BENEFIT PAYMENT GATEWAY

Merchant Integration Document

Publishing Date: 05 Nov 2020

Version 1.2

Disclosure and Confidentiality

Disclosure Agreement

This document is a confidential document and Member bank/institution should disclose this document fully or partially, to third party/merchants only when the required NDA (Non-disclosure Agreement) is signed.

For the purposes of this rule, proprietary and confidential information shall include, but not be limited to, the following:

- All information concerning the technical standards and procedures used in implementing and operating the Service and the Network.
- Information not previously disclosed to the public concerning any member or Service that is under consideration for inclusion in the operation of the Network.

Table of Contents

Contents

Revision

Version Number	Date Created	Description of Changes
1.0	28 May 2020	Initial Draft preparation
1.1	07 June 2020	Added REST API integration steps
1.2	29 June2020	Updated Merchant notification flow.
1.3	14 March 2021	Add one more public IP
1.4	27 Novmber 2022	- Update UAT URL - Update Benefit Logo

Chapter 1 About this Book

This is a technical document that explains the various methods of integration that is supported by BENEFIT Payment Gateway along with sample code snippets to help the merchant to integrate the merchant portal to Payment Gateway and process payments.

1.1. Target Audience

The target audiences for this document are as:

- Any developer/integrator working on behalf / for any merchant who has signed up with BENEFIT member banks/institutions to process transactions on BENEFIT Payment Gateway

- Any administrator or acquirer institution user who requires appropriate information of BENEFIT Payment Gateway integration process.

Chapter 2 Merchant Prerequisites

Readers of this user guide should be familiar with basic either of the languages Java, ASP .Net and PHP.

Hardware Prerequisites

Merchants can use their existing hardware for transaction processing via Payment Gateway. Merchants may have a variety of arrangements for hosting their websites and thus have relevant security mandates for internet access controls and checks. This may include utilization of a Proxy Server which presents informed challenges. It is recommended that the merchant use a Public IP during the integration testing for transaction processing to the Payment Gateway. The merchant should ensure the Payment Gateway Domain and IP address is enabled at the firewall for both incoming and outgoing request/response

Software Prerequisites

The merchant should have the requisite software for connecting to the Payment Gateway depending on the merchant application environment. The merchant may use combinations of OS/Web Server/ Application server whilst setting up and operating the website. Standard Software options are listed below, this list is for reference use only

Operating Systems - Windows 2000/ 2003 / 2008 Server, Sun Solaris, IBM AIX Web/Application Servers - Web/Application Server that support JSP & ASP .Net. The current version with all required patches is recommended to ensure success. Software Installation – Basic software that are required for Web/Application server should be installed at the merchant site. (Java/JDK for JSP integration is essential; similarly, .NET frame work is essential for ASP.NET integration)

Chapter 3 Merchant Integration Process

3.1. Important Notes

1. In order for "BENEFIT Payment Gateway" to process any payment request (**both, test and real requests**), the request must be sent from a **valid domain/IP address** hosted over the **Internet**, as well as, the request **must be** sent **without changing the port number**. For illustration, following are sample links with default port number:

   http://www.companyName.com/**test**Payment
   (this link **doesn't change** the default port number for **http** "port 80")

   http://www.companyName.com:70/**test**Payment
   (this link changes the default port number for **http** from port "80" to "70" which **will result in errors**)

   https://www.companyName.com/**test**Payment
   (this link **doesn't change** the default port number for **https** "port 443")

   https://www.companyName.com:400/**test**Payment
   (this link changes the default port number for **https** from port "443" to "400" which **will result in errors**)

2. The total length of the link (URL) **must not exceeds 254 characters**. Otherwise, an error will occur.

3. Make sure from network/firewall side that the traffic is allowed (**back and forth**) for the following:

   https://test.BENEFIT-gateway.bh | https://www.BENEFIT-gateway.bh
   In case you don’t support domain whitelisting (which is strongly not recommended; domain whitelisting is the reliable approach) then alternatively whitelist the following:
   79.171.242.91 port 443 & 80 | 79.171.242.90 port 443 | 79.171.247.90 port 443 | 79.171.240.90 port 443

3.2. Integration Process

1. Merchant should login to certification environment and download the respective plug-in (based on the platform (Java/ASP etc.) the menu “Merchant Process -> Plug-in Download”
2. Download the “Resource” file & “Keystore” file from the menu “Merchant Process -> Resource File Download.
3. Integrate the plug-in and resource file with merchant web page.
4. Construct the Request message as expected by Payment Gateway
5. Process the response message receiving the response from Payment Gateway.

3.3. Download Plug-in

a. “Merchant Process -> Plug-in Download”:

For JAVA plugin:

- Step 1: Copy all the jar files and paste it in merchant application lib folder.
- Step 2: (if you are using older version than 1.7) Copy **bcprov-jdk15-145.jar** to (JDK) JAVA_HOME\jre\lib\ext
- Step 3: Add the below entry in the java.Security file (JAVA_HOME\jre\lib\security) security.provider.10=org.bouncycastle.jce.provider.BouncyCastleProvider

b. Supported versions

- JAVA – JDK 1.7 and 1.8
- ASP .NET – IKVM 7.2
- PHP – Version 7

3.4. BENEFIT Domain Certificate Import

- While performing the transaction if there is a ssl handshake exception or connectivity error then import the BENEFIT Payment Gateway (UAT/Production) domain certificates in the merchant applitlicaltion / environement.

3.5. Downloading Resource and Keystore Files

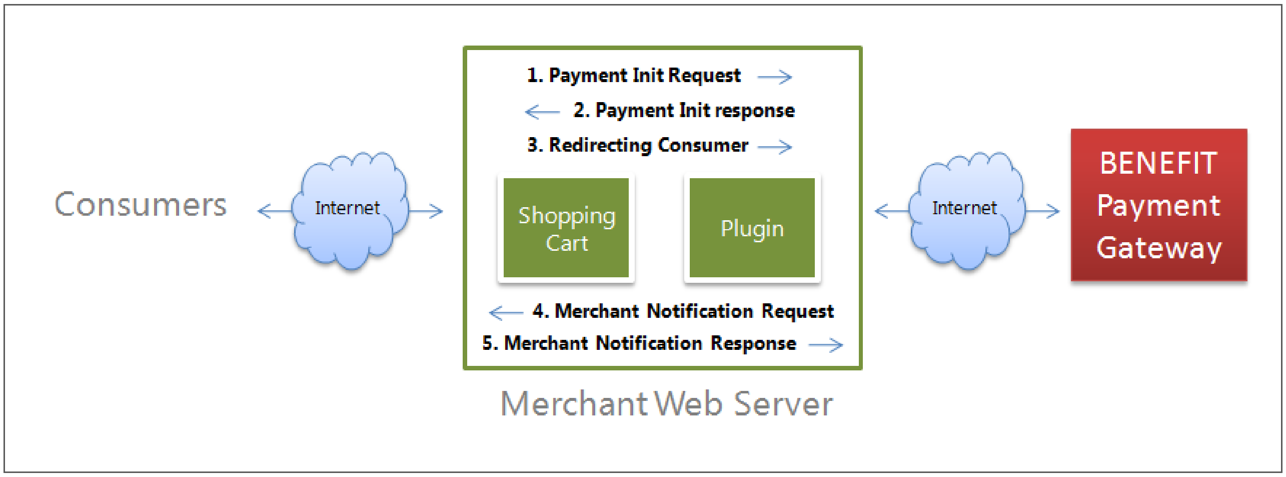
Click on Merchant Process > Resource File Download

a. Downloading KeyStore File

b. Downloading Resource File

c. Place the resource file "**resource.cgn**" and keystore file “KeyStore.bin” within your website files (**The folder that will contain the resource file "resource.cgn" must have the *execute*, *write* and *read* permissions**). The **location** of the resource file will be used later in the communication with "BENEFIT Payment Gateway".

3.6. Transaction Flow



Transaction Processing Flow |

- Customer shops online through Merchant Website and selects payment option as BENEFIT Payment Gateway.
- Merchant forwards the payment request (**1. Payment Init Request**) to BENEFIT Payment Gateway and a URL is received from the plug-in (**2. Payment Init response**).
- The consumer is redirected to the URL hosted on "BENEFIT Payment Gateway" (**3. Redirecting Consumer**) wherein customer enters his card details.
- PIN will be allowed to enter through virtual keypad with random key position.
- BENEFIT Payment Gateway performs the risk check as per the risk profile configured.
- Once the transaction is processed, BENEFIT Payment gateway initiates the encrypted request to notify the transaction result to merchant (**4. Merchant Notification Request**).
- Merchant acknowledge the notification request with a response page. For plugin integration merchant will provide response page with keyword “REDIRECT=” followed by the response URL and for REST API integration merchant will provide JSON response (**5. Merchant Notification Response**).
- Once acknowledgment received from merchant, BENEFIT Payment Gateway provides the encrypted final response to merchant and this will be URL redirection.

3.7. Plugin Integration

3.6.1. Sample For Payment Request (JAVA)

```
<%@page import="com.fss.plugin.iPayBenefitPipe"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%
    iPayBenefitPipe pipe = new iPayBenefitPipe();

        // Do Not change the values of the following parameters
        pipe.setAction("1");
        pipe.setCurrency("048");
        pipe.setLanguage("USA");

        // modify the following to reflect your "Alias Name", "resource" & "keystore" file path.
        pipe.setAlias("EnterAliasName");
        pipe.setResourcePath("E:\\resources\\");
        pipe.setKeystorePath("E:\\resources\\");

        // modify the following to reflect your pages URLs
        pipe.setResponseURL("https://www.abc.com/def");
        pipe.setErrorURL("https://www.abc.com/ghi");

        // modify the following to reflect your transaction details
        pipe.setAmt("1.234");
        pipe.setTrackId("123456");
        pipe.setUdf1("Enter some value");
        pipe.setUdf2("Enter some value");
        pipe.setUdf3("Enter some value");
        pipe.setUdf4("Enter some value");
        pipe.setUdf5("Enter some value");

        String error = "";
        int val = pipe.performPaymentInitializationHTTP();
        If (Val() == 0){
            response.sendRedirect(pipe.getWebAddress());
        }
        else {
            error = pipe.getError();
        }
%>
```

