TechProcess Payment Services Ltd

Single URL -Integration & Installation Guide

Last updated 23rd JAN 2018 Mumbai

“Current Kit Version 5.4”

**Document Control**

|  |  |
| --- | --- |
| Document Title | Single URL Integration and Installation Kit - For Java Framework 1.5 and Higher Versions |
| Document ID |  |
| Document Current Version | CURRENT KIT VERSION 5.3 |
| Effective Date | 23rd JAN 2018 |

**Document Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version No. | Version Date | Revised By | Revision  Description | Change Approver | |
| Name | Designation |
| Ver. 5.4 | 23rd JAN 2018 |  | Process Described | 'Rajeev Narayanan' | EVP |
|  |  |  |  |  |  |

**Document Revision Log For The Project**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Project Version | Change Reference | Changed By |
|  |  |  |  |
|  |  |  |  |

**Document Publication History**

|  |  |
| --- | --- |
| Prepared By  Name – Preeti Jha  Designation – Sr. Business Analyst  (TPSL) | Signature - |
| Reviewed By  Name –Rahul Mahapatra  Designation – Chief Manager - Technology  (TPSL) | Signature - |

*.*

Contents

[1. Introduction of “The Single URL” Integration & Installation Guide 4](#_Toc504489349)

[2. What is “Single URL- capability”? 4](#_Toc504489350)

[3. Payment -Transaction Flow 5](#_Toc504489351)

[3.1. Flow Description 6](#_Toc504489352)

[4. Technical Prerequisites 7](#_Toc504489353)

[5. Request Type Integration 9](#_Toc504489358)

[Description of Request Types: 11](#_Toc504489359)

[“Transaction Request Types”: 13](#_Toc504489360)

[5.1. “T” Request Type Integration 13](#_Toc504489361)

[5.2. “TRC” Request Type Integration 16](#_Toc504489362)

[5.3. “TWC” Request Type Integration 19](#_Toc504489363)

[5.4. “TCC” Request Type Integration 21](#_Toc504489364)

[5.5. “TWI” Request Type Integration 24](#_Toc504489365)

[5.6. “TIC” Request Type Integration (IVR SYSTEM) 27](#_Toc504489366)

[5.7. “TIO” Request Type Integration (IVR SYSTEM) 29](#_Toc504489367)

[5.8. “TWD” Request Type Integration (Normal Sale) 31](#_Toc504489368)

[5.9. “TWD” Request Type Integration (Preauth Reserve) 34](#_Toc504489382)

[5.10. “TWD” Request Type Integration (Preauth Consume) 39](#_Toc504489383)

[5.11 “TRS” or “T” Request Type Integration (SI On Card and E-Mandate) 42](#_Toc504489384)

[6. “Verification Request Types” 45](#_Toc504489385)

[6.1. “S” Request Type Integration 45](#_Toc504489386)

[6.2. “O” Request Type Integration 47](#_Toc504489387)

[7. Sample Verification Response for Different Transaction Status For System 51](#_Toc504489388)

[8. “Refund Initiation Request Type” 52](#_Toc504489389)

[8.1 “R” Request Type Integration 52](#_Toc504489390)

[9. Server to Server Payment Posting 54](#_Toc504489391)

[ TPSL PG System will push the transaction response for every successful transaction to the pre-defined merchant URL in secured server to server communication. 54](#_Toc504489392)

[10. List of Parameters 56](#_Toc504489393)

# Introduction of “The Single URL” Integration & Installation Guide

The Payment-Gateway technical integration guide helps you, with the detailed steps you need to follow while installing and integrating “The single URL Feature” with your online platform.

**The guide covers below points:**

* Detailed explanation of integration kit and its business benefits.
* Helps you plan the PG integration, implementation, rollout, testing & Go Live with your Team.
* End to end Payment-Transaction flow is briefly explained functionally & technically for your Technical Team.
* Technical prerequisites are listed for the integration of this feature.
* Sample string format are listed your Tech Team.
* Parameters are well explained for building data exchange logic between PG platform and Merchant’s platform.

# What is “Single URL- capability”?

The Single URL capability provides a **secured** data exchange mechanism between “TechProcess PG Platform” and “Merchant’s online platform over internet.

|  |
| --- |
| **Secured Data Exchange – Figure -1.0** |

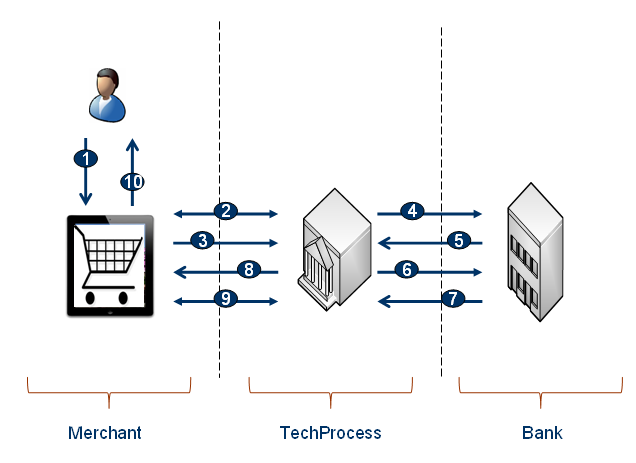
When two Financial Applications are connected, the database connection strings, passwords authenticity and authorization are required careful consideration therefore TechProcess has created a secured data exchange capability, termed as “The Single URL” capability.

The supreme safeguarded methods have been utilized that provides secured environment to exchange sensitive data between “TechProcess PG Platform” and “Merchant’s online platform over internet.

**Other than the Secured Environment features the “Single URL Capability” has following other Benefits also:**

* Merchant not need to configure multiple URLs to integrate with TechProcess PG Platform, as only one URL can technically handle all the request types, that are exchanged during transaction (payment) being made.
* Minimized development & implementation effort at Merchant’s end.
* An encryption technology as **AES encryption** is done for better data security during data exchange between merchant and TPSL system.
* User friendly bank selection page with minimum clicks for the end User.
* The user friendly browser page provides the user experience such as speedy information display, faster browser & faster data loading that minimizes the user log off probability that helps to increase Higher Success Ratio.
* The “Card capturing page” can be integrated with either Merchant’s end, TPSL end or Bank ends as per your requirement.

# Payment -Transaction Flow



**Payment Transaction Flow – Figure -2.0**

# Flow Description

|  |  |
| --- | --- |
| **Steps** | **Description** |
| **Step 1:** | As described in the above **Payment Transaction Flow – Figure -2.0,** Customer comes onto the merchant website, selects the goods/services that he intends to purchase, he adds the goods to the shopping cart. Customer now moves to the checkout process.  *Note: If the Bank selection pages it at merchant end then the Customer will select bank at merchant’s website.*  In this case merchant will send Bank Code in transaction request.  If the bank selection page is at TechProcess end, then merchant code need not be sent in the transaction request. |
| **Step 2:** | Merchant will send request to TechProcess with the payment details. The request parameters are mentioned below in [***List of Parameter***](#Paramerter)***.***  TechProcess will validate the request and store the payment details in system.  TechProcess will generate token number for the request and send URL in response.  Sample URL:  For UAT:-  Web-ServiceURL:- 1. "https://payments.paynimo.com/PaynimoProxy/services/TransactionUATDetails"  Redirection URL:-  <https://www.tekprocess.co.in/PaymentGateway/txnreq.pg?id=03903acc-9e0a-4b9b-ae86-7661a353aa65>  Production:-  Web-Service URL  "https://payments.paynimo.com/PaynimoProxy/services/TransactionLiveDetails"  Redirection URL  https://www.tpsl-india.in/PaymentGateway/txnreq.pg?id=03903acc-9e0a-4b9b-ae86-7661a353aa65 |
| **Step 3:** | On getting of this URL, Merchant will redirect the customer to the received URL.  Customer will get below mentioned option for payment.   * Credit Card * Debit Card * Net Banking * Cash Cards   Customer will select the desired option & bank and proceed. |
| **Step 4:** | TechProcess will redirect the customer to the bank.  Customer will enter his login credentials and process the transaction |
| **Step 5:** | Bank will debit customer’s account and provide status to TechProcess. |
| **Step 6 & 7:** | TechProcess will perform online verification and update the status in system. |
| **Step 8:** | TechProcess will forward response to the merchant with payment details and checksum.  For Corporate Net banking transactions in the response where the maker had done the success transaction it is updated as Initiated and in response 0398 code is sent with desc as Transaction Initiated.  Initiated transaction will get success / Failed post approval from Bank. This status can be pulled using O request type. |
| **Step 9:** | Merchant will validate the checksum and send the request for dual verification. |
| **Step 10:** | Merchant notifies cardholder of transaction response. |

# Technical Prerequisites

|  |  |
| --- | --- |
| For single URL integration Merchant & TPSL needs to ensure below mentioned facilitation:  Java Prerequisites | With technical Prerequisites |

* The Merchant’s web application should be in JAVA Environment.
* Utilization of webservice ShoppincartWSClient.jar(annexure 3.1)
* Error Property file to be shared by TPSL for (annexure 3.2)
* Port to be open at both (Merchant and TPSL) ends
* MID(MerchantCode), this will be shared by TPSL
* key and iv pair for encryption and decryption , this will be shared by TPSL

.Net Prerequisites

* Port to be open at both (Merchant and TPSL) ends
* MID(MerchantCode), this will be shared by TPSL
* key and iv pair for encryption and decryption , this will be shared by TPSL
* DotNetIntegrationKit.DLL
* Error Property file
* Log File
* web.Config

**Installation Steps**

* 1. Need to refer the provided **DLL** by TPSL from integration kit.
  2. Please make sure to use only, the provided **key and iv pair**.
  3. To know the error details, please refer the **error property** file which is part of integration kit.
  4. Write Configurable Values in **Text file** OR **Web.Config** file :
     1. Text File :
        1. Create  **“ConfigFile.txt” inside** “**Configuration” folder.**
        2. Keep“**Configuration” folder inside the project folder and give necessary access to read that file.**
        3. To maintain log information, please create the **“Log”** folder in Configuration folder which is inside the project folder and give necessary access to read/write that file. (which you gave in .txt file)
        4. Keep ErrorMessage.property in Configuration folder
     2. Web.config File :

<add key="ErrorFile" value="ErrorMessage.property"/>

<add key="LogFilePath" value="Log"/>

<add key="TPSLService" value="https://payments.paynimo.com/PaynimoProxy/services/TransactionUATDetails"/>

