSecret);*

After calling any of the above method, you can get Merchant ID, Terminal ID, Bank ID, Version Number & Encrypted data with hash included by calling *getMERCHANT\_ID(),getTERMINAL\_ID(),getBANK\_ID(),getVERSION()* and *getENC\_DATA*() respectively. Here is sample code you can use :

*<input type="hidden" name="MerchantId" id="MerchantId" value="<%=encObj.getMERCHANT\_ID()%>"/>*

*<input type="hidden" name="TerminalId" id="TerminalId" value="<%=encObj.getTERMINAL\_ID()%>"/>*

*<input type="hidden" name="BankId" id="BankId" value="<%=encObj.getBANK\_ID()%>"/>*

*<input type="hidden" name="Version" id="Version" value="<%=encObj.getVERSION()%>"/>*

*<input type="hidden" name="EncData" id="EncData" value="<%=encObj.getENC\_DATA()%>"/>*

Note : 1. For request generation, refer MerchantRedirect.jsp file.

2. You will get error messages if any of the field is missing or some exception occurs.

1. DECRYPTION FOR RESPONSE FROM ISGPay TO MERCHANT:

For getting response from PG, jar provides ISGPayDecryption class to help you with the decryption and hash matching part. There are 2-ways you can do decryption & hash matching . 1st you need to create object of ISGPayDecryption class. Then you can call any one of below two methods :

* + 1. Method 1st: LinkedHashMap<String, String> decrypt(HttpServletRequest, String, String) method:
       1. 1st parameter is HttpServletRequest object.
       2. 2nd parameter is your encryption key provided by ISG which will be used for decryption purpose.
       3. 3rd parameter is your Salt(SecureSecret) provided by ISG which will be used for hash generation of response from PG.

Here is sample code for this method :

ISGPayDecryption decObj = new ISGPayDecryption();

LinkedHashMap<String, String> hmDecryptedValue = decObj.decrypt(request, sDecryptionKey, sSecureSecret);

* + 1. Method 2nd: decrypt(LinkedHashMap<String, String>, String ,String ) method:
       1. 1st parameter is Map object which have all your request parameter name as key and request values as value.
       2. 2nd parameter is your encryption key provided by ISG which will be used for decryption purpose.
       3. 3rd parameter is your Salt(SecureSecret) provided by ISG which will be used for hash generation of response from PG.

Here is sample code for this method :

*LinkedHashMap<String, String> hmDecryptedValue = new LinkedHashMap<String, String>();*

*...*

*...*

*...*

*ISGPayDecryption decObj = new ISGPayDecryption();*

*decObj.decrypt(hmDecryptedValue, sDecryptionKey, sSecureSecret);*

Map you have passed as parameter will only be used to get all decrypted response parameter.   
  
Note : 1. For getting response from PG, refer MerchantResponse.jsp file.

2. You will get error messages if any of the field is missing or some exception occurs in returned Map with “ErrorMessage” key .

3. You can check response hash is matched or not with returned Map’s “hashValidated” key value. If it is “CORRECT” then hash is matched and if it is “INVALID HASH” then response hash and generated hash is not matched.

3 Transaction TYPES

This chapter covers the transaction types supported by the ISGPay.

**3.1** **PURCHASE**

**Request Transaction –**

The list of fields required for each Purchase transaction request is listed in table 1 & 2 with appropriate

‘TxnType’ value.

**Response Transaction –**

The list of fields that will be sent in response for each Purchase transaction request is listed in table 3 & 4

**3.2 REFUND**

**Request Transaction –**

The list of fields required for each Refund transaction request is listed in table 5 with appropriate

‘TxnType’ value.

**Response Transaction –**

The list of fields that will be sent in response for each Refund transaction request is listed in table 6

**REFUND REQUEST:**

**Table 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Required / Optional /**  **Conditional** | **Data Type** | **Length** | **Sample Data** |
| 1. | TxnRefNo | M | Alphanumeric | 1-40 | TEST-160816111226 |
|  |  |  |  |  |  |
| 2. | PassCode | M | Alphanumeric | 10 | JJRS7608 |
|  |  |  |  |  |  |
| 3. | TerminalId | M | Alphanumeric | 8 | 20001562 |
|  |  |  |  |  |  |
| 4. | MerchantId | M | Alphanumeric | 1-16 | TESTMERCHANT001 |
|  |  |  |  |  |  |
| 5. | BankId | M | Alphanumeric | 6 | ISG027 |
|  |  |  |  |  |  |
| 6. | TxnType | M | Alphabetic | 3-7 | Refund |
| 7. | RetRefNo | C | Numeric | 12 | 622966334257 |
|  |  |  |  |  |  |
| 8. | AuthCode | C | Alphanumeric | 6 | 123456 |
|  |  |  |  |  |  |
| 9. | RefundAmount | C | Numeric | 1-12 | 10 |