3.6.2. Sample For Payment Response (JAVA)

```
<%@page import="com.fss.plugin.iPayBenefitPipe"%>
<%
    iPayBenefitPipe pipe = new iPayBenefitPipe();

    pipe.setResourcePath("E:\\resources\\");
    pipe.setKeystorePath("E:\\resources\\");
    pipe.setAlias("EnterAliasName");

    String errorText = "";
    String trandata = "";

    try {
        errorText = request.getParameter("ErrorText");
    }
    catch(Exception e) {
    }

    try {
        trandata = request.getParameter("trandata");
    }
    catch(Exception e) {
    }

    String response_code = "";
    if (trandata != null){
        int result = pipe.parseEncryptedRequest(trandata);
        if (result == 0){
            if (pipe.getResult().equals("CAPTURED")){
                out.print("REDIRECT=https://www.abc.com/approved.jsp");
            }
            else if (pipe.getResult().equals("NOT CAPTURED")){
                response_code = pipe.getAuthRespCode();
                //following is a list of values returned in response_code:
                //00 Approved
                //14 Invalid card number
                //33 Expired card
                //36 Restricted card
                //38 Allowable PIN tries exceeded
                //51 Insufficient funds
                //54 Expired card
                //55 Incorrect PIN
                //61 Exceeds withdrawal amount limit
                //62 Restricted Card
                //65 Exceeds withdrawal frequency limit
                //75 Allowable number PIN tries exceeded
                //76 Ineligible account
                //78 Refer to Issuer
                //91 Issuer is inoperative
                out.print("REDIRECT=https://www.abc.com/declined.jsp");
            }
            else if (pipe.getResult().equals("DENIED BY RISK")){
                out.print("REDIRECT=https://www.abc.com/rejected.jsp");
            }
            else{
                out.print("REDIRECT=https://www.abc.com/error.jsp");
            }
        }
        else{
            if (errorText != null){
                out.print("REDIRECT=https://www.abc.com/response_error.jsp");
            }
            else{
                out.print("REDIRECT=https://www.abc.com/response_unknown_error.jsp");
            }
        }
    }
%>
```

“Merchant Notification” Feature:

Based on the data received from BENEFIT Payment Gateway, a server-side code must write the desired URL in the page itself preceded with **the phrase “REDIRECT=” (the phrase must be in upper-case only)** in case the “Merchant Notification” feature is enabled. Otherwise (“Merchant Notification” feature is disabled), a page redirection must be done by the response page. Please **note** that when “Merchant Notification” feature is enabled then **only server-side codes** are allowed within the response page; any HTML, CSS and Javascript codes are **not allowed**.

NOTE: Other involved pages like success page, non-success page and exception page which are created by your side, they must be available and working fine at all times. Any invalid URL will result in errors. You may use the same response page to process other functions like inserting related data into your website database.

Sample for data received from BENEFIT Payment Gateway:

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF
02CF6A0A5643F31C78340913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90
D29338545DA582B0F3500BA93753136376905 31C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B
0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9AAF38E044F0A346943485067
78257040533E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

3.6.3. Code snippet for ASP Integration

a. BENEFIT Hosted Payment Integration (Purchase)

For transaction flow kindly refer section 3.5,

```
/** Request Processing**/
//Merchant can connect iPay BENEFIT Plugin usingbelow step
iPayBENEFITPipe pipe = new iPayBENEFITPipe();

//Initialization
String resourcePath = “c:\\resourcepath”;
String keystorePath = “c:\\ resourcepath”;
```

If Merchant Notification enabled at Merchant level,

Merchant need to send Merchant response page URL to check merchant is active or not, before receiving final response from PG

String recieptURL=“http://www.demomerchant.com/response.aspx”;

If Merchant Notification disabled at Merchant level,

Merchant can send actual result page URL to get final response from PG

```
String ResponseURL=“http://www.demomerchant.com/result.aspx”;
String errorURL= “http://www.demomerchant.com/error.aspx”;
String action=”1”;               // 1 – Purchase
String aliasName = “aliasName”;  //Terminal Alias Name
String currency = “414”;         //Transaction Currency (ex: “414 ”)
String language = “language”;    //English – “EN”, Arabic – “AR”
String amount = “1000.000”;      //Transaction Amount
String trackid = “109088888”;     //Merchant Track ID
//User Defined Fields
String Udf1=
“”;//Always keep it
empty
String Udf2= “Udf2”;
String Udf3= “Udf3”;
String Udf4= “Udf4”;
String Udf5= “Udf5”;

//Set Values
pipe.trackId = trackid;
pipe.Alias = aliasName;
pipe.ResourcePath = resourcePath;
pipe.Language = language;
pipe.Action = action;
pipe.Amt = amount;
pipe.Currency = currency;
pipe.Udf1 = Udf1;
pipe.Udf2 = Udf2;
pipe.Udf3 = Udf3;
pipe.Udf4 = Udf4;
pipe.Udf5 = Udf5;
pipe.KeystorePath = resourcePath;
pipe.ResponseURL= ResponseURL;
pipe.ErrorURL = errorURL;
//For Hosted Payment Integration , the method to be called is
int val = pipe.PerformPaymentInitializationHTTP();
if(val==0){
       Response.Redirect (pipe.WebAddress);

       //To redirect the web address.
}else{
    pipe.Error;
}

/** End of Request Processing**/

/** Response received from Payment Gateway to Merchant **/
```

If Merchant notification is disabled, then BENEFIT Payment gateway will provide the final response in URL redirection. Below is the sample response from BENEFIT PG to merchant

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF
02CF6A0A5643F31C78340913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90
D29338545DA582B0F3500BA937531363769053 1C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B
0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9AAF38E044F0A346943485067
78257040533E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

```
/** Request Processing**/

//Merchant can connect iPay BENEFIT Plugin with the below step

java_require('http://localhost/plugins/phpnormal/iPayBENEFITPipe.jar');
$myObj = new Java("com.fss.plugin.iPayBENEFITPipe");

//initialization
$resourcePath = "c:\\resourcepath";
$keystorePath = "c:\\ keystorePath";
```

If Merchant Notification enabled at Merchant level,

Merchant need to send Merchant response page URL to check merchant is active or not, before receiving final response from PG

```
$ResponseURL= "http://www.demomerchant.com/response.jsp";
```

If Merchant Notification disabled at Merchant level,

Merchant can send actual result page URL to get final response from PG

```
$ResponseURL= "http://www.demomerchant.com/result.jsp";
$errorURL= "http://www.demomerchant.com/error.jsp";
$action="1";            // 1-Purchase
$aliasName = "TerminalaliasName";  //Terminal Alias Name.
$currency = "currency";            //Transaction Currency. (ex: "414")
$language = "language"; (ex: "USA")
$amount="10.000";                   //Transaction Amount.
$trackid = "109088888";             //Merchant Track ID.
//User Defined Fields.

$Udf2= "Udf2";
$Udf3= "Udf3";
$Udf4= "Udf4";
$Udf5= "Udf5";

//Set Values
$myObj->setResourcePath(resourcePath);
$myObj->setKeystorePath(keystorePath);
$myObj->setAlias(aliasName);
$myObj->setAction( action );
$myObj->setCurrency(currency);
$myObj->setLanguage(language);
$myObj->setResponseURL(receiptURL );
```

```
$myObj->setErrorURL(errorURL);
$myObj->setAmt(amount);
$myObj->setTrackId(trackid);

$myObj->setUdf2(Udf2);
$myObj->setUdf3(Udf3);
$myObj->setUdf4(Udf4);
$myObj->setUdf5(Udf5);


/** In plugin the method to be called is **/
$result = $myObj->performPaymentInitializationHTTP();

//To redirect the web address.
if($result == 0) {
$url=$myObj->getWebAddress();
header( 'Location:'.$url ); //Redirect to Payment Gateway page
} else {
$myObj->getError(); //Problem in connecting the Payment Gateway
}
/** End of Request Processing**/



/** Response received from Payment Gateway to Merchant **/
```

If Merchant notification is disabled, then BENEFIT Payment gateway will provide the final response in URL redirection. Below is the sample response from BENEFIT PG to merchant

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF
02CF6A0A5643F31C78340913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90
D29338545DA582B0F3500BA9375313637690531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B
0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9AAF38E044F0A346943485067
78257040533E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

If merchant notification enabled for the merchant, then follow the below process.

Merchant Notification flow

- Once the transaction is processed, BENEFIT Payment gateway initiates the encrypted request with response parameters to notify the transactionresult. Merchant has to decrypt the response using the decrption process given below.
- Merchant acknowledge the notification request with a response page at their end which will have only the keyword "REDIRECT=" followed by the response URL.
- Once acknowledgment received from merchant, BENEFIT Payment Gateway provides the final encrypted response to merchant and this will be URL redirection.
- Merchant has to decrypt the response using the decrption process given below and show response to the customer.

Below is the sample merchant notification request and response,

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C7834091
3929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313637690
531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9A
AF38E044F0A346943485067782570405 33E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

Acknowledgment response from merchant to PG:

REDIRECT=http://www.demomerchant.com/result.jsp

If BENEFIT payment gateway receives the acknowledgement from merchant, PG logs the re-sponse and provide the response back to merchant. This will be URL redirection.

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C7
8340913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313
637690531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC
577300F9AAF38E044F0A346943485067782570405 33E4FD48A9C
61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

If BENEFIT payment gateway receives the acknowledgement from merchant, PG logs the re-sponse and provide the response back to merchant. This will be URL redirection.

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C78340
913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA937531363769053
1C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9AAF3
8E044F0A346943485067782570405 33E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

```
//To decrypt the above response, Merchant should follow the below step:
//Merchant have to set certain fields in plugin as in request processing.
//Create the plugin object as, iPayBENEFITPipe pipe = new iPayBENEFITPipe ();
//Initialization
String resourcePath = “c:\\resourcepath”; String keystorePath = “c:\\ resourcepath”;
String aliasName = “aliasName”;  //Terminal Alias Name
String trandata = Request.Form["trandata "].ToString();     //which is getting from payment
gateway response.
//Set Values
pipe.Alias = aliasName; pipe.ResourcePath = resourcePath; pipe.KeystorePath = resourcePath;

String errorText = Request.Form["ErrorText"].ToString();

The method to be called is,
int result = pipe.parseEncryptedRequest(trandata);

if(result!=0){ pipe.Error;
}else{

if(errorText==null){
       pipe.Result;
       pipe.Date;
       pipe.Ref;
       pipe.TrackId;
       pipe.TransId;
       pipe.Amt;

       pipe.Udf2;
       pipe.Udf3;
       pipe.Udf4;
       pipe.Udf5;
       pipe.PaymentId;
       pipe.Auth;
       pipe.Date;
       pipe.ResponseCode;
}
else{
       Request.Form["ErrorText"];
       Request.Form["paymentid"];
       Request.Form["Error"];
       Request.Form["trackid"];
       Request.Form["amt"];

       Request.Form["udf2"];
       Request.Form["udf3"];
       Request.Form["udf4"];
       Request.Form["udf5"];
}
}
/** End of Response Processing**/
```

Success Response Parameters	**Description**
pipe.Result	**CAPTURED** for Purchase transaction. Other than above value considered the transaction as failure
pipe.Date	Contains Postdate
pipe.Ref	Transaction reference no.
pipe. TrackId	Containts merchant unique track id
pipe.TransId	PG Transactuion ID
pipe.Udf2 pipe.Udf3 pipe.Udf4 pipe.Udf5	UDF Values sent during initial leg
pipe.PaymentId	PG PaymentID
pipe.Amt	Transaction amount
pipe.getResponseCode	Contains Response code
pipe.getAuth	Contains Auth code
pipe.getDate	Contains Post Date

Failure Response Parameters (If Request.Form["ErrorText"] is empty**)**	**Description**
pipe.Result	Contains the result code of PG/Switch decline cases **(NOT CAPTURED, DENIED BY RISK, HOST TIMEOUT)**
pipe. TrackId	Containts merchant unique transaction id
pipe.Amt	Transaction amount

pipe.Udf2 pipe.Udf3 pipe.Udf4 pipe.Udf5	UDF Values sent during initial leg
pipe.Ref	Transaction reference no.
pipe.TransId	PG Transactuion ID.
pipe.PaymentId	PG PaymentID

Failure Response Parameters (If Request.Form["ErrorText"]; is not empty**)**	**Description**
Request.Form["ErrorText"]	Contains the error description of the error.
Request.Form["paymentid"]	PG PaymentID
Request.Form["trackid"]	Merchant track id
Request.Form["amt"]	Transaction amount
Request.Form["udf2"] Request.Form["udf3"] Request.Form["udf4"] Request.Form["udf5"]	UDF Values sent during initial leg
Request.Form["Error"]	Contains Error code of the error.