<add key="IsFixedPath" value="N"/>

<add key="IsCustomLog" value="Y"/>

* 1. IF **IsFixedPath = Y then write whole Path as follows :**

For Example:

|  |
| --- |
| ErrorFile= H:\\Documents\\DATA\\Projects\\PG\_IntegrationSingleURL\_ReadFromTxtFile\\PGSingleURlKit\\Configuration\\ErrorMessage.property LogFilePath=H:\\Documents\\DATA\\Projects\\PG\_IntegrationSingleURL\_ReadFromTxtFile\\PGSingleURlKit\\Configuration\\Log\\  TPSLService=https://payments.paynimo.com/PaynimoProxy/services/TransactionUATDetails  IsFixedPath=Y  IsCustomLog=N |

Else **IsFixedPath = N** then give only folder name and File Name

For Example:

|  |
| --- |
| ErrorFile= ErrorMessage.property  LogFilePath=Log  TPSLService=https://payments.paynimo.com/PaynimoProxy/services/TransactionUATDetails  IsFixedPath=N  IsCustomLog=N |

* 1. If Do not want to print Key,IV in Logs then set **IsCustomLog = Y** otherwise set it as **IsCustomLog = N**

PHP Prerequisites

* Merchant should be consume WebService.
* Port to be open at both (Merchant and TPSL) ends
* MID(MerchantCode), this will be shared by TPSL
* key and iv pair for encryption and decryption , this will be shared by TPSL

Other Technology Prerequisites

* Merchant should be consume WebService.
* Port to be open at both (Merchant and TPSL) ends
* MID(MerchantCode), this will be shared by TPSL
* key and iv pair for encryption and decryption , this will be shared by TPSL



# Request Type Integration

There are several “Request Types” available in the “Single URL Feature” that can be configured by your Technology Team to process the transaction.

|  |
| --- |
|  |

**Secured Data Exchange via Request Type – Figure -2.0**

Your system or online platform can trigger “Request Type” to TPSL PG Platform. On the basis of verification in the backend, the result (Response) will be displayed for the user.

# Description of Request Types:

|  |  |  |
| --- | --- | --- |
| **Request Type** | **Request Type Description** | **Rational** |
| **Transaction Request Types** | | |
| T | Payload is for normal **T**ransaction processing. | Only for normal Type Of Transaction. |
| **TRC** | **T**ransaction & **R**egistration with **C**ard | To be used for transaction processing and registering the card in TPSL Card Vault Service |
| **TWC** | **T**ransaction **W**ith **C**ard | To be used if the card data is captured at merchants end. |
| **TCC** | **T**ransaction with **C**ardID & **C**VV | To be used by the merchant whose customer card is already vaulted and CVV captured at merchant end. |
| **TWI** | **T**ransaction **W**ith **I**MPS | To be used if the IMPS data is captured at merchants end (MMID, Mobile No, OTP). |
| **TIC** | **T**ransaction *for* **I**VR with **C**ard**.** | To be used when merchant wants to make the transaction through IVR. |
| **TIO** | **T**ransaction *for* **I**VR with **O**TP | To be used when merchant wants to make the transaction through IVR along with OTP. |
| **TWD** **(Normal Sale)** | **T**ransaction **W**ith **M**otoBanks | To be used if the card data is captured at merchants end. |
| **TWD (Preauth Reserve)** | **T**ransaction **W**ith **M**otoBanks | To be used if the card data is captured at merchants end. |
| **TWD (Preauth Consume)** | **T**ransaction **W**ith **M**otoBanks | To be used if the card data is captured at merchants end. |
| **TRS** | **T**ransactionwithCA**R**Dfor **SI** | To be used when card and SI details capture is at merchant end |
| **Verification Request Types** | | |
| **S** | Real-time Verification Request | Use it for doing dual verification in real time |
| **O** | **O**ffline Verification Request | Use it for offline verification of the transaction. |
| **Refund Initiation Request Type** | | |
| **R** | **R**efund Initiation Request | To mark Refund against a particular transaction |

# “Transaction Request Types”:

# “T” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“T”** |
| Description | Merchant’s requirement is to allow end customer only pay the payment and not opt for standing instructions or register themselves for SI, neither opt for InstaPay feature where InstaPay allows end customer to pay recurring payment without entering all the card details except CVV number. The business requirement is to only allow customer to make payment where TPSL provides the card capturing page that will be hosted by TPSL. |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | Rqst\_type=T|rqst\_kit\_vrsn=1.0.1|tpsl\_clnt\_cd=T2551|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={email:myname@domain.com}|  rqst\_amnt=12.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/LoginModule/Test.jsp|  s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_120.0\_0.0|  clnt\_dt\_tm=14-03-2014|tpsl\_bank\_cd=470|hash=8cf71e0eb3f7273113b7e73e0db671d8 |
| Sample Response Format | If Request Type=”T” Response  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL925653354|tpsl\_bank\_cd=470|tpsl\_txn\_id=117203405|txn\_amt=2.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{email:sunil.sahu@techproces.co.in}{mob:9856987456}|tpsl\_txn\_time=12-11-2014 11:30:32|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=e6455494-439d-4e12-b333-abc087e869b8|hash=0631abd915a399ab39292bc76a78b1ea8e16bfba |
| List of parameters For Transaction Payload – (Request Type T) | **Conditional**: If Bank selection is at merchant side then provide bank code otherwise should be empty.  The request parameters are mentioned below in [***List of Parameter***](#Paramerter)***.*** |
| For Request Type = “T”: Making a request for online transaction | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “T”, refer to list of parameters of “T” request Type for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:** TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("T");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://payments.paynimo.com/PaynimoProxy/services/TransactionLiveDetails");  objTransactionRequestBean.setCustID("19872627");  objTransactionRequestBean.setStrTPSLTxnID("");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”); this should be in seconds  objTransactionRequestBean.setAccountNo("32567891042"); this should be numeric  If merchant will send account No then he should have send broking bank code in setStrBankCode field.  Call the objTransactionRequestBean.getTransactionToken() method . This method will return  two possible output.   1. Will get the transaction URL with Unique token ID.   **Example:**  https://www.tekprocess.co.in/PaymentGateway/txnreq.pg?id=8bdb7304-e6cd-4b08-8168-aa51c438e42b   1. If any validation fail will get only error code. (Starts with ERROR)   For Corporate Netbanking transactions in the response where the maker had done the success transaction it is updated as Initiated and in response 0398 code is sent with desc as Transaction Initiated.  Initiated transaction will get success / Failed post approval from Bank. This status can be pulled using O request type.  **Sample DotNet Code:**    **Sample PHP Code:** |
| Steps to get response data | Respone = msg=VsLXhgOWN2ZopVLnoncP93KVhgHHE%2BpTb5OHrSIh2p%2BSAlgE9NYwF%2F7sKFDqRh%2FNjAmNXumeJP89%0D%0A2oAjgGotnbyCfGdzYS5EYX1j732fUJWJ6plrboJ0v0QCWHhXBA3UBZ33Nbf3VEjOm8xhxnv8PMnd%0D%0AIVpRf%2BQXrSvLRh%2F%2FueNz6PS49RMHjHJmcW6T5PmJpvAv9jkHVCksTFQqFh9izVNZv43jj4vjdr%2Fz%0D%0A1iOU5%2FU0g0uKpZ0hNqzhDXeR0f1OOgcuJ51XsJdZVk9qGUJkoneH8rRR9WKw6h6QoqYSxfq6NoGb%0D%0AiKSvanTo2HbqPeVfOz%2BOYJEIj5aTm5Ju9kk1t2OxFdcVD8dwT6cnzd6tbBd8xHj6aIwPF2hQ%2BGAu%0D%0AJBwMy9g8Z4MOoVHlIRCWgR0f75yBKUQhPsOBxrM231Yy%2F9iPRAV5y4olNL6pE4wYdosseXaAWznh%0D%0ACp9Il6DA38KAnvnMKmHvSo%2BAU4cLrOd9zdORsU177VhnnwzQBJptOgKxh4o1mmRQavMrUmTOKd2y%0D%0Ac0uZmIi8r2yBCebbJpyC594%3D&tpsl\_mrct\_cd=T2658  After getting the response from PG to do the below mention step.   1. Create the object of TransactionResponseBean Class . 2. Set all the bean value as define below.   beanObj.setIv(iv.getBytes());  beanObj.setKey(key.getBytes());  **In below method merchant needs to set the msg parameter value**  beanObj.setResponsePayload(msg); //set the pg response message    **Call** the beanObj.getResponsePayload() method. This method return the below  mention response.  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL210915453|tpsl\_bank\_cd=470|tpsl\_txn\_id=117203569|txn\_amt=2.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{email:sunil.sahu@techproces.co.in}{mob:9856987456}|tpsl\_txn\_time=12-11-2014 16:58:24|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=c9831306-23e8-4d54-ac18-b7faa8edb73e|hash=d924c4bdc8b8e542453345bb2b719318b26219e9 |