**REFUND REPONSE:**

**Table 6**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Required / Optional /**  **Conditional** | **Data Type** | **Length** | **Sample Data** |
| 1. | TxnRefNo | M | Alphanumeric | 1-40 | TEST-160816111226 |
|  |  |  |  |  |  |
| 2. | PassCode | M | Alphanumeric | 8 | JJRS7608 |
|  |  |  |  |  |  |
| 3. | TerminalId | M | Numeric | 8 | 20001562 |
|  |  |  |  |  |  |
| 4. | MerchantId | M | Alphanumeric | 1-16 | TESTMERCHANT001 |
|  |  |  |  |  |  |
| 5. | TxnType | M | Alpha | 2 | Refund |
|  |  |  |  |  |  |
| 6. | RetRefNo | C | Numeric | 12 | 622966334257 |
|  |  |  |  |  |  |
| 7. | AuthCode | C | Numeric | 6 | 123456 |
|  |  |  |  |  |  |
| 8. | RefundAmount | M | Numeric | 1-12 | 10 |
|  |  |  |  |  |  |
| 9. | ResponseCode | M | Alphanumeric | 1-3 | 00 |
|  |  |  |  |  |  |
| 10. | Amount | C | Numeric | 1-12 | 100 |
|  |  |  |  |  |  |
| 11. | BankId | M | Alphanumeric | 6 | ISG027 |
|  |  |  |  |  |  |
| 12. | Message | M | Alphanumeric | 1-255 | Transaction successful |

**3.6 STATUS QUERY**

**Request Transaction –**

The list of fields required for each Status transaction request is listed in table 7 with appropriate ‘TxnType’ value.

**Response Transaction –**

The list of fields that will be sent in response for each Status transaction request is listed in table 8

**STATUS REQUEST:**

**Table 7**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Required /**  **Optional /**  **Conditional** | **Data Type** | **Length** | **Sample Data** |
| 1. | TxnRefNo | M | Alphanumeric | 1-40 | TEST-160816111226 |
|  |  |  |  |  |  |
| 2. | MerchantId | M | Alphanumeric | 1-16 | TESTMERCHANT001 |
|  |  |  |  |  |  |
| 3. | BankId | M | Alphanumeric | 6 | ISG027 |
|  |  |  |  |  |  |
| 4. | PassCode | M | Alphanumeric | 10 | JJRS7608 |
|  |  |  |  |  |  |
| 5. | TerminalId | M | Alphanumeric | 8 | 20001562 |
|  |  |  |  |  |  |
| 6. | TxnType | M | Alphabetic | 3-7 | Status |

**STATUS RESPONSE:**

**Table 8**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Required /**  **Optional /**  **Conditional** | **Data Type** | **Length** | **Sample Data** |
| 1. | TxnRefNo | M | Alphanumeric | 1-40 | TEST-160816111226 |
|  |  |  |  |  |  |
| 2. | MerchantId | M | Alphanumeric | 1-16 | TESTMERCHANT001 |
|  |  |  |  |  |  |
| 3. | BankId | M | Alphanumeric | 6 | ISG027 |
|  |  |  |  |  |  |
| 4. | TerminalId | M | Alphanumeric | 8 | 20001562 |
|  |  |  |  |  |  |
| 5. | TxnType | M | Alphabetic | 3-7 | Status |
|  |  |  |  |  |  |
| 6. | ResponseCode | M | Alphanumeric | 3 | 00 |
|  |  |  |  |  |  |
| 7. | Message | M | Alphanumeric | 1-255 | Transaction Successful |
|  |  |  |  |  |  |
| 8. | Amount | C | Numeric | 1-12 | 2995 |
|  |  |  |  |  |  |
| 9. | RetRefNo | C | Numeric | 12 | 622966334257 |
|  |  |  |  |  |  |
| 10. | AuthCode | C | Numeric | 6 | 123456 |
|  |  |  |  |  |  |

4 response codes

This chapter covers the common Response Codes supported by the ISG pay.