Registering ASP. NET Plug-in the Merchant Server

- Kindly Place the provided DLL's into some Physical Path of the system.

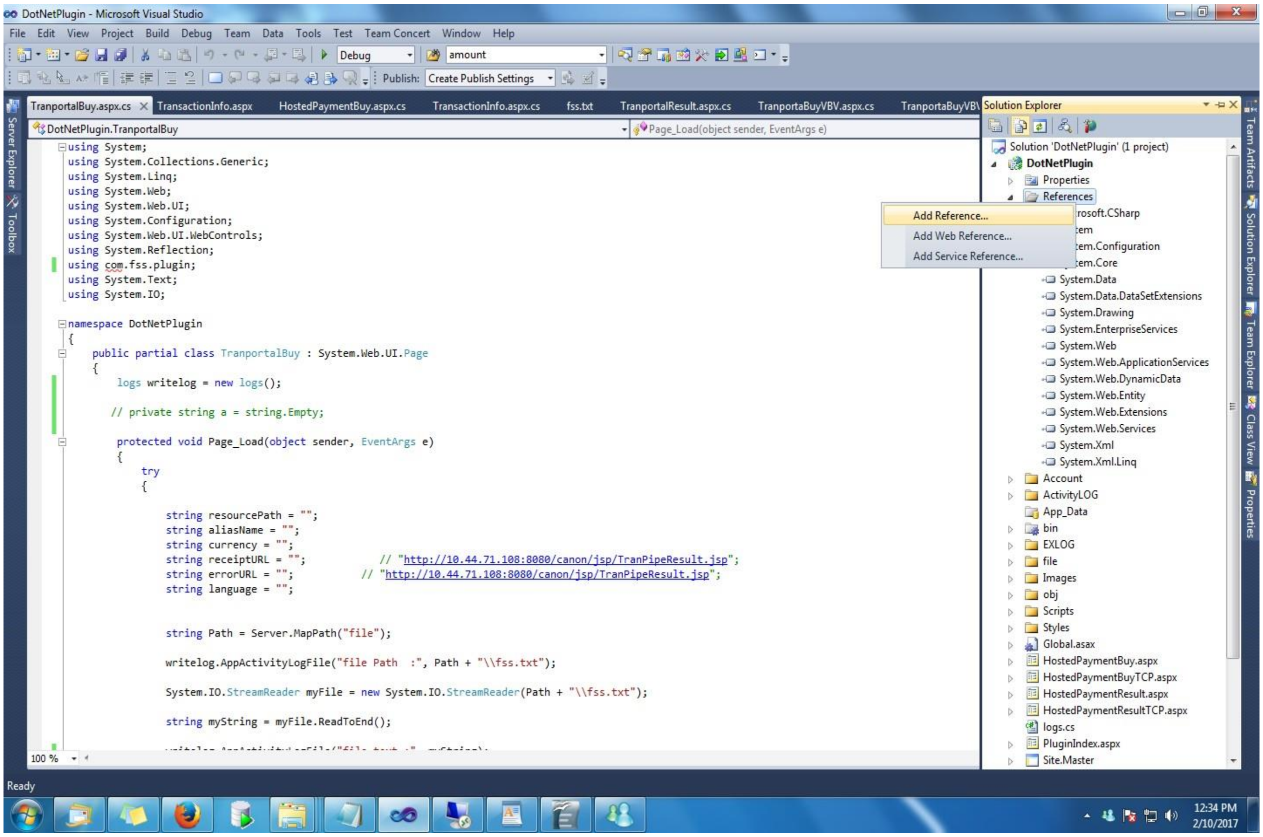
- And Refer the DLL's into the application by following the below steps

  DLL's to be referred:

  - iPayBENEFITPipe.dll
  - supporting DLL’s

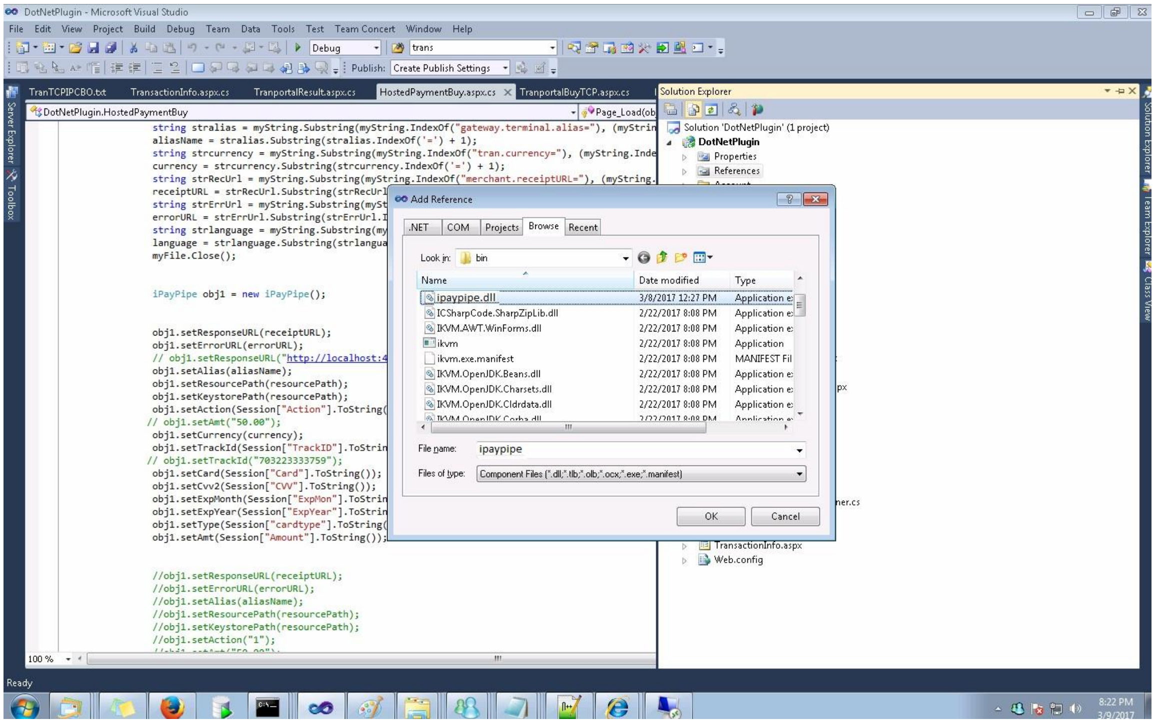
- How to add :

Right Click the Reference tab of the Solution Explorer in the Visual studio IDE and click Add Reference



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Configuration;
using System.Web.UI.WebControls;
using System.Reflection;
using com.fss.plugin;
using System.Text;
using System.IO;

namespace DotNetPlugin
{
    public partial class TranportalBuy : System.Web.UI.Page
    {
        logs writelog = new logs();

       // private string a = string.Empty;

        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {

                string resourcePath = "";
                string aliasName = "";
                string currency = "";
                string receiptURL = "";           // "http://10.44.71.108:8080/canon/jsp/TranPipeResult.jsp";
                string errorURL = "";          // "http://10.44.71.108:8080/canon/jsp/TranPipeResult.jsp";
                string language = "";


                string Path = Server.MapPath("file");

                writelog.AppActivityLogFile("file Path  :", Path + "\\fss.txt");

                System.IO.StreamReader myFile = new System.IO.StreamReader(Path + "\\fss.txt");

                string myString = myFile.ReadToEnd();

                writelog.AppActivityLogFile("file text :", myString);
```

In the Add Reference Dialog box, browse the DLL's Physical path and select the DLL and click Ok



```
string stralias = myString.Substring(myString.IndexOf("gateway.terminal.alias="), (myStrin
aliasName = stralias.Substring(stralias.IndexOf('=') + 1);
string strcurrency = myString.Substring(myString.IndexOf("tran.currency="), (myString.Inde
currency = strcurrency.Substring(strcurrency.IndexOf('=') + 1);
string strRecUrl = myString.Substring(myString.IndexOf("merchant.receiptURL="), (myString.
receiptURL = strRecUrl.Substring(strRecUrl
string strErrUrl = myString.Substring(mySt
errorURL = strErrUrl.Substring(strErrUrl.I
string strlanguage = myString.Substring(my
language = strlanguage.Substring(strlangua
myFile.Close();


iPayPipe obj1 = new iPayPipe();


obj1.setResponseURL(receiptURL);
obj1.setErrorURL(errorURL);
// obj1.setResponseURL("http://localhost:4
obj1.setAlias(aliasName);
obj1.setResourcePath(resourcePath);
obj1.setKeystorePath(resourcePath);
obj1.setAction(Session["Action"].ToString(
// obj1.setAmt("50.00");
obj1.setCurrency(currency);
obj1.setTrackId(Session["TrackID"].ToStrin
// obj1.setTrackId("703223333759");
obj1.setCard(Session["Card"].ToString());
obj1.setCvv2(Session["CVV"].ToString());
obj1.setExpMonth(Session["ExpMon"].ToStrin
obj1.setExpYear(Session["ExpYear"].ToStrin
obj1.setType(Session["cardtype"].ToString(
obj1.setAmt(Session["Amount"].ToString());