# “TRC” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“TRC”** |
| Description | Merchant’s requirement is to allow End Customer to make payment by entering the card details where card capturing page is provided by the Merchant. Also Merchant wishes to store card information at Merchant’s database to register the card.  Note\*\**The card registration is required for the 1st time payment so that the customer can pay using only CVV number for the subsequent transaction. However for the subsequent transaction the Merchant needs to send “TCC” Request Type.* |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | **Request data format:**  rqst\_type=TRC|rqst\_kit\_vrsn=1.0.3|tpsl\_clnt\_cd=T2658|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|rqst\_amnt=2.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/TWCTest/Responsepayload.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_2.0\_0.0|clnt\_dt\_tm=11-03-2014|tpsl\_bank\_cd=9980|card\_details=fbd440061a1b866333e4756e1137c37ca5b84569919b60d19f6e86f33efaee6511ea060747780f8e1b7e55e2b0e0e69880cfb5e9a2bc593d4f9b019fd9f9ed2a076d9abb9f9f1fcf64631d17974a30b1d3fd01c3ff1fc2918801ad0f0834f8e4336c51d3003ba3067159d9b243a445deaa4f07e60661c66dbf22ac01bf7b2adffee17c979ba01837bfabbc9b3a0b689b6f8b58e705c4375921de09af1a7c0057c03c255c847411a50d6383f951b27f61|hash=033cd6718532a848fae444db26fbba1d8a1d4820 |
| Sample Response Format | **If Request Type=”TRC” Response**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL399698281|tpsl\_bank\_cd=10140|tpsl\_txn\_id=117203564|txn\_amt=2.00  |clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{email:sunil.sahu@techproces.co.in}{mob:9856987456}|  tpsl\_txn\_time=12-11-2014 16:40:54|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=6308eff9-87db-40f1-8a17-0097c4264818|card\_id=4077|  BankTransactionID=114111201389206|alias\_name=VISA\_DBT\_\*\*\*\*9847|hash=d9cf87c453fbc49f32536db6097d9e80cc7d3397 |
| List of parameters For Real Time Transaction verification Payload – (Request Type TRC) | For Transaction Payload – (Request Type TRC)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
| For Transaction & Registration with Card Payload    For Request Type = “TRC” : Make a request to initiate a transaction along with card registration while card capture is at merchant end. | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “TRC”, refer to refer to list of parameters of “TRC” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:** TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TRC");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID("71047882");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrMMID("");  objTransactionRequestBean.setStrOTP("");  objTransactionRequestBean.setCardID("");  objTransactionRequestBean.setCustID("19872627");  objTransactionRequestBean.setCardName("dheeraj");  objTransactionRequestBean.setCardNo("1245142514251425");  objTransactionRequestBean.setCardCVV("125");  objTransactionRequestBean.setCardExpMM("05");  objTransactionRequestBean.setCardExpYY("2014");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  Call the objTransactionRequestBean.getTransactionToken () method.  This method will return  Two possible outputs.   1. Will get the transaction URL with Unique token ID.   **Example:**  https://www.tekprocess.co.in/PaymentGateway/txnreq.pg?id=8bdb7304-e6cd-4b08-8168-aa51c438e42b   1. If any validation fail will get only error code. (Starts with ERROR)   **Sample DotNet Code:**    **Sample PHP Code:** |
| Steps to get response data | Response = msg=VsLXhgOWN2ZopVLnoncP93KVhgHHE%2BpTb5OHrSIh2p%2BSAlgE9NYwF%2F7sKFDqRh%2FNjAmNXumeJP89%0D%0A2oAjgGotnbyCfGdzYS5EYX1j732fUJWJ6plrboJ0v0QCWHhXBA3UBZ33Nbf3VEjOm8xhxnv8PMnd%0D%0AIVpRf%2BQXrSvLRh%2F%2FueNz6PS49RMHjHJmcW6T5PmJpvAv9jkHVCksTFQqFh9izVNZv43jj4vjdr%2Fz%0D%0A1iOU5%2FU0g0uKpZ0hNqzhDXeR0f1OOgcuJ51XsJdZVk9qGUJkoneH8rRR9WKw6h6QoqYSxfq6NoGb%0D%0AiKSvanTo2HbqPeVfOz%2BOYJEIj5aTm5Ju9kk1t2OxFdcVD8dwT6cnzd6tbBd8xHj6aIwPF2hQ%2BGAu%0D%0AJBwMy9g8Z4MOoVHlIRCWgR0f75yBKUQhPsOBxrM231Yy%2F9iPRAV5y4olNL6pE4wYdosseXaAWznh%0D%0ACp9Il6DA38KAnvnMKmHvSo%2BAU4cLrOd9zdORsU177VhnnwzQBJptOgKxh4o1mmRQavMrUmTOKd2y%0D%0Ac0uZmIi8r2yBCebbJpyC594%3D&tpsl\_mrct\_cd=T2658  After getting the response from PG to do the below mention step.   1. Create the object of TransactionResponseBean Class. 2. Set all the bean value as define below.   beanObj.setIv(iv.getBytes());  beanObj.setKey(key.getBytes());  **In below method merchant needs to set the msg parameter value**  beanObj.setResponsePayload(msg); //set the pg response message  **Call** the beanObj.getResponsePayload() method. This method return the below  mention response.  txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=7421002|tpsl\_bank\_cd=470|tpsl\_txn\_id=71047298|txn\_amt=100.00|tpsl\_txn\_time=05-03-2014 23:13:12|card\_ID=12345|Alias\_name=VISA\_DBT\_\*\*\*\*\_1234|BankTransactionId =1234567|rqst\_token= |

# “TWC” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“TWC”** |
| Description | Merchant’s requirement is to allow End Customer to make payment by entering the card details where card capturing page is provided by the Merchant. |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | **Request data format**:  rqst\_type=TWC|rqst\_kit\_vrsn=1.0.3|tpsl\_clnt\_cd=T2658|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|rqst\_amnt=2.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/TWCTest/Responsepayload.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_2.0\_0.0|clnt\_dt\_tm=11-03-2014|tpsl\_bank\_cd=9980|card\_details=fbd440061a1b866333e4756e1137c37ca5b84569919b60d19f6e86f33efaee6511ea060747780f8e1b7e55e2b0e0e69880cfb5e9a2bc593d4f9b019fd9f9ed2a076d9abb9f9f1fcf64631d17974a30b1d3fd01c3ff1fc2918801ad0f0834f8e4336c51d3003ba3067159d9b243a445deaa4f07e60661c66dbf22ac01bf7b2adffee17c979ba01837bfabbc9b3a0b689b6f8b58e705c4375921de09af1a7c0057c03c255c847411a50d6383f951b27f61|hash=033cd6718532a848fae444db26fbba1d8a1d4820 |
| Sample Response Format | **If Request Type=”TWC” Response**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068 |
| List of parameters For Real Time Transaction verification Payload – (Request Type TWC) | For Transaction Payload – (Request Type TWC)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TWC”, refer to refer to list of parameters of “TWC” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TWC");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“”);  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  objTransactionRequestBean.setCardNo(“11111323213121111”);  objTransactionRequestBean.setCardName(“johney boss”);  objTransactionRequestBean.setCardExpMM(“11”);  objTransactionRequestBean.setCardExpYY(“2017”);  objTransactionRequestBean.setCardCVV(“111”);  **Note:** Use key and iv shared by TPSL.  **Sample DotNet Code:**    **Sample PHP Code:** |
| **Steps to get response data** | **Response = msg=VsLXhgOWN2ZopVLnoncP93KVhgHHE%2BpTb5OHrSIh2p%2BSAlgE9NYwF%2F7sKFDqRh%2FNjAmNXumeJP89%0D%0A2oAjgGotnbyCfGdzYS5EYX1j732fUJWJ6plrboJ0v0QCWHhXBA3UBZ33Nbf3VEjOm8xhxnv8PMnd%0D%0AIVpRf%2BQXrSvLRh%2F%2FueNz6PS49RMHjHJmcW6T5PmJpvAv9jkHVCksTFQqFh9izVNZv43jj4vjdr%2Fz%0D%0A1iOU5%2FU0g0uKpZ0hNqzhDXeR0f1OOgcuJ51XsJdZVk9qGUJkoneH8rRR9WKw6h6QoqYSxfq6NoGb%0D%0AiKSvanTo2HbqPeVfOz%2BOYJEIj5aTm5Ju9kk1t2OxFdcVD8dwT6cnzd6tbBd8xHj6aIwPF2hQ%2BGAu%0D%0AJBwMy9g8Z4MOoVHlIRCWgR0f75yBKUQhPsOBxrM231Yy%2F9iPRAV5y4olNL6pE4wYdosseXaAWznh%0D%0ACp9Il6DA38KAnvnMKmHvSo%2BAU4cLrOd9zdORsU177VhnnwzQBJptOgKxh4o1mmRQavMrUmTOKd2y%0D%0Ac0uZmIi8r2yBCebbJpyC594%3D&tpsl\_mrct\_cd=T2658**  **After getting the response from PG to do the below mention step.**   1. **Create the object of TransactionResponseBean Class.** 2. **Set all the bean value as define below.**   **beanObj.setIv(iv.getBytes());**  **beanObj.setKey(key.getBytes());**  **In below method merchant needs to set the msg parameter value**  **beanObj.setResponsePayload(msg); //set the pg response message**  **Call the beanObj.getResponsePayload() method.**  **This method return the below**  **mention response.**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8mar2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068 |