|  |  |  |
| --- | --- | --- |
| **Response Code** | **Response Message** | **Details** |
| 00 | Success | Successful transaction |
| VER | Validation Error | Occurs if field data is incorrect |
| HNM | Hash Not Match | Occurs if the data is tampered |
| MNE | Merchant Not Enrolled | Occurs if merchant is not enrolled |
| STO | Session Timeout | User’s session is timed out |
| IER | Internal Error | System Error |
| CAN | Cancel | User pressed Cancel Button |
| IVR | Invalid Request | Invalid Request |
| RAE | Request Amount Exceeded | Occurs if Request Amount is greater than Purchase Transaction Amount. |
| RNF | Request Not Found | Requested Transaction not found |
| RDF | Duplicate Request Found | Duplicate Transaction Request found |
| DOI | Duplicate Order ID | Duplicate order id sent by merchant |
| 01 | Refer To Card Issuer | Card Issuer has indicated there is a problem with the card number. The customer should contact their bank or the customer should use an alternate card. |
| NBF | Transaction Failed | When Net-Banking transaction is failed |
| 03 | Invalid Merchant Details | This Error indicates that either your merchant is not available or the details entered are incorrect. |
| 04 | Capture Card Or Hot listed Card | Issuer Specific Decline The card has been reported lost or stolen |
| 05 | Do Not Honor | Issuer specific decline when CCV or Expiry Date doesn’t Match. |
| N7 | Do Not Honor | Issuer specific decline when CCV or Expiry Date doesn’t Match. |
| 06 | Issuer System Error | From Issuer we are receiving Response |
| 07 | Pickup Card | Issuer Specific Decline The card has been reported lost or stolen |
| 08 | Transaction Timed Out | Failed to received response from Visa/Master Timeout |
| 10 | Partial Approval | Approved for Partial Amount |
| 12 | Invalid Transaction | Issuer has declined the transaction because of an invalid format or field. This indicates the card details were incorrect. Check card data entered and try again. |
| IT | Invalid Transaction | Issuer has declined the transaction because of an invalid format or field. This indicates the card details were incorrect. Check card data entered and try again. |
| 13 | Invalid Amount | Issuer has declined the transaction because of an invalid Amount format or field. |
| 14 | Invalid Card Number | Issuer has declined the transaction as the credit card number is incorrectly entered, or does not exist. Check card details and try processing again. |
| 15 | Invalid Issuer | The customer’s card issuer does not exist. Check the card information and try processing the transaction again. |
| 21 | No Action Taken | The customer’s card issuer has indicated there is a problem with the card number. The customer should contact their card issuer and/or use an alternate credit card. |
| 25 | Unable To Locate Record In File | Issuer does not recognize the card details. The customer should check the card information and try processing the transaction again. |
| 30 | Switch ISO Format Error (Invalid Acquirer Institute ID) | Issuer does not recognize the transaction details being entered. This is due to a Data format error. |
| FE | Switch ISO Format Error (Merchant Request Data Format Issue) | Issuer does not recognize the transaction details being entered. This is due to a Data format error. |
| 31 | Invalid BIN | issuer has declined the transaction as BIN is Not Present or Not a valid Bin |
| 32 | Partial Reversal | Approved for Partial Amount Reversal |
| 34 | Suspected Fraud | Issuer has declined the transaction as there is a suspected fraud on this Card number. |
| 59 | Suspected Fraud | Issuer has declined the transaction as there is a suspected fraud on this Card number. |
| 41 | Lost Card | Issuer has declined the transaction as the card has been reported lost |
| 43 | Stolen Card | Card has been reported as Stolen. |
| 51 | Insufficient Funds | Insufficient Funds |
| 52 | No Checking Account | Issuer has declined the transaction as the card number is associated to a cheque account that does not exist. |
| 53 | No Savings Account | Issuer has declined the transaction as the card number is associated to a savings account that does not exist. |
| 54 | Expired Card | Issuer has declined the transaction as the card appears to have expired |
| 57 | Transaction Not Permitted To Issuer Or Cardholder | Issuer has declined the transaction as this card cannot be used for this type of transaction. |
| 58 | Transaction Not Permitted To Acquirer Or Merchant | Bank has declined the transaction as this card cannot be used for this type of transaction, associated with a test credit card number. The customer should use an alternate credit card or contact their bank. |
| 60 | Contact Card Acquirer | Bank has declined the transaction. The customer should contact their bank and retry the transaction. |
| 61 | Exceeds Withdrawal Amount Limit | Issuer has declined the transaction as it will exceed the customer’s card limit. |
| 62 | Restricted Card | Issuer has declined the transaction as the card has some restrictions. The customer should contact their bank for further information. |
| 63 | Security Violation | Issuer has declined the transaction. The customer should use an alternate credit card and contact their card issuer if the problem persists. |
| 65 | Exceeds Withdrawal Count Limit | Issuer has declined the transaction as the customer has exceeded the withdrawal frequency limit. |
| 68 | Response Received Late | Transaction Response not receive in time |
| 70 | Contact Card Issuer | Issuer has declined the transaction. The customer should contact their bank and retry the transaction. |
| 91 | Issuer Unavailable | Issuer is unable to be contacted to authorize the transaction |
| 94 | Duplicate Transmission Detected | Issuer has declined the transaction as this transaction appears to be a duplicate transmission |
| 96 | Issuer System Failure | Issuer was not able to process the transaction. The customer should attempt to process this transaction again. |
| D1 | Connection Timed Out While Connecting to Visa/Master Directory Server | Internal issue related to PG. |
| D2 | Visa/master Card Directory Server didn’t Send any Response in Specified Time | Visa / MasterCard Directory server not responded within specified time duration. |