//obj1.setResponseURL(receiptURL);
//obj1.setErrorURL(errorURL);
//obj1.setAlias(aliasName);
//obj1.setResourcePath(resourcePath);
//obj1.setKeystorePath(resourcePath);
//obj1.setAction("1");
```

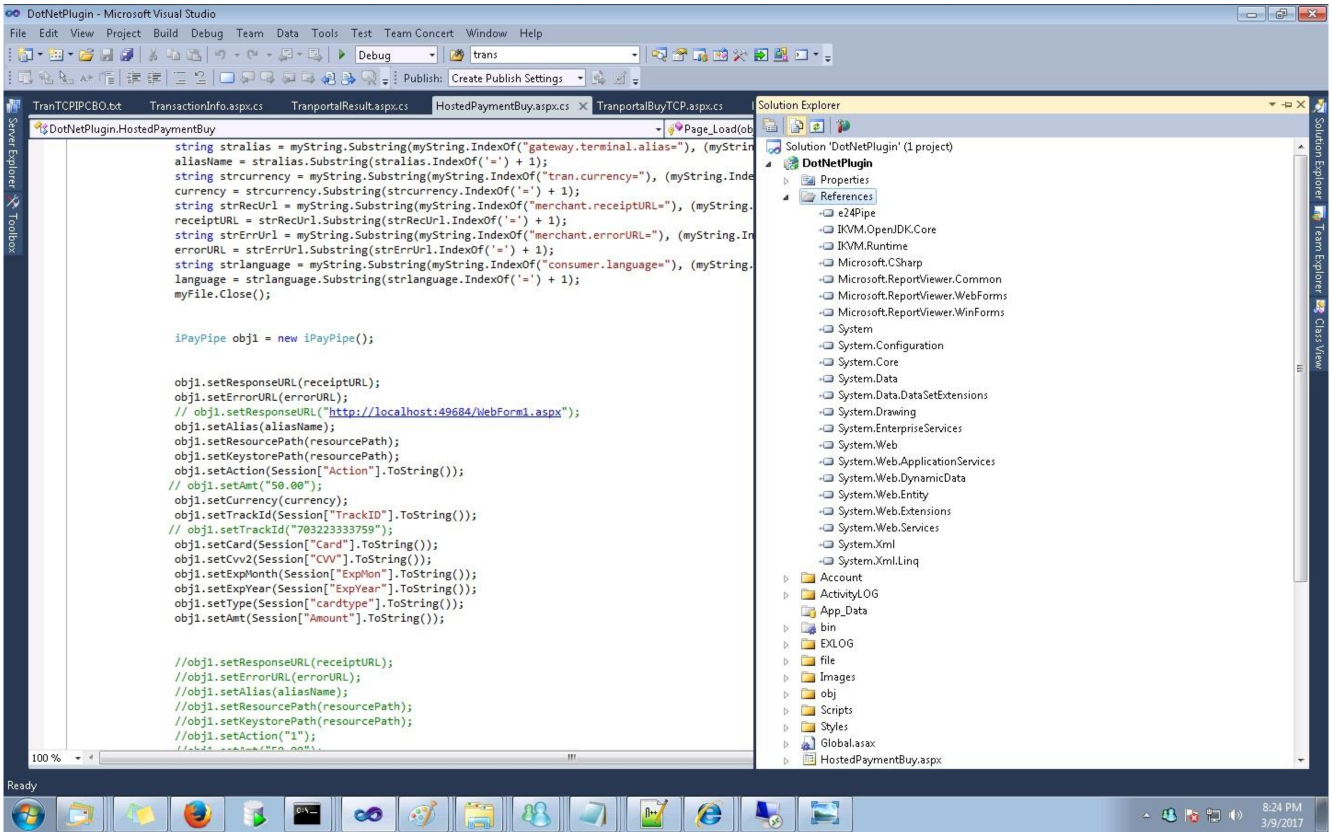
Look in: bin

Name	Date modified	Type
ipaypipe.dll	3/8/2017 12:27 PM	Application e:
ICSharpCode.SharpZipLib.dll	2/22/2017 8:08 PM	Application e:
IKVM.AWT.WinForms.dll	2/22/2017 8:08 PM	Application e:
ikvm	2/22/2017 8:08 PM	Application
ikvm.exe.manifest	2/22/2017 8:08 PM	MANIFEST Fil
IKVM.OpenJDK.Beans.dll	2/22/2017 8:08 PM	Application e:
IKVM.OpenJDK.Charsets.dll	2/22/2017 8:08 PM	Application e:
IKVM.OpenJDK.Cldrdata.dll	2/22/2017 8:08 PM	Application e:

File name: ipaypipe

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

After adding, you could able to see the added DLL into the list of References



```
 string stralias = myString.Substring(myString.IndexOf("gateway.terminal.alias="), (myStrin
 aliasName = stralias.Substring(stralias.IndexOf('=') + 1);
 string strcurrency = myString.Substring(myString.IndexOf("tran.currency="), (myString.Inde
 currency = strcurrency.Substring(strcurrency.IndexOf('=') + 1);
 string strRecUrl = myString.Substring(myString.IndexOf("merchant.receiptURL="), (myString.
 receiptURL = strRecUrl.Substring(strRecUrl.IndexOf('=') + 1);
 string strErrUrl = myString.Substring(myString.IndexOf("merchant.errorURL="), (myString.In
 errorURL = strErrUrl.Substring(strErrUrl.IndexOf('=') + 1);
 string strlanguage = myString.Substring(myString.IndexOf("consumer.language="), (myString.
 language = strlanguage.Substring(strlanguage.IndexOf('=') + 1);
 myFile.Close();


 iPayPipe obj1 = new iPayPipe();


 obj1.setResponseURL(receiptURL);
 obj1.setErrorURL(errorURL);
 // obj1.setResponseURL("http://localhost:49684/WebForm1.aspx");
 obj1.setAlias(aliasName);
 obj1.setResourcePath(resourcePath);
 obj1.setKeystorePath(resourcePath);
 obj1.setAction(Session["Action"].ToString());
// obj1.setAmt("50.00");
 obj1.setCurrency(currency);
 obj1.setTrackId(Session["TrackID"].ToString());
// obj1.setTrackId("703223333759");
 obj1.setCard(Session["Card"].ToString());
 obj1.setCvv2(Session["CVV"].ToString());
 obj1.setExpMonth(Session["ExpMon"].ToString());
 obj1.setExpYear(Session["ExpYear"].ToString());
 obj1.setType(Session["cardtype"].ToString());
 obj1.setAmt(Session["Amount"].ToString());


 //obj1.setResponseURL(receiptURL);
 //obj1.setErrorURL(errorURL);
 //obj1.setAlias(aliasName);
 //obj1.setResourcePath(resourcePath);
 //obj1.setKeystorePath(resourcePath);
 //obj1.setAction("1");
```

likewise add the remaining DLL's into the appplication.

- After adding the **DLL file**, change the property “Copy Local” to “true” so the **DLL file** will be copied to output folder each time after publishing the application.

- Inorder to use the iPayBENEFITPipe into the application, Add the java package into the Dotnet application by adding the ,using com.fss.plugin; namespace into the corresponding ASP pages.

- once process are done correctly you could able to see all the DLL's (referred & Supported) into the BIN folder of the Application.

3.6.4. Code snippet for PHP Integration

a. BENEFIT Hosted Payment Integration (Purchase)

For transaction flow kindly refer section 3.5,

```
/** Request Processing**/

//Merchant can connect iPay Benefit Plugin with the below step

require('../libfiles/iPayBenefitPipe.php');

$myObj =new iPayBenefitPipe();

//initialization
$resourcePath = “c:\\resourcepath”;
$keystorePath = “c:\\ keystorePath”;
```

If Merchant Notification enabled at Merchant level,
Merchant need to send Merchant response page URL to check merchant is active or not, before receiving final response from PG

```
$ResponseURL= “http://www.demomerchant.com/response.jsp”;
```

If Merchant Notification disabled at Merchant level,
Merchant can send actual result page URL to get final response from PG

```
$ResponseURL= “http://www.demomerchant.com/result.jsp”;
$errorURL= “http://www.demomerchant.com/error.jsp”;
$action=”1”;            // 1-Purchase
$aliasName = “TerminalaliasName”;  //Terminal Alias Name.
$currency = “currency”;            //Transaction Currency. (ex: “414”)
$language = “language”; (ex: “USA”)
$amount=“10.000”;                    //Transaction Amount.
$trackid = “109088888”;              //Merchant Track ID.
//User Defined Fields.

$Udf2= “Udf2”;
$Udf3= “Udf3”;
$Udf4= “Udf4”;
$Udf5= “Udf5”;

//Set Values
$myObj->setResourcePath(resourcePath);
$myObj->setKeystorePath(keystorePath);
$myObj->setAlias(aliasName);
$myObj->setAction( action );
$myObj->setCurrency(currency);
$myObj->setLanguage(language);
$myObj->setResponseURL(receiptURL );
$myObj->setErrorURL(errorURL);
$myObj->setAmt(amount);
$myObj->setTrackId(trackid);

$myObj->setUdf2(Udf2);
$myObj->setUdf3(Udf3);
$myObj->setUdf4(Udf4);
```

```
$myObj->setUdf5(Udf5);

/** For Bank Hosted Payment Integration, the method to be called is **/
if(trim($myObj->performPaymentInitializationHTTP())!=0)
 {
echo("ERROR OCCURED! SEE CONSOLE FOR MORE DETAILS");
return;
}else{
 $url=$myObj->getwebAddress();
 echo "<meta http-equiv='refresh' content='0;url=$url'>";
                }

/** End of Request Processing**/
```

```
/** Response received from Payment Gateway to Merchant **/
```

If Merchant notification is disabled, then Benefit Payment gateway will provide the final response in URL redirection. Below is the sample response from Benefit PG to merchant

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF
02CF6A0A5643F31C78340913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90
D29338545DA582B0F3500BA9375313637690531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B
0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9AAF38E044F0A346943485067
78257040533E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

If merchant notification enabled for the merchant, then follow the below process.

Merchant Notification flow

- Once the transaction is processed, Benefit Payment gateway initiates the encrypted request with response parameters to notify the transactionresult. Merchant has to decrypt the response using the decrption process given below.
- Merchant acknowledge the notification request with a response page at their end which will have only the keyword “REDIRECT=” followed by the response URL.
- Once acknowledgment received from merchant, Benefit Payment Gateway provides the final encrypted response to merchant and this will be URL redirection.
- Merchant has to decrypt the response using the decrption process given below and show response to the customer.

Below is the sample merchant notification request and response,

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C7834091
3929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313637690
531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9A
AF38E044F0A346943485067782570405 33E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

Acknowledgment response from merchant to PG:

```
REDIRECT=http://www.demomerchant.com/result.jsp
```

If Benefit payment gateway receives the acknowledgement from merchant, PG logs the re-sponse and provide the response back to merchant. This will be URL redirection.

```
trandata=1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C7
8340913929D90879615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313
637690531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC
577300F9AAF38E044F0A346943485067782570405533E4FD48A9C
61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

```
//To decrypt the above response, Merchant should follow the below step:

Merchant have to set certain fields in iPay Benefit Plugin to process the
response. Create the plugin object as,

//initialization

$resourcePath = c:\\resourcepath"; //Mandatory

$keystorePath = “c:\\ resourcepath”; //Mandatory

//Terminal Alias Name in the merchant portal for the terminal

$ aliasName = “aliasName”;


$myObj =new iPayBenefitPipe();

$myObj->setAlias(trim($aliasName));

$myObj->setResourcePath(trim($resourcePath));

$myObj->setKeystorePath(trim($keystorePath)); //The method to be called for decrypting the
response send by Payment Gateway

$returnValue = $myObj->parseEncryptedRequest($_GET['trandata']);
```

Case1: If the return value from this method is “0” and $_GET[“ErrorText”] is null, then Merchant can get the decrypted data of the response fields.