# “TCC” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“TCC”** |
| Description | Assuming the card is registered using “TRC” request type at merchant’s end, now the Merchant’s requirement is to allow End Customer to make payment by entering the only CVV number against the register card to pay subsequent payment.  Note\*\**The CVV will be captured on Merchant’s portal.* |
| Event /Trigger of Request Type |  |
| Sample Request Format | **Request Format: Payload generation at Merchant End:**  Final Payload to be sent to token generation system with ~(Tilda) appended and MID added at the end :-    encAes(rqst\_type=TCC|rqst\_kit\_vrsn=1.0.1|tpsl\_clnt\_cd=T2551|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={email:myname@domain.com}|  rqst\_amnt=12.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/LoginModule/Test.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_120.0\_0.0~Test\_120.0\_0.0|clnt\_dt\_tm=14-03-2014|tpsl\_bank\_cd=470|card\_details= 2dd03a3c9fbfb552f139c9801dace20880391cbf3e5002978fbf05753074ab1c837279863e17dc4fd4448c2eeb61eaf6|Card\_ID=123456|T2856~**encAes =AES Encryption**    Note: The encrypted response will have merchant code appended at the end. |
| Sample Response Format | **If Request Type=”TCC” Response**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1755422387|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228582|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{mob:9856987456}|tpsl\_txn\_time=14-11-2014 09:38:57|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=65fe6900-db52-4635-a584-3065dbcbf09b|card\_id=1108|BankTransactionID=2014112574211|hash=d7baacbfe190ec26e6743005c6750691b4c88a3f |
| List of parameters For Real Time Transaction verification Payload – (Request Type TCC) | For Transaction Payload – (Request Type TCC)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
| For Transaction with CardID & CVV Payload    For Request Type = “TCC”: Make a request to initiate transaction against Card ID. | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “TCC”, refer to refer to list of parameters of “TCC” for setting parameter values in TransactionRequestBean.**  Call the objTransactionRequestBean.getTransactionToken() method. This method return the below mention response.  **Sample Java Code:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TCC");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“”);  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setCardID("1234");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  objTransactionRequestBean.setCardNo(“”);  objTransactionRequestBean.setCardName(“”);  objTransactionRequestBean.setCardExpMM(“”);  objTransactionRequestBean.setCardExpYY(“”);  objTransactionRequestBean.setCardCVV(“111”); if available then send  **Note:** Use key and iv shared by TPSL.  **Example :**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1755422387|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228582|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{mob:9856987456}|tpsl\_txn\_time=14-11-2014 09:38:57|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=65fe6900-db52-4635-a584-3065dbcbf09b|card\_id=1108|BankTransactionID=2014112574211|hash=d7baacbfe190ec26e6743005c6750691b4c88a3f  **Sample DotNet Code:**    **Sample PHP Code:** |
| **Steps to get response data** | **Response = msg=VsLXhgOWN2ZopVLnoncP93KVhgHHE%2BpTb5OHrSIh2p%2BSAlgE9NYwF%2F7sKFDqRh%2FNjAmNXumeJP89%0D%0A2oAjgGotnbyCfGdzYS5EYX1j732fUJWJ6plrboJ0v0QCWHhXBA3UBZ33Nbf3VEjOm8xhxnv8PMnd%0D%0AIVpRf%2BQXrSvLRh%2F%2FueNz6PS49RMHjHJmcW6T5PmJpvAv9jkHVCksTFQqFh9izVNZv43jj4vjdr%2Fz%0D%0A1iOU5%2FU0g0uKpZ0hNqzhDXeR0f1OOgcuJ51XsJdZVk9qGUJkoneH8rRR9WKw6h6QoqYSxfq6NoGb%0D%0AiKSvanTo2HbqPeVfOz%2BOYJEIj5aTm5Ju9kk1t2OxFdcVD8dwT6cnzd6tbBd8xHj6aIwPF2hQ%2BGAu%0D%0AJBwMy9g8Z4MOoVHlIRCWgR0f75yBKUQhPsOBxrM231Yy%2F9iPRAV5y4olNL6pE4wYdosseXaAWznh%0D%0ACp9Il6DA38KAnvnMKmHvSo%2BAU4cLrOd9zdORsU177VhnnwzQBJptOgKxh4o1mmRQavMrUmTOKd2y%0D%0Ac0uZmIi8r2yBCebbJpyC594%3D&tpsl\_mrct\_cd=T2658**  **After getting the response from PG to do the below mention step.**   1. **Create the object of TransactionResponseBean Class.** 2. **Set all the bean value as define below.**   **beanObj.setIv(iv.getBytes());**  **beanObj.setKey(key.getBytes());**  **In below method merchant needs to set the msg parameter value**  **beanObj.setResponsePayload(msg); //set the pg response message**  **Call the beanObj.getResponsePayload() method.**  **This method return the below**  **mention response.**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1755422387|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228582|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{mob:9856987456}|tpsl\_txn\_time=14-11-2014 09:38:57|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=65fe6900-db52-4635-a584-3065dbcbf09b|card\_id=1108|BankTransactionID=2014112574211|hash=d7baacbfe190ec26e6743005c6750691b4c88a3f |

# “TWI” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“TWI”** |
| Description | Merchant’s requirement is to allow end customer to make a payment using IMPS as a payment option. End customer will enter IMPS details such as MMID, Mobile number & OTP at Merchant’s website |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | Request data format:  rqst\_type=TWI|rqst\_kit\_vrsn=1.0.3|tpsl\_clnt\_cd=T2658|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|rqst\_amnt=2.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/TWITest/Responsepayload.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_2.0\_0.0|clnt\_dt\_tm=11-03-2014|tpsl\_bank\_cd=9980|imps\_details=fbd440061a1b866333e4756e1137c37ca5b84569919b60d19f6e86f33efaee6511ea060747780f8e1b7e55e2b0e0e69880cfb5e9a2bc593d4f9b019fd9f9ed2a076d9abb9f9f1fcf64631d17974a30b1d3fd01c3ff1fc2918801ad0f0834f8e4336c51d3003ba3067159d9b243a445deaa4f07e60661c66dbf22ac01bf7b2adffee17c979ba01837bfabbc9b3a0b689b6f8b58e705c4375921de09af1a7c0057c03c255c847411a50d6383f951b27f61|hash=033cd6718532a848fae444db26fbba1d8a1d4820 |
| Sample Response Format | **Sample Response**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=7421002|tpsl\_bank\_cd=470|tpsl\_txn\_id=71047298|txn\_amt=100.00|tpsl\_txn\_time=05-03-2014 23:13:12|BankTransactionID=12345| sha1-hash |
| List of parameters For Real Time Transaction verification Payload – (Request Type TWI) | The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TWI”, refer to refer to list of parameters of “TWI” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TWI");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“”);  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setStrMMID("1231221");  objTransactionRequestBean.setStrOTP("123123");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds    **Note:** Use key and iv shared by TPSL.  Call the objTransactionRequestBean.getTransactionToken() method.  This method will return two possible output.  1 If any validation fail will get only error code. (Starts with ERROR)  2 txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=7421002|tpsl\_bank\_cd=470|tpsl\_txn\_id=71047298|txn\_amt=100.00|tpsl\_txn\_time=05-03-2014 23:13:12|BankTransactionID=12345| sha1-hash  **Sample DotNet Code:**    **Sample PHP Code:** |
| **Steps to get response data** | **Response = msg=VsLXhgOWN2ZopVLnoncP93KVhgHHE%2BpTb5OHrSIh2p%2BSAlgE9NYwF%2F7sKFDqRh%2FNjAmNXumeJP89%0D%0A2oAjgGotnbyCfGdzYS5EYX1j732fUJWJ6plrboJ0v0QCWHhXBA3UBZ33Nbf3VEjOm8xhxnv8PMnd%0D%0AIVpRf%2BQXrSvLRh%2F%2FueNz6PS49RMHjHJmcW6T5PmJpvAv9jkHVCksTFQqFh9izVNZv43jj4vjdr%2Fz%0D%0A1iOU5%2FU0g0uKpZ0hNqzhDXeR0f1OOgcuJ51XsJdZVk9qGUJkoneH8rRR9WKw6h6QoqYSxfq6NoGb%0D%0AiKSvanTo2HbqPeVfOz%2BOYJEIj5aTm5Ju9kk1t2OxFdcVD8dwT6cnzd6tbBd8xHj6aIwPF2hQ%2BGAu%0D%0AJBwMy9g8Z4MOoVHlIRCWgR0f75yBKUQhPsOBxrM231Yy%2F9iPRAV5y4olNL6pE4wYdosseXaAWznh%0D%0ACp9Il6DA38KAnvnMKmHvSo%2BAU4cLrOd9zdORsU177VhnnwzQBJptOgKxh4o1mmRQavMrUmTOKd2y%0D%0Ac0uZmIi8r2yBCebbJpyC594%3D&tpsl\_mrct\_cd=T2658**  **After getting the response from PG to do the below mention step.**   1. **Create the object of TransactionResponseBean Class.** 2. **Set all the bean value as define below.**   **beanObj.setIv(iv.getBytes());**  **beanObj.setKey(key.getBytes());**  **In below method merchant needs to set the msg parameter value**  **beanObj.setResponsePayload(msg); //set the pg response message**  **Call the beanObj.getResponsePayload() method.**  **This method return the below**  **mention response.**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=7421002|tpsl\_bank\_cd=470|tpsl\_txn\_id=71047298|txn\_amt=100.00|tpsl\_txn\_time=05-03-2014 23:13:12|BankTransactionID=12345| sha1-hash |

# “TIC” Request Type Integration (IVR SYSTEM)

|  |  |
| --- | --- |
| Request Type | **“TIC”** |
| Description | Merchant’s requirement is to allow End Customer to make payment through IVR System. |
| Event /Trigger of Request Type | On customer request for payment Merchant’s IVR system will initiate this request |
| Sample Request Format | **Request data format**:  rqst\_type=TIC|rqst\_kit\_vrsn=1.0.3|tpsl\_clnt\_cd=T2658|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|rqst\_amnt=2.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/TWCTest/Responsepayload.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_2.0\_0.0|clnt\_dt\_tm=11-03-2014|tpsl\_bank\_cd=9980|card\_details=fbd440061a1b866333e4756e1137c37ca5b84569919b60d19f6e86f33efaee6511ea060747780f8e1b7e55e2b0e0e69880cfb5e9a2bc593d4f9b019fd9f9ed2a076d9abb9f9f1fcf64631d17974a30b1d3fd01c3ff1fc2918801ad0f0834f8e4336c51d3003ba3067159d9b243a445deaa4f07e60661c66dbf22ac01bf7b2adffee17c979ba01837bfabbc9b3a0b689b6f8b58e705c4375921de09af1a7c0057c03c255c847411a50d6383f951b27f61|hash=033cd6718532a848fae444db26fbba1d8a1d4820 |
| Sample Response Format | **If Request Type=”TIC” Response**  token  Example C495B3E7-79A0-413C-B82A-1632920CB244  For any error it will be starts with ERROR |
| **Details of TIC** | For Transaction Payload – (Request Type TIC)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TIC”, refer to refer to list of parameters of “TIC” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:**  sTransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TIC");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“”);  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setCustID(“1234”);  objTransactionRequestBean.setCardNo(“1111123123121111”);  objTransactionRequestBean.setCardName(“Rajeev Narayanan”);  objTransactionRequestBean.setCardExpMM(“11”);  objTransactionRequestBean.setCardExpYY(“2017”);  objTransactionRequestBean.setCardCVV(“111”);  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  objTransactionRequestBean.setAccountNo("32567891042"); this should be numeric  **Note:** Use key and iv shared by TPSL.  Call the objTransactionRequestBean.getTransactionToken() method.  This method will return  two possible output.   1. Will get the transaction URL with Unique token ID.   **Example:**  C495B3E7-79A0-413C-B82A-1632920CB244   1. If any validation fail will get only error code. (Starts with ERROR)   **Sample DotNet Code:**  **Sample PHP Code:** |