Ex:

```
// To get result. Value “CAPTURED” is success and other values must be treated as
failure.

$myObj->getResult();

//To get payment ID.

$myObj->getPaymentId();

//To get Transaction ID.

$myObj->getTransId();

// To get Amount

$myObj->getAmt();

// To get Track ID

$myObj->getTrackId()

// To get RRN  No
```

```
$myObj->getRef()
// To get Error
$myObj->getError()
// To get Error description
$myObj->getError_text()
$myObj->getAuth();
$myObj->getDate();
$myObj->getAuthRespCode();
```

Case2: If $_GET[“ErrorText”] is not null, then do the following to get response fields.

```
//To get error
$_GET[“ErrorText”]
// To get Transaction ID
$_GET['tranid']
// To get Payment ID
$_GET['paymentid']
```

Case3: If the return value from this method is not “0”, then Merchant will get the error from below mentioned steps.

```
//To get error
$myObj->getError();
```

Case4: If trandata is null, then merchant need to follow below step to get response fields from Payment Gateway.

```
//To get Error
$_GET[“ErrorText”]
//To get Payment ID
$_GET['paymentid'];
/** End of Response Processing**/
```

/ End of Response Processing**/**

Success Response Parameters	Description
$myObj->getResult()	**CAPTURED** for Purchase transaction. Other than above value considered the transaction as failure
$myObj->getDate()	Contains Postdate
$myObj->getRef()	Transaction reference no.

$myObj->getTrackId()	Containts merchant unique track id
$myObj->getTransId()	PG Transactuion ID
$myObj->getUdf2() $myObj->getUdf3() $myObj->getUdf4() $myObj->getUdf5()	UDF Values sent during initial leg
$myObj->getPaymentId()	PG PaymentID
$myObj->getAmt()	Transaction amount
$myObj->getAuthRespCode()	Contains Response code
$myObj->getAuth()	Contains Auth code
$myObj->getDate()	Contains Post Date

Failure Response Parameters (If $_GET[“ErrorText”] is empty**)**	**Description**
$myObj->getResult();	Contains the result code of PG/Switch decline cases **(NOT CAPTURED, DENIED BY RISK, HOST TIMEOUT)**
$myObj-> getTrackId();	Containts merchant unique transaction id
$myObj->getAmt()	Transaction amount
$myObj->getUdf2() $myObj->getUdf3() $myObj->getUdf4() $myObj->getUdf5()	UDF Values sent during initial leg
$myObj->getRef()	Transaction reference no.
$myObj->getTransId()	PG Transactuion ID.
$myObj->getPaymentId()	PG PaymentID

Failure Response Parameters (If $_GET[“ErrorText”] is not empty**)**	**Description**
$_GET[“ErrorText”]	Contains the error description of the error.
$_GET[“paymentid”]	PG PaymentID
$_GET[“trackid”]	Merchant track id
$_GET[“amt”]	Transaction amount
$_GET[“udf2”] $_GET[“udf3”] $_GET[“udf4”] $_GET[“udf5”]	UDF Values sent during initial leg

3.8. Transaction Request/Response Parameters

Bank/institution Hosted Transaction Request/Response Parameters (Purchase)

Request Parameters:

S.No	Field	Type	Max	Mandatory	Description
1.	id	Alphanumeric	255	Yes	TranPortal Identification Number. The Payment Gateway Bank/institution administrator issues the TranPortal ID to identify the merchant and terminal for transaction processing. (ex : 101001)
2.	password	Alphanumeric	15	Yes	TranPortal Password. The Payment Gateway Bank/institution administrator issues the TranPortal password to authenticate the merchant and terminal. Merchant data will be encrypted and password securely hidden as long as the merchant is issuing an https post for transmitting the data to Payment Gateway. (ex : merchantdemo123)
3.	action	Numeric	1	Yes	Transaction Action Type, "1" for Purchase and “4” for Authorization.
4.	amt	Numeric	10	Yes	Transaction Amount, the amount should not contain thousands separators or currency symbols. (ex : 11.250)
5.	currency	Numeric	3	Yes	The currency code of the transaction (ex : 414)

6.	trackid	Alphanumeric	40	No	A unique tracking id issued by the merchant's system which is stored with the transaction. (Avoid spaces and extended characters. BENEFIT recommends merchant to use numeric format) (ex : trc433123687)
7.	language	Alpha	3	Yes	The language in which Payment Page has to be presented. For English - EN For Arabic – AR
8.	responseURL	Alphanumeric	255	Yes	The merchant URL where Payment Gateway send the authorization response. (ex: http://10.44.71.46/demomerchant/successresult.jsp)
9.	errorURL	Alphanumeric	255	Yes	The merchant URL where Payment Gateway send the response in case any error while processing the transaction. (ex: http://10.44.71.46/demomerchant/errorresult.jsp)
10	Udf1	Alphanumeric	255	No	Always keep it empty
11.	udf2	Alphanumeric	255	No	Allows Merchant to send any additional information
12.	udf3	Alphanumeric	255	No	Allows Merchant to send any additional information.
13.	udf4	Alphanumeric	255	No	Allows Merchant to send any additional information
14.	udf5	Alphanumeric	255	No	Allows Merchant to send any additional information

Response Parameters:

S.No	Field	Type	Description
1.	paymentid	Numeric	Unique Id generated by Payment Gateway. Merchant MUST map this payment ID with merchant track Id and transaction amount for future transaction reference. (ex : 501201906013287030)
2.	result	Alpha	Describes the result of the transaction (ex: CAPTURED).
3.	auth	Alphanumeric	The resulting authorization number of the transaction from the issuing bank/institution. This number or series of letters is used for referential purposes by some acquiring/issuing bank/institution/institutions and should be stored properly. (ex: 099673)
4.	amt	Numeric	Transaction Amount as sent by merchant in the request
5.	ref	Alphanumeric	The resulting reference number of the transaction. This number or series of letters is used for referential purposes by some acquiring/issuing bank/institution/institutions and should be stored properly.
6.	postdate	Numeric	Transaction Date in the format of the authorization system, Post Date is business logic date of respective issuing bank/institution and may not be same as the actual transaction date.

7.	trackid	Alphanumeric	Track ID value that was sent by merchant in the transaction request
8.	tranid	Numeric	Unique Transaction ID generated by Payment Gateway, this transaction ID is required for any future support/subsequent transaction (Capture) to be performed by merchant on Payment Gateway, merchant should store this parameter properly
9.	ErrorText	Alphanumeric	Describes the error with Error code. (ex. IPAY0100126-Brand not enabled)
10.	Error	Alphanumeric	Error code of the error. (ex. IPAY0100126)
11.	authRespCode	Alphanumeric	This is Response Code i.e. the reason code for the transaction. Merchant can use this code for additional validation of the transaction and can also use this response code field along with the reason code description list. (ex:00)
12	Udf1	Alphanumeric	Should be empty
13.	udf2	Alphanumeric	Udf field sent in the request
14.	udf3	Alphanumeric	Udf field sent in the request

15.	udf4	Alphanumeric	Udf field sent in the request
16.	udf5	Alphanumeric	Udf field sent in the request

Merchant Hosted Transaction Request/Response Parameters (Purchase)

Request Parameters:

S.No	Field	Type	Max	Mandatory	Description
1.	id	Alphanumeric	255	Yes	TranPortal Identification Number. The Payment Gateway Bank/institution administrator issues the TranPortal ID to identify the merchant and terminal for transaction processing. (ex : 101001)
2.	password	Alphanumeric	15	yes	TranPortal Password. The Payment Gateway Bank/institution administrator issues the TranPortal password to authenticate the merchant and terminal. Merchant data will be encrypted and password securely hidden as long as the merchant is issuing an https post for transmitting the data to Payment Gateway. (ex : merchantdemo123)

3.	action	Numeric	1	Yes	Describes the transaction type 1-Purchase
4.	amt	Numeric	10	Yes	Transaction Amount as sent by merchant in the request
5.	currency	Numeric	3	No	The currency code of the transaction (ex : 414)
6.	trackid	Alphanumeric	40	No	A unique tracking id issued by the merchant's system which is stored with the transaction. (Avoid spaces and extended characters. BENEFIT recommends merchant to use numeric format) (ex : trc433123687)
8	Udf1	Alphanumeric	255	No	Always keep it empty
9.	udf2	Alphanumeric	255	No	Allows Merchant to send any additional information
10.	udf3	Alphanumeric	255	No	Allows Merchant to send any additional information

11.	udf4	Alphanumeric	255	No	Allows Merchant to send any additional information

12.	udf5	Alphanumeric	255	No	Allows Merchant to send any additional information
13.	transid	Numeric	19	No	Transaction ID/Payment ID/Track ID/Sequence Number received from Payment Gateway for the original purchase transaction. This field is only to be passed when subsequent transactions are performed.
14.	responseURL	Alphanumeric	255	Yes	The merchant URL where Payment Gateway send the authorization response. (ex: http://10.44.71.46/demomerchant/successresult.jsp)
15.	errorURL	Alphanumeric	255	Yes	The merchant URL where Payment Gateway send the response in case any error while processing the transaction. (ex: http://10.44.71.46/demomerchant/errorresult.jsp)
16.	type	Alpha	2	Yes	Payment Instrument (Ex.”D”)

Response Parameters:

S. No	Field	Type	Description
1.	payid	Numeric	Original transaction’s payment id
2.	result	Alphabetic	Describes the result of the transaction

3.	auth	Alphanumeric	The resulting authorization number of the transaction from the issuing bank/institution. This number or series of letters is used for referential purposes by some acquiring/issuing bank/institution and should be stored properly. (ex: 099673)
4.	ref	Alphanumeric	The resulting reference number of the transaction. This number or series of letters is used for referential purposes by some acquiring/issuing bank/institution/institutions and should be stored properly.
5.	postdate	Numeric	Transaction Date in the format of the authorization system, Post Date is business logic date of respective issuing bank/institution and may not be same as the actual transaction date.
6.	error	Alphanumeric	Contains error code of error.
7.	trackid	Alphanumeric	Track ID value that was sent by merchant in the transaction request
8.	tranid	Numeric	Unique Transaction ID generated by Payment Gateway, this transaction ID is required for any future support/subsequent transaction (Capture) to be performed by merchant on Payment Gateway, merchant should store this parameter properly
9	Udf1	Alphanumeric	Should be empty
10.	udf2	Alphanumeric	Udf field sent in the request
11.	udf3	Alphanumeric	Udf field sent in the request
12.	udf4	Alphanumeric	Udf field sent in the request
13.	udf5	Alphanumeric	Udf field sent in the request

14.	authRespCode	Alphanumeric	This is Response Code i.e. the reason code for the transaction. Merchant can use this code for additional validation of the transaction and can also use this response code field along with the reason code description list. (ex:00)
15.	amt	Numeric	Transaction Amount as sent by merchant in the request

Chapter 4 Merchant Integration Process using REST APIS

4.1 Bank Hosted Transaction Flow(Purchase Transaction)

1. Customer visits the merchant site and selects the merchandise and confirms to pay using the payment gateway.
2. Merchant redirects the customer with the transaction data to the payment gateway.
3. Payment Gateway verifies the transaction request from the merchant and generates the payment ID and Payment URL.
4. Payment gateway will provide the response to merchant with Payment ID and Payment URL.
5. Merchant to redirect the Payment URL along with payment ID.
6. PG presents the hosted payment page to customer and prompts the customer to enter the card credentials.
7. Payment gateway will process the transaction with the respective switch.
8. Once the transaction complete, PG will verify the notification flag configured at merchant level and provide the response accordingly.

If Notification enabled at Merchant level,

1. Payment gateway will initiate server to server call to send the transaction status notification request to merchant and wait for the acknowledgement.
2. If PG receives the acknowledgment from merchant, then Payment Gateway logs the response and sends the final response to merchant. The final response will be URL redirection.
3. Merchant displays transaction result to customer.
4. In case, if PG does not receive the acknowledgment from merchant then PG will initiate the VOID transaction to respective scheme to reverse the transaction.

If Notification disabled at Merchant level,

1. Payment Gateway logs the response and sends final response to merchant and this will be URL redirection.
2. Merchant displays transaction result to customer.

Below are the prerequisites for merchants to integrate with Benefit Payment gateway

1. Tranportal ID
2. Tranportal Password
3. Resource Key
4. Payment gateway endpoint

Resource key is required for encryption and decryption. Bank user can fetech the details under **Bank Portal - View information - View Plugins** and share the resource key in a secured manner.

Payment gateway endpoint - https://test.benefit-gateway.bh/payment/API/hosted.htm

Below are the sample JSON request and response,

Please refer the appendix section for the detail description of request and Response parameters

Request from Merchant to Benefit Payment gateway:

```
[{
      "id":"IPAYlCR6qZF7q6w",
      “trandata”:”B38EAB17207C551A0DD86571A9E64486258D57B8926F2BF0A4F91A4FF3506A
E3EB829ABAEB12F9014D2C5BA66EB5808B947F3FADE1840FC16A2A704DBA33AEC2CA499D1B45F227
CCD18FA5E788CBA35D3E236F82AB6A4375E1BBAF710919E3A10A5B0F975A737D150FD54313641A46
A108DF8F6A49367F48197DED9C0BA0299F20D4959FD84EFC227D7D7E29AF6293DE55320F5A5E7173
E15AD114F0388D892AE8B310F3A2EFE42E864403763BC3B18684AD11D5F9828C8FB10558CAE2D6AF
1B52C556E53ED2961829F62F4589E305B53983CAAAD168ECC75EA938CCD4F7EC3471C0A8A2B14079
1C9135FA7AB1C21FCA35295583AF3E19CB4CFAEE27F175628566BE0D41BFA06CD9CC5779E637F22E
743CC9CF0AAC6C3877C7CA3BBCCF76706FB056E43689C8035F684C6B3DC00CCE03D1945B2DA0F6F1
1E7EC1EC09E4B6F731791EF28F28E34DAD0FB6E2958E482BE24A635B2779AE741FA0029A41FFD9DD
7E4E121307C65A567F5164B57D5EAED252681A065A1BE703076C29A1E636C75B646011DFE5A0A63C
CC164A3C1186B8FBD93E845D89B2A1DCCD6A366CAF77BFF7EDA627824A600B668216F17B1BFE8465
7C405DF95AD21957A9D1193D6D80BAE63E1296BEAAB26096EB78C557109554E438DDA997A9D85D15
B3EC680EBA5EBCDCB558A2D0DF9460B79BE88D063342CD6B0B3C26C3965962F25A9BD42CBFC2843E
F044C54153A8DD4FDB7D162270CDEA617A43E99BAA036FDACD7F6DC59E264BBD59”



}]
```

- Trandata will contain below parameters encrypted with AES algorithm with CBC Mode, PKCS5Padding with initialization vector value **PGKEYENCDECIVSPC** under Resource Key

Plain Trandata:

```
[{“amt”:”12.00”,”action”:”1”,”password”:”q@a68O$27@JLkcK”,”id”:”IPAYlCR6qZF7q6w”
,”currencycode”:”048”,”trackId”:”12345656789”,”udf1”:””,”udf2”:”udf2text”,”udf3”
:”udf3text”,”udf4”:”udf4text”,”udf5”:”udf5text”,
"responseURL":"http://www.companyName.com/testPayment",
"errorURL":"http://www.companyName.com/testPayment"}]
```

- Benefit Payment gateway internally validates the request and gives payment ID and payment page URL in the response in case of successful validation, if failure then error code and description will be provided. The below response will be in plain format and there is no encryption for the below. Merchant can directly parse the response-based status and result fields as mentioned below.

```
Success:
[{

"status": "1",
"result":"100201931620827468:https://test.BENEFIT-Gateway.bh"//Payment
ID:Paymentpage URL
“error”:null,
“errorText”:null
}]

Failure:
[{
"status": "2",
"error":"IPAY0100124”,
“errorText”:” Problem occurred while validating transaction data”
“result”:null
}]
```

- If success, Merchant will get the payment page URL like the below sample from result

  https://test.BENEFIT-Gateway.bh?PaymentID=100201931620827468

  now merchant need to redirect to the above PG url, PG will process the transaction and send response back to merchant.

  - If Merchant notification is disabled, then Benefit Payment gateway will provide the final response in URL redirection. Below is the sample response from Benefit PG to merchant

```
/** Response received from Payment Gateway to Merchant **/
Merchant can get below trandata using request.getParameter(“trandata”); method
1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C78340913929D9087
9615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313637690
531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9A
AF38E044F0A3469434850677825704053 3E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

Plain trandata in response

```
[{“paymentId”:10029375849,”result”:”captured”,”amt”:10.55,”date”:1221,
”ref”:”935110000001”,”udf1”:”udf1text”,”udf2”:”udf2text”,”udf3”:”udf3text”,”udf4”:”ud
f4text”,”udf5”:”udf5text”,”trackId”,”3423423”,”transId”:1242345345234,“authRespC
ode”:”00” ,”authCode”:”000000”}]
```

```
If merchant notification enabled for the merchant, then follow the below
process.
```

Merchant Notification flow

- Once the transaction is processed, Benefit Payment gateway initiates the encrypted request with response parameters to notify the transactionresult. Merchant has to decrypt the response using the decrption process given below.
- Merchant acknowledge the notification request with a response page at their end which will have only the keyword “REDIRECT=” followed by the response URL.
- Once acknowledgment received from merchant, Benefit Payment Gateway provides the final encrypted response to merchant and this will be URL redirection.
- Merchant has to decrypt the response using the decrption process given below and show response to the customer.

Below is the sample merchant notification request and response,

Request to Merchant:

```
1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C78340913929D9087
9615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313637690
531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9A
AF38E044F0A34694348506778257040533E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

Merchant need to decrypt above trandata with terminal resource key which is provide by bank and merchant will send the final response url to PG.

Acknowledgment response url from merchant to PG:

```
REDIRECT=http://www.demomerchant.com/result.jsp
```

If Benefit payment gateway receives the acknowledgement from merchant, PG logs the re-sponse and provide the response back to merchant. This will be URL redirection.

Below is final response to merchant:

```
Merchant can get below trandata using request.getParameter(“trandata”); method
```

```
1C1A967D16C877543E0A1DD90D2933853F05BB0AA2E6B47BB77D27EEAF32EF02CF6A0A5643F31C78340913929D9087
9615DFA9CDCCBB03761B9AE87CD76FE8633E0A1DD90D29338545DA582B0F3500BA9375313637690
531C6D7F8F7489C7CD8B73F9EC7E1C622DDD06B0809A709C2CB2A6EFD72F36FEB044B73810204E69FC577300F9A
```

```
AF38E044F0A34694348506778257040533E4FD48A9C61DD83E906AB93110CE0E57A1C548FA589B8F8566FC9F
```

Merchant need to decrypt above trandata with terminal resource key which is provide by bank and send response to customer.

- If there is no acknowledgement from Merchant After initiating the merchant notification, PG will wait for the response based on the configured time and no acknowledgement from merchant then PG will void the transaction and update the response back to merchant error URL received in the initial API request.

  Response from PG to merchant if no acknowledgement from Merchant:

  Merchant can directly get the below parameter values by using request.getParameter() method

  paymentid
  Error
  ErrorText
  trackid
  amt

4.2 Sample Encryption and Decryption Code for JAVA / .NET

```
public static String encryptAES(String key,String encryptString) throws Exception{

          String AES_IV = “PGKEYENCDECIVSPC”;
Byte[] encryptedText=null;
         IvParameterSpecivspec=null;
         SecretKeySpecskeySpec=null;
          Cipher cipher=null;
         Byte[] text=null;
          String s=null;
         try{
               ivspec = new IvParameterSpec(AES_IV.getBytes("UTF-8"));
               skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
                cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
               cipher.init(Cipher.ENCRYPT_MODE, skeySpec,ivspec);
                text = encryptString.getBytes("UTF-8");
               encryptedText = cipher.doFinal(text);
                s = byteArrayToHexString(encryptedText);
          }catch(Exception e){
               e.printStackTrace();
          }
          finally
          {
               encryptedText=null;
               ivspec=null;
               skeySpec=null;
                cipher=null;
                text=null;
          }
           return s.toUpperCase();
    }
```

Note: Before encrypting encryptString value. merchant needs to encode the value with URL Encoder.

```
public static String decryptAES(String key,StringencryptedString) throws Exception{
          String AES_IV = “PGKEYENCDECIVSPC”;
         SecretKeySpecskeySpec=null;
         IvParameterSpecivspec=null;
          Cipher cipher =null;
         Byte[] textDecrypted=null;
         Try{
               Byte[] b = hexStringToByteArray(encryptedString);
               skeySpec = new SecretKeySpec(key.getBytes("UTF-8"),"AES");
               ivspec = new IvParameterSpec(AES_IV.getBytes("UTF-8"));
                cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
               cipher.init(Cipher.DECRYPT_MODE, skeySpec,ivspec);
               textDecrypted = cipher.doFinal(b);
          }catch(Exception e){
               e.printStackTrace();
          }
          finally
          {
               skeySpec=null;
               ivspec=null;
                 cipher =null;
          }
```

```
            return(new String(textDecrypted));
        }
```

Note: After decrypting encryptedString value. Merchant needs to decode the textDecrypted value with URL Decoder.

4.3 Sample Encryption and Decryption Code For JAVASCRIPT

```
function aesEncrypt(trandata,key)
{
   var iv = "PGKEYENCDECIVSPC";
   var rkEncryptionIv = aesjs.utils.utf8.toBytes(iv);
   var enckey= aesjs.utils.utf8.toBytes(key);
   var aesCtr = new aesjs.ModeOfOperation.cbc(enckey, rkEncryptionIv);
   var textBytes = aesjs.utils.utf8.toBytes(trandata);
   var encryptedBytes = aesCtr.encrypt(aesjs.padding.pkcs7.pad(textBytes));
   var encryptedHex = aesjs.utils.hex.fromBytes(encryptedBytes);
   return encryptedHex;
}
```

Note: Before encrypting trandata value. merchant needs to encode the value with URL Encoder.

```
function AESdecryption(encryptedHex,key)
{
   var iv = "PGKEYENCDECIVSPC";
var enckey= aesjs.utils.utf8.toBytes(key);
var rkEncryptionIv = aesjs.utils.utf8.toBytes(iv);
var encryptedBytes = aesjs.utils.hex.toBytes(encryptedHex);
var aesCbc = new aesjs.ModeOfOperation.cbc(enckey, rkEncryptionIv);
var decryptedBytes = aesCbc.decrypt(encryptedBytes);
var decryptedText = aesjs.utils.utf8.fromBytes(decryptedBytes);
return decryptedText;
}
```

Note: After decrypting encryptedHex value. Merchant needs to decode the decryptedText value with URL Decoder.