# “TIO” Request Type Integration (IVR SYSTEM)

|  |  |
| --- | --- |
| Request Type | **“TIO”** |
| Description | Merchant’s requirement is to allow End Customer to make payment through IVR System. |
| Event /Trigger of Request Type | On customer request for payment Merchant’s IVR system will initiate this request with OTP for authorization. |
| Sample Request Format | **Request data format**:  rqst\_type=TIO|rqst\_kit\_vrsn=1.0.6|tpsl\_clnt\_cd=T2490|tpsl\_txn\_id=9d453ae0-c711-4115-9ce6-37b9472cc1f2|otp=123123|hash=109ab1d64aab489af4f553532bcc71b7ca4dbd8c |
| Sample Response Format | **If Request Type=”TIO” Response**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=TXN0001|tpsl\_bank\_cd=10590|tpsl\_txn\_id=228501|txn\_amt=1.00|clnt\_rqst\_meta={itc:email:myname@domain.com}{email:myname@domain.com}{mob:9888888888}{custname:xyz}|tpsl\_txn\_time=10-11-2014 09:32:54|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=8a5719f2-d8d6-4b6b-8f21-b82b2d179958|alias\_name=NA|card\_id=|BankTransactionID=20141110093200054228501|hash=e2fa11edc451d979c7c03f665b894d0b69fca107 |
| **Details of TIC** | For Transaction Payload – (Request Type TIO)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TIO”, refer to refer to list of parameters of “TIC” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TIO");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(token);  objTransactionRequestBean.setOtp(“123456”)  objTransactionRequestBean.setStrMerchantCode("T2490");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  **Note:** Use key and iv shared by TPSL.  **Sample DotNet Code:**  **Sample PHP Code:** |
| Steps to get response data | Response = msg=VsLXhgOWN2ZopVLnoncP93KVhgHHE%2BpTb5OHrSIh2p%2BSAlgE9NYwF%2F7sKFDqRh%2FNjAmNXumeJP89%0D%0A2oAjgGotnbyCfGdzYS5EYX1j732fUJWJ6plrboJ0v0QCWHhXBA3UBZ33Nbf3VEjOm8xhxnv8PMnd%0D%0AIVpRf%2BQXrSvLRh%2F%2FueNz6PS49RMHjHJmcW6T5PmJpvAv9jkHVCksTFQqFh9izVNZv43jj4vjdr%2Fz%0D%0A1iOU5%2FU0g0uKpZ0hNqzhDXeR0f1OOgcuJ51XsJdZVk9qGUJkoneH8rRR9WKw6h6QoqYSxfq6NoGb%0D%0AiKSvanTo2HbqPeVfOz%2BOYJEIj5aTm5Ju9kk1t2OxFdcVD8dwT6cnzd6tbBd8xHj6aIwPF2hQ%2BGAu%0D%0AJBwMy9g8Z4MOoVHlIRCWgR0f75yBKUQhPsOBxrM231Yy%2F9iPRAV5y4olNL6pE4wYdosseXaAWznh%0D%0ACp9Il6DA38KAnvnMKmHvSo%2BAU4cLrOd9zdORsU177VhnnwzQBJptOgKxh4o1mmRQavMrUmTOKd2y%0D%0Ac0uZmIi8r2yBCebbJpyC594%3D&tpsl\_mrct\_cd=T2658  After getting the response from PG to do the below mention step.   1. Create the object of TransactionResponseBean Class . 2. Set all the bean value as define below.   beanObj.setIv(iv.getBytes());  beanObj.setKey(key.getBytes());  In below method merchant needs to set the msg parameter value  beanObj.setResponsePayload(msg); //set the pg response message  **Call** the beanObj.getResponsePayload() method. This method return the below  mention response.  txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=TXN0001|tpsl\_bank\_cd=10590|tpsl\_txn\_id=228501|txn\_amt=1.00|clnt\_rqst\_meta={itc:email:myname@domain.com}{email:myname@domain.com}{mob:9820596973}{custname:xyz}|tpsl\_txn\_time=10-11-2014 09:32:54|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=8a5719f2-d8d6-4b6b-8f21-b82b2d179958|alias\_name=NA|card\_id=|BankTransactionID=20141110093200054228501|hash=e2fa11edc451d979c7c03f665b894d0b69fca107 |

# “TWD” Request Type Integration (Normal Sale)

|  |  |
| --- | --- |
| Request Type | **“TWD”** |
| Description | Merchant’s requirement is to allow End Customer to make payment by entering the card details where card capturing page is provided by the Merchant. Its optional to set TxnType as SALE. |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | **Request data format**:  rqst\_type=TWD|rqst\_kit\_vrsn=1.0.9|tpsl\_clnt\_cd=T2490|clnt\_txn\_ref=TXN0002|clnt\_rqst\_meta={itc:email:myname@domain.com}{email:myname@domain.com}{mob:9820596973}{custname:xyz}|rqst\_amnt=1.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp |rqst\_rqst\_dtls=Test\_1.0\_0.0|clnt\_dt\_tm=12-01-2015|tpsl\_bank\_cd=9980|mob=9820596973|TxnType=SALE|card\_details=cfa568b4214ab00e919c0feec4195dc63ac75cdb838aaee296dcbfca331e33b5a260a15ca576ff46dd01bfd4e5689f33cf0fb13f2632592c3fe7178f3b60fe922f6f02698870f0ed46ea7a6a72b6dd1952714f63edabf0b02ce42f8f94ffa26a6a122e770802bb0857d8cff47df01d868d882d5190cf031e5004891686c8e8eb3918fbc5a26e70438e2fbd185bd28e38aed82325c46db2b681b5bc496871f2cfe5b7c96de774ffbc99c5795acb02ced8|hash=148655c31eb147a7f5b73d893fb209c9dd4ccb40 |
| Sample Response Format | **If Request Type=”TWD” Response**  **Server to Server Response Send to Merchant**  token=201f5da7-4c09-4362-8216-0a763851ffd6|url=aHR0cHM6Ly90cHNsdmtzcnY2MDQ2L1BheW1lbnRHYXRld2F5L1Rlc3RTQklNb3RvLmpzcA==|method=Post|form=sbim|params=cGFufjQ2MjI3MzU0NTA2MTE2MzgsZXhwaXJ5fjE3MDMsbWVyY2hhbnRJRH5OQSxjdXJyZW5jeVZh  **Final Response Send to Merchant**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068 |
| List of parameters For Real Time Transaction verification Payload – (Request Type TWD) | For Transaction Payload – (Request Type TWD)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TWD”, refer to refer to list of parameters of “TWD” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TWD");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("19-01-2015");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“”);  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  objTransactionRequestBean.setCardNo(“11111323213121111”);  objTransactionRequestBean.setCardName(“johney boss”);  objTransactionRequestBean.setCardExpMM(“11”);  objTransactionRequestBean.setCardExpYY(“2017”);  objTransactionRequestBean.setCardCVV(“111”);  objTransactionRequestBean.setStrTxnType("SALE");  objTransactionRequestBean.setStrTxnSubType("");  String redirectionURL=objTransactionRequestBean.getTransactionToken();  if(redirectionURL.startsWith("ERR")||redirectionURL.startsWith("EX")|| redirectionURL.startsWith("RE")){  //**Transaction processing is unsuccessful**  } else{  java.util.Map map=new java.util.HashMap();  String strRetFormData="";  String issuerAcsUrl="";  String strRespData[]=redirectionURL.split("\\|");  for(String responseParam: strRespData){  String param[]=responseParam.split("\\=");  map.put(param[0], param[1]);  }  issuerAcsUrl = new String(com.tp.pg.util.CryptoLibrary.decrypt((String)map.get("url")));  strRetFormData=new String(com.tp.pg.util.CryptoLibrary.decrypt((String)map.get("params")));  }  //Merchant will use this data to create the form and submit.  //Web Form creation.  <html>  <body>  <form action="<%=issuerAcsUrl%>" method="post">  <%  String strFormData[]=strRetFormData.split("\\,");  for(String strData : strFormData){  String strDataArr[]=strData.split("\\~");  if(strDataArr.length==2){  %>  <input type="hidden" name="<%=strDataArr[0]%>" value="<%=strDataArr[1]%>">  <% } %>  <input type="submit" name="submit" value="GO">  </form>  </body>  </html>  When user click on submit button bank 3D page will be shown. On successful authentication final response send to merchant after decrypting will be :-  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068  **Note:** Use key and iv shared by TPSL.  **Sample DotNet Code:**    **Sample PHP Code:** |
|  |  |