4.4 Sample Encryption and Decryption Code For PHP Encryption:

```
function encryptAES($str,$key)
{
$str = $this->pkcs5_pad($str);
$ivlen = openssl_cipher_iv_length($cipher="aes-256-cbc");
$iv="PGKEYENCDECIVSPC";
$encrypted = openssl_encrypt($str, "aes-256-cbc",$key, OPENSSL_ZERO_PADDING, $iv);
$encrypted = base64_decode($encrypted);
$encrypted=unpack('C*', ($encrypted));
$encrypted=$this->byteArray2Hex($encrypted);
$encrypted = urlencode($encrypted);
return $encrypted;
}
```

Note: Before encrypting transaction data, data needs to be encoded using URL-Encoder

Decryption:

```
function decryptAES ($codcode, $)
{
$code = $this->hex2ByteArray(trim($code));
$code=$this->byteArray2String($code);
$iv = "PGKEYENCDECIVSPC";
$code = base64_encode($code);
$decrypted = openssl_decrypt($code, 'AES-256-CBC', $key, OPENSSL_ZERO_PADDING,
$iv);
return $this->pkcs5_unpad($decrypted);
}
```

Note: After decrypting transaction data needs to be decoded using URL-Decoder

4.5 Transaction Resquest and Response

Bank Hosted Purchase Transaction request and response Parameters

Request from Merchant to Benefit Payment gateway:

S. No	Fields	Mandatory/Optional	Field Type	Description
1	id	M	Alphanumeric	Tranportal ID. Merchant can download the Tranportal id from Merchant portal
2	trandata	M	Alphanumeric	All the below request parameters encrypted and pass the encrypted value in trandata. For Ex: [{“amt”:”12.00”,”action”:”1”,”password”:”q@a68O$27@JLkcK”,”id”:”IPAYlCR6qZF7q6w”,”currencycode”:”048”,”trackId”:”12345656789”,”udf1”:”udf1text”,”udf2”:”udf2text”,”udf3”:”udf3text”,”udf4”:”udf4text”,”udf5”:”udf5text”, "responseURL":"http://www.companyName.com/testPayment", "errorURL":"http://www.companyName.com/testPayment"}]

Detailed description of Plain Trandata request parameters,

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	amt	M	Numeric	Transaction amount
2	action	M	Numeric	It defines the transactions actions Purchase: 1 Refund: 2 Void: 3 Inquiry: 8
3	password	M	Alphanumeric	Tranportal password. Merchant download the same

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
				in merchant portal.
4	id	M	Alphanumeric	Tranportal ID. Merchant download the same in merchant portal
5	currencycode	M	Numeric	3-digit currency code of BHD. Ex:048
6	trackId	M	Numeric	Merchant unique reference no
7	udf1	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
8	udf2	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
9	udf3	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
10	udf5	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
11	responseURL	M	Alphanumeric	The merchant success URL where Payment Gateway send the notification request.
12	errorURL	M	Alphanumeric	The merchant error URL where Payment Gateway send the response in case any error while Processing the transaction.

Initial Response from PG to Merchant,

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	status	M	Numeric	If the request validation success, then status will be '1'. If the validation failed, then status will be '2'
2	result	M	Alphanumeric	It contains payment ID and Payment URL if the validation success else this will be null
3	error	C	Alphanumeric	If validation failed, then Payment gateway will provide the respective error code
4	errorText	C	Alphanumeric	If validation failed, then Payment gateway will provide the respective error description

Notification request from PG to Merchant,

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	trandata	M	Alphanumeric	All the below response parameters encrypted and send the encrypted value in trandata Ex:

				[{“paymentId”:100201935166676976,”result”:”CAPTURED”,”ref”:”935110000001”,”transId”:201935166561122,”date”:1217,”trackId”:”1003383844”,”udf1”:””,”udf2”:””,”udf3”:”8870091137”,”udf4”:”FC”,”udf5”:”Tidal5”,”amt”:”70.0,”authRespCode”:”00”,”authCode”:”000000”}]

Detailed description of Plain trandata parameters

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	paymentId	M	Numeric	Unique ID generated by payment gateway. Based on this payment Id merchant can match the final URL redirection response
2	result	M	Alphanumeric	Transaction status
3	ref	M	Numeric	Transaction reference number (RRN)
4	transId	M	Numeric	Unique transaction Id generated by Payment gateway and merchant can use this id for initiating supported transactions (Void, refund and inquiry)
5	date	M	Numeric	Transaction date and time
6	trackId	M	Numeric	Merchant unique reference no
7	udf1	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
				then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
8	udf2	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
9	udf3	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
10	udf5	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
11	amt	M	Numeric	Transaction amount
12	authRespCode	M	Numeric	Auth response code provided by PG
13	authCode	M	Numeric	Auth code provided by PG

Notification response from Merchant to Payment gateway

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	Response URL	M	Alphanumeric	Merchant response URL for which PG will provide the final URL redirection response

Final URL redirection response from payment gateway to Merchant

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	paymentid	M	Numeric	Unique payment Id generated by PG and merchant can use this ID to match the response from PG
2	trandata	C	Alphanumeric	All the below response parameters encrypted and send the encrypted value in trandata Ex: [{“paymentId”:100201935166676976,”result”:”CAPTURED”,”ref”:”935110000001”,”transId”:201935166561122,”date”:1217,”trackId”:”1003383844”,”udf1”:””,”udf2”:””,”udf3”:”8870091137”,”udf4”:”FC”,”udf5”:”Tidal5”,”amt”:”70.0,”authRespCode”:”00”, authCode”:”12345”}]
3	Error	C	Alphanumeric	If any error, PG will send the error code
4	ErrorText	C	Alphanumeric	If any error, PG will send the error description
5	trackid	O	Alphanumeric	Merchant track ID
6	tranid	O	Numeric	Transaction ID
7	auth	O	Numeric	Auth code provided by PG
8	ref	O	Numeric	Transaction Reference number

9	amount	O	Numeric	Transaction Amount
10	result	O	Alphanumeric	Transaction status
11	udf1	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
12	udf2	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
13	udf3	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
14	udf4	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
15	udf5	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response.

				Merchant should ensure that field is left blank when no data needs to be passed.
16	udf6	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
17	udf7	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
18	udf8	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
19	udf9	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
20	udf10	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that

				field is left blank when no data needs to be passed.
21	udf11	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
22	udf12	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
23	udf13	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
24	udf14	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
25	udf15	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no

				data needs to be passed.
26	postdate	O	Date	Transaction Date
27	avr	O	Alphabetic	Value will be Y or N
28	authRespCode	O	Alphanumeric	Authorization Response Code From PG
29	threeDSServerTranID	O	Alphanumeric	Used for 3D Secured Transactions
30	dsTranID	O	Alphanumeric	Used for 3D Secured Transactions
31	acsTranID	O	Alphanumeric	Used for 3D Secured Transactions

Detailed description of Plain trandata parameters

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
1	paymentId	M	Numeric	Unique ID generated by payment gateway. Based on this payment Id merchant can match the final URL redirection response
2	result	M	Alphanumeric	Transaction status
3	ref	M	Numeric	Transaction reference number (RRN)
4	transId	M	Numeric	Unique transaction Id generated by Payment gateway and merchant can use this id for initiating supported transactions (Void, refund and inquiry)
5	date	M	Numeric	Transaction date and time
6	trackId	M	Numeric	Merchant unique reference no
7	udf1	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response.

S. No	Fields	Mandatory /Optional /Conditional	Field Type	Description
				Merchant should ensure that field is left blank when no data needs to be passed.
8	udf2	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
9	udf3	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
10	udf5	O	Alphanumeric	The user (merchant) defines these fields. The field data is passed along with a transaction request and then returned in the transaction response. Merchant should ensure that field is left blank when no data needs to be passed.
11	amt	M	Numeric	Transaction amount
12	authRespCode	M	Numeric	Auth response code provided by PG
13	authCode	M	Numeric	Auth code provided by PG

Action Codes

Action	Action code

Purchase	1
Refund	2
Inquiry	8
Void	3

Result

Result	Description
CAPTURED	Captured result will be considered as transaction success
NOT CAPTURED	This will be considered as transaction failure
VOIDED	Success for Void transaction
DENIED BY RISK	If the Risk validation failed, then PG will decline the transaction with this result
HOST TIMEOUT	If there is no response from respective interchange during authorization, then PG will provide the Host timeout result.

Chapter 5 Response Code and Description:

Response Code	Error Description
14	Invalid card number
33	Expired card
36	Restricted card
38	Allowable PIN tries exceeded
51	Not sufficient funds
54	Expired card
55	Incorrect PIN
61	Exceeds withdrawal amount limit
62	Restricted Card
65	Exceeds withdrawal frequency limit
75	Allowable number PIN tries exceeded
76	Ineligible account

78	Refer to Issuer
91	Issuer is inoperative

Chapter 6 Error Code and Description:

Error Code	Error Description
IPAY0100001	Missing error URL.
IPAY0100002	Invalid error URL.
IPAY0100003	Missing response URL.
IPAY0100004	Invalid response URL.
IPAY0100005	Missing tranportal id.
IPAY0100006	Invalid tranportal ID.
IPAY0100007	Missing transaction data.
IPAY0100008	Terminal not enabled.

IPAY0200001	Problem occurred while getting terminal.
IPAY0100009	Institution not enabled.
IPAY0200002	Problem occurred while getting institution details.
IPAY0100010	Institution has not enabled for the encryption process.
IPAY0200003	Problem occurred while getting merchant details.
IPAY0100011	Merchant has not enabled for encryption process.
IPAY0100012	Empty terminal key.
IPAY0100013	Invalid transaction data.
IPAY0100014	Terminal Authentication requested with invalid tranportal ID data.
IPAY0100015	Invalid tranportal password.
IPAY0100016	Password security not enabled.
IPAY0200004	Problem occurred while getting password security rules.
IPAY0200005	Problem occurred while updating terminal details.
IPAY0100018	Terminal password expired.
IPAY0200006	Problem occurred while verifying tranportal password.
IPAY0100020	Invalid action type.
IPAY0100022	Invalid currency.
IPAY0100023	Missing amount.
IPAY0100025	Invalid amount or currency.
IPAY0100026	Invalid language id
IPAY0100028	Invalid user defined field1.
IPAY0100029	Invalid user defined field2.
IPAY0100031	Invalid user defined field4.
IPAY0100032	Invalid user defined field5.
IPAY0100033	Terminal action not enabled.
IPAY0100034	Currency code not enabled.
IPAY0100035	Problem occurred during merchant hashing process.
IPAY0100036	UDF Mismatched'
IPAY0100038	Unable to process the request.