# “TWD” Request Type Integration (Preauth Reserve)

|  |  |
| --- | --- |
| Request Type | **“TWD”** |
| Description | Merchant’s requirement is to allow End Customer to make payment using Preauth Mode. For this merchant will send card details to PG alongwith TxnType as PREAUTH and TxnSubType as RESERVE. |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | **Request data format**:  rqst\_type=TWD|rqst\_kit\_vrsn=1.0.9|tpsl\_clnt\_cd=T2490|clnt\_txn\_ref=TXN0002|clnt\_rqst\_meta={itc:email:myname@domain.com}{email:myname@domain.com}{mob:9820596973}{custname:xyz}|rqst\_amnt=1.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp |rqst\_rqst\_dtls=Test\_1.0\_0.0|clnt\_dt\_tm=12-01-2015|tpsl\_bank\_cd=9980|mob=9820596973|TxnType=PREAUTH|TxnSubType=RESERVE|card\_details=cfa568b4214ab00e919c0feec4195dc63ac75cdb838aaee296dcbfca331e33b5a260a15ca576ff46dd01bfd4e5689f33cf0fb13f2632592c3fe7178f3b60fe922f6f02698870f0ed46ea7a6a72b6dd1952714f63edabf0b02ce42f8f94ffa26a6a122e770802bb0857d8cff47df01d868d882d5190cf031e5004891686c8e8eb3918fbc5a26e70438e2fbd185bd28e38aed82325c46db2b681b5bc496871f2cfe5b7c96de774ffbc99c5795acb02ced8|hash=7c4092f2ed3d874ce4ce8dc732054a69884c5196 |
| Sample Response Format | **If Request Type=”TWD” Response**  **Server to Server Response Send to Merchant**  token=201f5da7-4c09-4362-8216-0a763851ffd6|url=aHR0cHM6Ly90cHNsdmtzcnY2MDQ2L1BheW1lbnRHYXRld2F5L1Rlc3RTQklNb3RvLmpzcA==|method=Post|form=sbim|params=cGFufjQ2MjI3MzU0NTA2MTE2MzgsZXhwaXJ5fjE3MDMsbWVyY2hhbnRJRH5OQSxjdXJyZW5jeVZh  **Final Response Send to Merchant**  For Successful transaction:-  txn\_status=0200|txn\_msg=PREAUTH\_SUCCESS|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068  For Unsuccessful transaction:-  txn\_status=0399|txn\_msg=Transaction Failed| txn\_err\_msg = ERROR010\_Bank\_Authentication\_Failed|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068 |
| List of parameters For Real Time Transaction verification Payload – (Request Type TWD) | For Transaction Payload – (Request Type TWD)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TWD”, refer to refer to list of parameters of “TWD” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TWD");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("1.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_1.0\_0.0");  objTransactionRequestBean.setTxnDate("19-01-2015");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“”);  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  objTransactionRequestBean.setCardNo(“11111323213121111”);  objTransactionRequestBean.setCardName(“johney boss”);  objTransactionRequestBean.setCardExpMM(“11”);  objTransactionRequestBean.setCardExpYY(“2017”);  objTransactionRequestBean.setCardCVV(“111”);  objTransactionRequestBean.setStrTxnType("PREAUTH");  objTransactionRequestBean.setStrTxnSubType("RESERVE");  String redirectionURL=objTransactionRequestBean.getTransactionToken();  if(redirectionURL.startsWith("ERR")||redirectionURL.startsWith("EX")|| redirectionURL.startsWith("RE")){  //**Transaction processing is unsuccessful**  } else{  java.util.Map map=new java.util.HashMap();  String strRetFormData="";  String issuerAcsUrl="";  String strRespData[]=redirectionURL.split("\\|");  for(String responseParam: strRespData){  String param[]=responseParam.split("\\=");  map.put(param[0], param[1]);  }  issuerAcsUrl = new String(com.tp.pg.util.CryptoLibrary.decrypt((String)map.get("url")));  strRetFormData=new String(com.tp.pg.util.CryptoLibrary.decrypt((String)map.get("params")));  }  //Merchant will use this data to create the form and submit.  //Web Form creation.  <html>  <body>  <form action="<%=issuerAcsUrl%>" method="post">  <%  String strFormData[]=strRetFormData.split("\\,");  for(String strData : strFormData){  String strDataArr[]=strData.split("\\~");  if(strDataArr.length==2){  %>  <input type="hidden" name="<%=strDataArr[0]%>" value="<%=strDataArr[1]%>">  <% } %>  <input type="submit" name="submit" value="GO">  </form>  </body>  </html>  When user click on submit button bank 3D page will be shown. On successful authentication final response send to merchant after decrypting will be :-  txn\_status=0200|txn\_msg=PREAUTH\_SUCCESS|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068  In case of unsuccessful authentication final response send to merchant after decrypting will be :-  txn\_status=0399|txn\_msg=Transaction Faile|txn\_err\_msg=ERROR010\_Bank\_Authetication\_Failed|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068  **Note:** Use key and iv shared by TPSL.  **Sample DotNet Code:**    **Sample PHP Code:** |
|  |  |

# “TWD” Request Type Integration (Preauth Consume)

|  |  |
| --- | --- |
| Request Type | **“TWD”** |
| Description | To process Preauth Consume transaction, Merchant need to set TxnType as PREAUTH, TxnSubType as CONSUME and tpsl\_tran\_id of PREAUTH RESERVE transaction. Set amount field which need to debited from customer account. |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | **Request data format**:  rqst\_type=TWD|rqst\_kit\_vrsn=1.0.9|tpsl\_clnt\_cd=T2490|clnt\_txn\_ref=TXN0002|clnt\_rqst\_meta={itc:email:myname@domain.com}{email:myname@domain.com}{mob:9820596973}{custname:xyz}|rqst\_amnt=1.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp |rqst\_rqst\_dtls=Test\_1.0\_0.0|clnt\_dt\_tm=12-01-2015|tpsl\_bank\_cd=9980|tpsl\_txn\_id=229730|mob=9820596973|TxnType=PREAUTH|TxnSubType=CONSUME|card\_details=77dcb87a055094907c29d38ea61562984b42070c22ce4b11a9298dccf515e57e3953c8d85f08613a8627002853a33c4b27d1cd5299cb7a360b3ff3e0e582510397dd6ce2104f0f4d56b1a95f2c4a4da73e32d98c2c6ef2d18a09a5e557db8907551760f58affbe68af541066abda0bed|hash=936c995add061dd3c4823cfb650daca03d676772 |
| Sample Response Format | **If Request Type=”TWD” Response**  For Successful transaction:-  txn\_status=0300|txn\_msg=TRANSACTION SUCCESSFUL|txn\_err\_msg=NA|clnt\_txn\_ref= TPSL909190284|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a69a7a4e-b494-4eae-8827-4d95093745e8|hash=77f7303dc01b421615f66e60bf126f6d06c03068  For Unsuccessful transaction:-  txn\_status=0299|txn\_msg=TRANSACTION FAILED|txn\_err\_msg=ERROR0112\_Bank\_Authentication\_Failed|clnt\_txn\_ref=TPSL909190284|tpsl\_bank\_cd=9980|tpsl\_txn\_id=229743|txn\_amt=1.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}|tpsl\_txn\_time=19-01-2015 15:38:16|tpsl\_rfnd\_id=NA|bal\_amt=NA|BankTransactionID=TPSL1253761541|rqst\_token=5d282939-ebd8-451f-9e47-e0fa066a4ae1|hash=8733b1f647520884621ffbcd5b5c6bccd4954a2e |
| List of parameters For Real Time Transaction verification Payload – (Request Type TWD) | For Transaction Payload – (Request Type TWD)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **If Request Type is “TWD”, refer to refer to list of parameters of “TWD” for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TWD");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("1.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_1.0\_0.0");  objTransactionRequestBean.setTxnDate("19-01-2015");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID(“**229730**”);  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  objTransactionRequestBean.setStrTxnType("PREAUTH");  objTransactionRequestBean.setStrTxnSubType("CONSUME");  String redirectionURL=objTransactionRequestBean.getTransactionToken();  if(redirectionURL.startsWith("ERR")||redirectionURL.startsWith("EX")|| redirectionURL.startsWith("RE")){  //**Transaction processing is unsuccessful**  } else{  TransactionResponseBean objTxnResp= TransactionResponseBean();  String[] reqParam = redirectionURL.split("&");  String msgParam = reqParam[0];  String strRes = msgParam.substring(4,msgParam.length());  objTxnResp.setResponsePayload(strRes);  objTxnResp.setKey(key.getBytes());  objTxnResp.setIv(iv.getBytes());  String output = objTxnResp.getResponsePayload());  }  **Note:** Use key and iv shared by TPSL.  **Sample DotNet Code:**    **Sample PHP Code:** |
|  |  |

# 5.11 “TRS” or “T” Request Type Integration (SI On Card and E-Mandate)

|  |  |
| --- | --- |
| Request Type | **“TRS”** |
| Description | If merchant has opted out for SI transactions, he needs to send parameters as below.  Here request type TRS. TRS request type can be used in case of SI transactions with cards. For SI transactions with cards, merchant can send Card details with TRS request type.  If the Card details cannot be sent by the merchant then merchant has to send request type as “T” with only mandate details and customer details. Card details need not be sent.  In case, merchant want to transact SI transaction using cards, he can not use TWC request type.  Additionally, for TRS or T for SI bank selection should be at Merchant end. i.e. merchant should send bank code in request.  Merchant need to send his existing clnt\_rqst\_meta parameter. Append parameters highlighted in yellow text using tild (~) character.  For TRS or T transaction of SI, merchant will get **mandate\_reg\_no** in response as a reference of SI transaction.  Sample clnt\_rqst\_meta Format : mandateData{amount\_type:M~frequency:ADHO~account\_number:914010009305862~ ifsc\_code:KKBK0001360~expiry\_date:02-02-2018~amount:5000~identifier:12345~schedule\_date:25-03-2017}  Description for additional SRC\_ITC parameter :   |  |  |  | | --- | --- | --- | | **Parameter Name** | **Possible Value** | **Required / Optional** | | amount\_type | M (Variable) / F (Fixed) | Required | | frequency | DAIL (Daily)  Week (Weekly)  MNTH (Monthly)  QURT (Quarterly)  MIAN (Semi Annually)  YEAR (Yearly)  BIMN (Bi- Monthly)  ADHO (As and when presented) | Required | | account\_number | Valid account number | Required only in case of SI / E Mandate transaction using Net Banking. | | ifsc\_code | Valid IFSC Code | Required only in case of SI / E Mandate transaction using Net Banking. | | schedule\_date | Debit start date | Required | | expiry\_date | Debit end date | Required | | amount | Transaction Amount | Required | | identifier | UDF Field | Optional | |
| Event /Trigger of Request Type | On a click of pay button the payment request gets triggered. |
| Sample Request Format | rqst\_type=TRS|rqst\_kit\_vrsn=1.0.11|tpsl\_clnt\_cd=T2658|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={ mandateData{amount\_type:M~frequency:ADHO~account\_number:914010009305862~ ifsc\_code:KKBK0001360~expiry\_date:02-02-2018~amount:5000~identifier:12345~schedule\_date:25-03-2017}}|rqst\_amnt=2.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/TRSTest/Responsepayload.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_2.0\_0.0|clnt\_dt\_tm=11-03-2014|tpsl\_bank\_cd=9980|cust\_id=abc123xyz|card\_details=fbd440061a1b866333e4756e1137c37ca5b84569919b60d19f6e86f33efaee6511ea060747780f8e1b7e55e2b0e0e69880cfb5e9a2bc593d4f9b019fd9f9ed2a076d9abb9f9f1fcf64631d17974a30b1d3fd01c3ff1fc2918801ad0f0834f8e4336c51d3003ba3067159d9b243a445deaa4f07e60661c66dbf22ac01bf7b2adffee17c979ba01837bfabbc9b3a0b689b6f8b58e705c4375921de09af1a7c0057c03c255c847411a50d6383f951b27f61|hash=033cd6718532a848fae444db26fbba1d8a1d4820 |
| Sample Response Format | **If Request Type=”TRS” Response**  txn\_status=0300|txn\_msg=success|txn\_err\_msg=|clnt\_txn\_ref=7421002|tpsl\_bank\_cd=9980|tpsl\_txn\_id=71047298|txn\_amt=100.00|tpsl\_txn\_time=05-03-2014 23:13:12|card\_ID=12345|Alias\_name=VISA\_DBT\_\*\*\*\*\_1234|BankTransactionId =1234567|rqst\_token= d3e37c1f-d521-405b-a654-edb5bc1473f6 |mandate\_reg\_no=751786473|hash=67de08e331815f1bbc7ddfea73602e82c0613d1e |
| List of parameters For Real Time Transaction verification Payload – (Request Type TRS) | For Transaction Payload – (Request Type TWC)  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
|  | **Sample Java Code:**  **Example:**  TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("TRS");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("EXISTING\_SRC\_ITC~ mandateData{amount\_type:M~frequency:ADHO~account\_number:914010009305862~ ifsc\_code:KKBK0001360~expiry\_date:02-02-2018~amount:5000~identifier:12345~schedule\_date:25-03-2017}");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("9980");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID("");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setCustID("19872627");//Mandatory  objTransactionRequestBean.setCardName("dheeraj");  objTransactionRequestBean.setCardNo("1245142514251425");  objTransactionRequestBean.setCardCVV("125");  objTransactionRequestBean.setCardExpMM("05");  objTransactionRequestBean.setCardExpYY("2014");  objTransactionRequestBean.setStrTimeOut(“1000”) //this should be in seconds  Call the objTransactionRequestBean.getTransactionToken() method . This method will return two possible outputs.   1. Will get the transaction redirection URL with Unique token ID.   Example:  https://www.tekprocess.co.in/PaymentGateway/txnreq.pg?id=8bdb7304-e6cd-4b08-8168-aa51c438e42b   1. If any validation fail will get only error code. (Starts with ERROR) 2. In case of error code please refer error code sheet.   **Sample DotNet Code:**    **Sample PHP Code:** |
|  |  |

# “Verification Request Types”

# “S” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“S”** |
| Description | A scenario where Merchants did receive the only success status and still wish to again to verify the payment status on real time before informing to end customer.  Note\*\*TPSL re-verifies the payment status from end Customer’s bank in case of net banking and from acquiring Bank in case of credit /debit card |
| Event /Trigger of Request Type | Backend – Server to Server Communication |
| Sample Request Format | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “S”, refer to list of parameters of “S” request Type for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:** TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("S");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID("71047882");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  Call the objTransactionRequestBean.getTransactionToken() method . This method will return  two possible output.   1. Will get the decrypted response 2. **Example:**   Rqst\_type=S|rqst\_kit\_vrsn=1.0.1|tpsl\_clnt\_cd=T2551|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={email:myname@domain.com}|  rqst\_amnt=12.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/LoginModule/Test.jsp|  s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_120.0\_0.0|  clnt\_dt\_tm=14-03-2014|tpsl\_bank\_cd=470|hash=8cf71e0eb3f7273113b7e73e0db671d8   1. If any validation fail will get only error code. (Starts with ERROR)   **Sample DotNet Code:**    **Sample PHP Code:** |
| Sample Response Format | **If Request Type=”S” Response**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Normal SUCCESS Transaction|clnt\_txn\_ref=TPSL209177648|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228587|txn\_amt=1.00|tpsl\_txn\_time=14-11-2014 10:03:25|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a4e3e263-bea5-4f63-ac04-e7a6822e03cb  **If Request Type=”S” Response (Auth code Specific Merchant)**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Normal SUCCESS Transaction|clnt\_txn\_ref=TPSL209177648|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228587|txn\_amt=1.00|tpsl\_txn\_time=14-11-2014 10:03:25|tpsl\_rfnd\_id=NA|bal\_amt=NA|auth=1234|rqst\_token=a4e3e263-bea5-4f63-ac04-e7a6822e03cb    **Please note: For auth code to get in response to merchants, TPSL need to configure (Customize Response) in Merchant Configuration Module.** |
| List of parameters For Real Time Transaction verification Payload – (Request Type S) | **Conditional**: clnt\_txn\_ref and tpsl\_txn\_time or tpsl\_txn\_id should be provided  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
| For Real Time Dual Verification Payload:  For Request Type = “S” : Make a request for transaction dual verification at transaction time. | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “S”, refer to refer to list of parameters of “S” for setting parameter values in TransactionRequestBean.** Call the objTransactionRequestBean.getTransactionToken() method. This method return the below mention response.  **Example:**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Normal SUCCESS Transaction|clnt\_txn\_ref=TPSL209177648|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228587|txn\_amt=1.00|tpsl\_txn\_time=14-11-2014 10:03:25|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=a4e3e263-bea5-4f63-ac04-e7a6822e03cb |

# “O” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“O”** |
| Description | In a scenario where final success or fail status is not received from TPSL to the Merchant then, Merchant can request status update though “O” request type. |
| Event /Trigger of Request Type | Backend – Server to Server Communication |
| Sample Request Format | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “O”, refer to list of parameters of “O” request Type for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**  **Example:** TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("O");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("TXN0001");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID("71047882");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  Call the objTransactionRequestBean.getTransactionToken() method . This method will return  two possible output.   1. Will get the decrypted response 2. **Example:**   Rqst\_type=O|rqst\_kit\_vrsn=1.0.1|tpsl\_clnt\_cd=T2551|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={email:myname@domain.com}|  rqst\_amnt=12.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/LoginModule/Test.jsp|  s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_120.0\_0.0|  clnt\_dt\_tm=14-03-2014|tpsl\_bank\_cd=470|hash=8cf71e0eb3f7273113b7e73e0db671d8   1. If any validation fail will get only error code. (Starts with ERROR)   **Sample DotNet Code:**    **Sample PHP Code:** |
| Sample Response Format | **If Request Type=”O” Response**  **FOR NO REFUND TRANSACTIONS**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Success Test Bank Transaction.|clnt\_txn\_ref=TPSL939584727|tpsl\_bank\_cd=470|tpsl\_txn\_id=165291964|txn\_amt=2.00|tpsl\_txn\_time=08-07-2015 14:15:41|tpsl\_rfnd\_id=NA|bal\_amt=2.00|REFUND\_DETAILS=NA|rqst\_token=41f717f6-c4fc-4497-9d2d-3e5a6c7fe43c  **Partial refund -:**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Success Test Bank Transaction.|clnt\_txn\_ref=TPSL939584727|tpsl\_bank\_cd=470|tpsl\_txn\_id=165291964|txn\_amt=2.00|tpsl\_txn\_time=08-07-2015 14:15:41|tpsl\_rfnd\_id=1187861|bal\_amt=1.00| REFUND\_DETAILS={SR\_NO:1~REF\_TXN\_ID:1187861~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED} |rqst\_token=41f717f6-c4fc-4497-9d2d-3e5a6c7fe43c           Transaction status will be success with refund details will be appended in response.  **Full refund -:**  txn\_status=0400|txn\_msg=REFUND|txn\_err\_msg=Success Test Bank Transaction.|clnt\_txn\_ref=TPSL939584727|tpsl\_bank\_cd=470|tpsl\_txn\_id=165291964|txn\_amt=2.00|tpsl\_txn\_time=08-07-2015 14:15:41|tpsl\_rfnd\_id=1187861, 1187862|bal\_amt=0.00|REFUND\_DETAILS={SR\_NO:1~REF\_TXN\_ID:1187861~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED},{SR\_NO:2~REF\_TXN\_ID:1187862~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED}|rqst\_token=  **For Auth code Specific merchants For both partial and full refund :**  txn\_status=0400|txn\_msg=REFUND|txn\_err\_msg=Success Test Bank Transaction.|clnt\_txn\_ref=TPSL939584727|tpsl\_bank\_cd=470|tpsl\_txn\_id=165291964|txn\_amt=2.00|tpsl\_txn\_time=08-07-2015 14:15:41|tpsl\_rfnd\_id=1187861, 1187862|bal\_amt=0.00 |**auth=1234**|REFUND\_DETAILS={SR\_NO:1~REF\_TXN\_ID:1187861~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED},{SR\_NO:2~REF\_TXN\_ID:1187862~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED}|rqst\_token=  **Please note: For auth code to get in response to merchants, TPSL need to configure (Customize Response) in Merchant Configuration Module.**  Transaction status will be returned as REFUND, with refund details will be appended in response.  In both the case this service will returns details of all the refunded transactions.  REFUND\_DETAILS={SR\_NO:1~REF\_TXN\_ID:1187861~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED},{SR\_NO:2~REF\_TXN\_ID:1187862~REF\_TXN\_AMT:1.00~REF\_TXN\_DATE:07-08-2015~STATUS:INITIATED} |
| For Offline Transaction verification Payload – (Request Type O) | **Conditional**: clnt\_txn\_ref and tpsl\_txn\_time or tpsl\_txn\_id should be provided.  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
| For Offline Transaction Verification Payload:  For Request Type = “O”: Make a request to check transaction status after online transaction. | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type =”O” refer to refer to list of parameters of “O” for setting parameter values in TransactionRequestBean.** Call the objTransactionRequestBean.getTransactionToken() method. This method return the below mention response.  **Example :**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Normal SUCCESS Transaction|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=00470b02-45a1-4d0a-bbdc-462edbaeb866  **Example For Auth code Specific merchants :**  txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Normal SUCCESS Transaction|clnt\_txn\_ref=TPSL1117741786|tpsl\_bank\_cd=9980|tpsl\_txn\_id=228583|txn\_amt=1.00|tpsl\_txn\_time=14-11-2014 09:42:39|tpsl\_rfnd\_id=NA|bal\_amt=NA|**auth=1234**|rqst\_token=00470b02-45a1-4d0a-bbdc-462edbaeb866 |

# Sample Verification Response for Different Transaction Status For System

---------------------------------------- For Success Transaction (T) -----------------------------------------

txn\_status=0300|txn\_msg=SUCCESS|txn\_err\_msg=Success Test Bank Transaction.|clnt\_txn\_ref=2358|tpsl\_bank\_cd=470|tpsl\_txn\_id=140813616|txn\_amt=10.00|tpsl\_txn\_time=09-02-2015 16:30:32|tpsl\_rfnd\_id=NA|bal\_amt=NA|base\_fee=10.00|tpsl\_charges =0.00|service\_tax =0.00|rqst\_token=27384ed2-4e0c-46b8-b729-aa50204473cb

---------------------------------------- For User Aborted Transaction (U) ----------------------------------

txn\_status=0395|txn\_msg=User Aborte|txn\_err\_msg=PENDING|clnt\_txn\_ref=6911|tpsl\_bank\_cd=470|tpsl\_txn\_id=134206249|txn\_amt=10.00|tpsl\_txn\_time=05-02-2015 16:23:59|tpsl\_rfnd\_id=NA|bal\_amt=NA|base\_fee=10.00|tpsl\_charges =0.00|service\_tax =0.00|rqst\_token=ead42300-c03e-43ed-94dc-ff3883108b6d

---------------------------------------- For Pending Transaction (P) -----------------------------------------

txn\_status=0396|txn\_msg=AWAITED|txn\_err\_msg=PENDING|clnt\_txn\_ref=2760|tpsl\_bank\_cd=470|tpsl\_txn\_id=117214981|txn\_amt=1.00|tpsl\_txn\_time=30-12-2014 12:01:30|tpsl\_rfnd\_id=NA|bal\_amt=NA|base\_fee=1.00|tpsl\_charges =0.00|service\_tax =0.00|rqst\_token=50651a5e-e517-4b7e-8916-de21dba7cf75

---------------------------------------- For Aborted Transaction (D) -----------------------------------------

txn\_status=0397|txn\_msg=ABORTED|txn\_err\_msg=PENDING|clnt\_txn\_ref=2699|tpsl\_bank\_cd=470|tpsl\_txn\_id=117214763|txn\_amt=1.00|tpsl\_txn\_time=29-12-2014 14:56:50|tpsl\_rfnd\_id=NA|bal\_amt=NA|base\_fee=1.00|tpsl\_charges =0.00|service\_tax =0.00|rqst\_token=1be8b2b6-87b8-432a-992c-bbcc3c6528a9

---------------------------------------- For Failed Transaction (F) -------------------------------------------

txn\_status=0399|txn\_msg=FAILED|txn\_err\_msg=Failed Test Bank Transaction. Transaction Cancelled by User|clnt\_txn\_ref=2356|tpsl\_bank\_cd=470|tpsl\_txn\_id=140813598|txn\_amt=10.00|tpsl\_txn\_time=09-02-2015 15:56:21|tpsl\_rfnd\_id=NA|bal\_amt=NA|base\_fee=10.00|tpsl\_charges =0.00|service\_tax =0.00|rqst\_token=4f209fcd-c5b1-4e7e-879f-ce4cd00a9aa1

---------------------------------------- For Corporate Transaction initiated(W)----------------------------

txn\_status=0398|txn\_msg=Transaction Initiated|txn\_err\_msg= Transaction Initiated|clnt\_txn\_ref=2760|tpsl\_bank\_cd=470|tpsl\_txn\_id=117214981|txn\_amt=1.00|tpsl\_txn\_time=30-12-2014 12:01:30|tpsl\_rfnd\_id=NA|bal\_amt=NA|base\_fee=1.00|tpsl\_charges =0.00|service\_tax =0.00|rqst\_token=50651a5e-e517-4b7e-8916-de21dba7cf75

# “Refund Initiation Request Type”

# “R” Request Type Integration

|  |  |
| --- | --- |
| Request Type | **“R”** |
| Description | Merchant’s requirement is to be able to mark refund manually from Admin module . However the request type “R” should be sent through API for TPSL to process the refund. |
| Event /Trigger of Request Type | Backend – Server to Server Communication |
| Sample Request Format | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “R”, refer to list of parameters of “R” request Type for setting parameter values in TransactionRequestBean.**  **Sample Java Code:**    **Example:** TransactionRequestBean objTransactionRequestBean=new TransactionRequestBean();  objTransactionRequestBean.setStrRequestType("R");  objTransactionRequestBean.setStrMerchantCode("T2663");  objTransactionRequestBean.setMerchantTxnRefNumber("");  objTransactionRequestBean.setStrAmount("120.00");  objTransactionRequestBean.setStrCurrencyCode("INR");  objTransactionRequestBean.setStrITC("email:myname@domain.com");  objTransactionRequestBean.setStrReturnURL("https://tpslvksrv6046/MerchantIntegrationClient/Responsepayload.jsp ");  objTransactionRequestBean.setStrShoppingCartDetails("Test\_120.0\_0.0");  objTransactionRequestBean.setTxnDate("24-12-2013");  objTransactionRequestBean.setStrBankCode("470");  objTransactionRequestBean.setWebServiceLocator("https://www.tpsl- india.in/PaymentGateway/services/TransactionDetailsNew");  objTransactionRequestBean.setStrTPSLTxnID("71047882");  objTransactionRequestBean.setStrMobileNumber("8451053257");  objTransactionRequestBean.setKey(key.getBytes());  objTransactionRequestBean.setIv(iv.getBytes());  objTransactionRequestBean.setStrCustomerName("xyz");  objTransactionRequestBean.setStrEmail("myname@domain.com");  objTransactionRequestBean.setStrTimeOut(“1000”) this should be in seconds  Call the objTransactionRequestBean.getTransactionToken() method . This method will return  two possible output.   1. Will get the decrypted response 2. **Example:**   Rqst\_type=R|rqst\_kit\_vrsn=1.0.1|tpsl\_clnt\_cd=T2551|clnt\_txn\_ref=TXN0001|clnt\_rqst\_meta={email:myname@domain.com}|rqst\_amnt=12.00|rqst\_crncy=INR|rtrn\_url=https://tpslvksrv6046/LoginModule/Test.jsp|s2s\_url=https://tpslvksrv6046/LoginModule/Test.jsp|rqst\_rqst\_dtls=Test\_120.0\_0.0|clnt\_dt\_tm=14-03-2014|tpsl\_bank\_cd=470|hash=8cf71e0eb3f7273113b7e73e0db671d8   1. If any validation fail will get only error code. (Starts with ERROR)   **Sample DotNet Code:**    **Sample PHP Code:** |
| Sample Response Format | **If Request Type=”R” Response**  txn\_status=0400|txn\_msg=Refund Initiated|txn\_err\_msg=|clnt\_txn\_ref=NA|tpsl\_bank\_cd=470|tpsl\_txn\_id=71047288|txn\_amt=100|tpsl\_txn\_time=05032014 124902687|tpsl\_rfnd\_id=543272|bal\_amt=0|rqst\_token= |
| List of parameters For Real Time Transaction verification Payload – (Request Type S) | **Conditional**: clnt\_txn\_ref and tpsl\_txn\_time or tpsl\_txn\_id should be provided.  The request parameters are mentioned in [***List of Parameter***](#Paramerter)***.*** |
| For Refund Transaction Payload    For Request Type = “R” : Make a request to refund a transaction. | Create the object of TransactionRequestBean class and set the required value in the bean object. **If Request Type is “R”, refer to refer to list of parameters of “R” for setting parameter values in TransactionRequestBean.** Call the objTransactionRequestBean.getTransactionToken() method. This method return the below mention response.  **Example :**  Res\_String= txn\_status=0400|txn\_msg=Refund Initiated|txn\_err\_msg=|clnt\_txn\_ref=14032014150942|tpsl\_bank\_cd=470|tpsl\_txn\_id=71048731|txn\_amt=120.00|tpsl\_txn\_time=14-03-2014 16:13:12|tpsl\_rfnd\_id=543339|bal\_amt=45.44|rqst\_token= |

# Server to Server Payment Posting

# TPSL PG System will push the transaction response for every successful transaction to the pre-defined merchant URL in secured server to server communication.

* This Response from TPSL PG system to Merchant will be same as browser to browser “T” type transaction response. Hence, merchant has to ensure that the transaction response is updated only once in its system.
* As this is a Payment posting, TPSL PG system expects an acknowledgment from Merchant server on receipt and processing of data. Merchant has to send an acknowledge of the response in the below format

**Format for acknowledgement to be received from the merchant server:**

**MRCT\_TXN\_ID|PG\_TXN\_ID|STATUS**

**Note:** Inthe following scenarios, TPSL PG will try to resend the transaction response for 3 times.

1. If TPSL PG system does not receive acknowledgement status 1,
2. If the connectivity is not established in between TPSL PG system and Merchant system.

|  |  |
| --- | --- |
| Steps to get response data | Respone = to1X75D6TLTHBXzNWwNkBblOSTgGOhnBnfMWwyseII7n3gTOV1lIu75gWSfhzbpJtz6XoEabWlg8H8eOmgIOzTmMxOcUdoonDEhmRklGm1hXlJ/CmusJCMPIXvQzc+fccoFofM5yPlqTb8xGOEN2A2jSmxWFsPqHBwkblBLVVRUQ82AqhpOTVbm5LIHTAMUKi/RPBLDSvdBe7VllcKqlHwR5uZLdRW1Oy1ZPK2TpH68LtzTK4UUg8C7cwgKoCsdVrualQnPTvL1gSm5dXWLUmuBPdJT8oC25PqFav9Oe3Ihjh0eha4Y2rZapaeXg29RP+JxHXhZczoYylSjKAP4EwtDB  After getting the response from PG to do the below mention step.   1. Create the object of TransactionResponseBean Class . 2. Set all the bean value as defined below   beanObj.setIv(iv.getBytes());  beanObj.setKey(key.getBytes());  beanObj.setResponsePayload(EncryptedRespone); //set the pg response message  **Call** the beanObj.getResponsePayload() method. This method return the below  mention response.  txn\_status=0300|txn\_msg=success|txn\_err\_msg=NA|clnt\_txn\_ref=TPSL210915453|tpsl\_bank\_cd=470|tpsl\_txn\_id=117203569|txn\_amt=2.00|clnt\_rqst\_meta={itc:NIC~TXN0001~122333~rt14154~8 mar 2014~Payment~forpayment}{email:email@techproces.co.in}{mob:9856987456}|tpsl\_txn\_time=12-11-2014 16:58:24|tpsl\_rfnd\_id=NA|bal\_amt=NA|rqst\_token=c9831306-23e8-4d54-ac18-b7faa8edb73e|hash=d924c4bdc8b8e542453345bb2b719318b26219e9 |
| Steps to send Acknowledgement | On Successful processing of above data Merchant server should write response on below format:  MRCT\_TXN\_ID|PG\_TXN\_ID|STATUS  MRCT\_TXN\_ID :Merchant Transaction Reference Number  PG\_TXN\_ID :TPSL Transaction id  STATUS :status of the transaction at merchant server  **Note:** Possible values 0 or 1 only  0 :Transaction received but not processed successfully at merchant server  1 :Transaction Processed successfully at merchant server. |



**Shopping Cart Sample Flow – Figure -4.0**

|  |  |
| --- | --- |
| Format :product1\_amount1\_commission1~ product2\_amount2\_commission2~ product3\_amount3\_commission3~  product1 : product code. (Provided by TPSL)  amount1 :amount of the product  commission1 :commission amount of the product | Shopping Cart Merchant |

# List of Parameters