IPAY0100039	Invalid payment id.
IPAY0200009	Problem occurred while getting payment details.
IPAY0100041	Payment details missing.
IPAY0100042	Transaction time limit exceeds.
IPAY0200011	Problem occurred while getting IP block details.
IPAY0100043	IP address is blocked already
IPAY0100044	Problem occurred while loading payment page.
IPAY0100045	Denied by Risk
IPAY0200013	Problem occurred while updating description details in payment log.
IPAY0100047	Payment Page validation failed due to invalid Order Status:
IPAY0100049	Transaction declined due to exceeding OTP attempts
IPAY0200015	Problem occurred while getting terminal details.
IPAY0100050	Invalid terminal key.
IPAY0100051	Missing terminal key.
IPAY0100053	Problem occurred while processing direct debit.
IPAY0100054	Payment details not available
IPAY0100056	Instrument not allowed in Terminal and Brand
IPAY0200016	Problem occurred while getting payment instrument.
IPAY0200018	Problem occurred while getting transaction details
IPAY0100057	Transaction denied due to invalid processing option action code
IPAY0100058	Transaction denied due to invalid instrument
IPAY0100059	Transaction denied due to invalid currency code.
IPAY0100060	Transaction denied due to missing amount.
IPAY0100061	Transaction denied due to invalid amount.
IPAY0100062	Transaction denied due to invalid Amount/Currency.
IPAY0100063	Transaction denied due to invalid track ID');
IPAY0100064	Transaction denied due to invalid UDF1
IPAY0100065	Transaction denied due to invalid UDF2
IPAY0100066	Transaction denied due to invalid UDF3

IPAY0100067	Transaction denied due to invalid UDF4
IPAY0100068	Transaction denied due to invalid UDF5
IPAY0100069	Missing payment instrument.
IPAY0100070	Transaction denied due to failed card check digit calculation.
IPAY0100074	Transaction denied due to missing expiry year.
IPAY0100075	Transaction denied due to invalid expiry year.
IPAY0100076	Transaction denied due to missing expiry month.
IPAY0100077	Transaction denied due to invalid expiry month.
IPAY0100078	Transaction denied due to missing expiry day.
IPAY0100079	Transaction denied due to invalid expiry day.
IPAY0100080	Transaction denied due to expiration date.
IPAY0100081	Card holder name is not present
IPAY0100082	Card address is not present
IPAY0100083	Card postal code is not present
IPAY0100085	Electronic Commerce Indicator is invalid
IPAY0100087	Card pin number is not present
IPAY0100088	Empty mobile number.
IPAY0100089	Invalid mobile number.
IPAY0100091	Invalid MMID.
IPAY0100092	Empty OTP number.
IPAY0100093	Invalid OTP number.
IPAY0100095	Terminal inactive.
IPAY0100096	IMPS for Institution Not Active for Transaction request, Institution :
IPAY0100098	Terminal Action not enabled for Transaction request, Terminal “termid” ,Tran Action : “action”',-'opted action was not supported by that terminal
IPAY0100099	Terminal Payment Instrument not enabled for Transaction request, Terminal “termid ” , Tran Instrument :"PAYMENT_INSTRUMENT"
IPAY0100100	Problem occurred while authorize
IPAY0200019	Problem occurred while getting risk profile details
IPAY0100102	Denied by risk : Maximum Floor Limit Check - Fail

IPAY0100103	Transaction denied due to Risk : Maximum transaction count
IPAY0100104	Transaction denied due to Risk : Maximum processing amount
IPAY0200022	Problem occurred while getting currency.
IPAY0100105	Action type not supported by maestro brand.
IPAY0100106	Invalid payment instrument
IPAY0200024	Problem occurred while getting brand rules details.
IPAY0100107	Instrument not enabled.
IPAY0200025	Problem occurred while getting terminal details.
IPAY0100109	Invalid subsequent transaction, payment id is null or empty.
IPAY0200026	Problem occurred while getting transaction log details.
IPAY0200027	Missing encrypted card number.
IPAY0100111	Card decryption failed.
IPAY0100113	“transaction id” is a subsequent transaction, but original transaction id is invalid :
IPAY0100114	Duplicate Record
IPAY0100115	Transaction denied due to missing original transaction id.
IPAY0100116	Transaction denied due to invalid original transaction id.
IPAY0100118	Transaction denied due to card number length error
IPAY0100119	Transaction denied due to invalid card number
IPAY0100120	Transaction denied due to invalid payment instrument for brand data.
IPAY0100121	Transaction denied due to invalid card holder name.
IPAY0100122	Transaction denied due to invalid address.
IPAY0100123	Transaction denied due to invalid postal code.
IPAY0100124	Problem occurred while validating transaction data
IPAY0100125	Payment instrument not enabled.
IPAY0100126	Brand not enabled.
IPAY0100129	Transaction denied due to Merchant ID mismatch
IPAY0100130	Transaction denied due to Terminal ID mismatch
IPAY0100131	Transaction denied due to Payment Instrument mismatch
IPAY0100132	Transaction denied due to Currency Code mismatch

IPAY0100133	Transaction denied due to Card Number mismatch
IPAY0100134	Transaction denied due to invalid Result Code
IPAY0100135	Problem occurred while doing perform action code reference id (Validate Original Transaction)
IPAY0200028	Problem occurred while loading default institution configuration (Validate Original Transaction)
IPAY0100136	Transaction denied due to previous capture check failure (Validate Original Transaction)
IPAY0100138	Transaction denied due to capture amount versus auth amount check failure (Validate Original Transaction)
IPAY0100141	Transaction denied due to authorization already captured (Validate Original Transaction)
IPAY0100142	Problem occurred while validating original transaction
IPAY0200030	No external connection details for Extr Conn id :
IPAY0200031	Alternate external connection details not found for the alt Extr Conn id :
IPAY0100143	Transaction action is null
IPAY0100144	ISO MSG is null. See log for more details!
IPAY0100145	Problem occurred while loading default messages in ISO Formatter
IPAY0100147	Problem occurred while formatting purchase request in B24 ISO Message Formatter
IPAY0100148	Problem occurred while hashing E-com PIN
IPAY0100150	Problem occurred while formatting Reverse purchase request in B24 ISO Message Formatter
IPAY0100152	Message Formatter
PAY0100153	Problem occurred while formatting Capture request in B24 ISO Message Formatter
IPAY0100155	Problem occurred while formatting reverse authorization request in B24 ISO Message Formatter
IPAY0100156	Problem occurred while formatting Reverse Capture request in B24 ISO Message Formatter
IPAY0100157	Problem occurred while formatting vpas capture request in B24 ISO Message Formatter
IPAY0100159	External message system error
IPAY0100160	Unable to process the transaction.
IPAY0100162	Problem occurred while validating IMPS

IPAY0100163	Problem occurred during transaction.
IPAY0100164	Transaction Not Processed due to Wrong ECI value
IPAY0100166	Transaction Not Processed due to Empty Authentication Status
IPAY0100167	Transaction Not Processed due to Invalid Authentication Status
IPAY0100168	Transaction Not Processed due to Empty Enrollment Status
IPAY0100169	Transaction Not Processed due to Invalid Enrollment Status
IPAY0200036	Problem occurred while getting amex payment log details.
IPAY0100172	Problem occurred while converting amount.
IPAY0100174	Problem occurred while calling amex web service.
IPAY0100017	Inactive terminal.
IPAY0100019	Invalid log in attempt.
IPAY0100021	Missing currency.
IPAY0100024	Invalid amount.
IPAY0100027	Invalid track id.
IPAY0100030	Invalid user defined field3.
IPAY0200007	Problem occurred while validating payment details
IPAY0200008	Problem occurred while verifying payment details.
IPAY0100037	Payment id missing.
IPAY0100040	Transaction in progress in another tab/window.
IPAY0200010	Problem occurred while updating payment details.
IPAY0200012	Problem occurred while updating payment log IP details.
IPAY0100046	Payment option not enabled.
IPAY0100048	Cancelled
IPAY0200014	Problem occurred during merchant response.
IPAY0100052	Problem occurred during merchant response encryption.
IPAY0100055	Invalid Payment Status
IPAY0200017	Problem occurred while getting payment instrument list
IPAY0100090	Empty MMID.
IPAY0100094	Sorry, this instrument is not handled

IPAY0100097	IMPS for Terminal Not Active for Transaction request, Terminal :
IPAY0100101	Denied by risk: Risk Profile does not exist
IPAY0200020	Problem occurred while performing transaction risk check
IPAY0200021	Problem occurred while performing risk check
IPAY0200023	Problem occurred while determining payment instrument
IPAY0100108	Perform risk check: Failed
IPAY0100110	Invalid subsequent transaction, Tran Ref id is null or empty.
IPAY0100112	Problem occurred in method loading original transaction data (card number, exp month / year) for orig_tran_id
IPAY0100117	Transaction denied due to missing card number.
IPAY0100137	Transaction denied due to credit amount greater than auth amount check failure (Validate Original Transaction)
IPAY0200029	Problem occurred while getting external connection details.
IPAY0200032	Problem occurred while getting external connection details for Extr Conn id :
IPAY0100146	Problem occurred while encrypting PIN
IPAY0100149	Invalid PIN Type
IPAY0100151	Problem occurred while formatting Credit request in B24 ISO Message Formatter
IPAY0100154	Problem occurred while formatting Reverse Credit request in B24 ISO Message Formatter
IPAY0100158	Host timeout
IPAY0100161	Merchant is not allowed for encryption process.
IPAY0100165	Transaction Not Processed due to Empty ECI value
IPAY0100176	Decrypting transaction data failed.
IPAY0100177	Invalid input data received.
IPAY0100178	Merchant encryption enabled.
IPAY0100179	IVR not enabled.
IPAY0100180	Authentication not available.
IPAY0100181	Card encryption failed.
IPAY0200037	Error occurred while getting Merchant ID

IPAY0100182	Vpas merchant not enabled.
IPAY0200038	Problem occurred while getting vpas merchant details.
IPAY0100183	Error occurred Due to byte PAReq is null
IPAY0100184	Error occurred while Parsing PAReq
IPAY0100185	Problem occurred while authentication
IPAY0100186	Encryption enabled.
IPAY0100187	Customer ID is missing for Faster Checkout.
IPAY0100188	Transaction Mode(FC) is missing for Faster Checkout.
IPAY0100189	Transaction denied due to brand directory unavailable
IPAY0200039	Problem occurred while getting Faster Checkout details.
IPAY0100190	Transaction denied due to Risk : Maximum transaction count
IPAY0100191	Denied by risk : Negative Card check - Fail
IPAY0100192	Transaction Not Processed due to Empty XID
IPAY0100193	Transaction Not Processed due to invalid XID
IPAY0100202	Error occurred in Determine Payment Instrument
IPAY0100194	Transaction denied due to Risk : Minimum Transaction Amount processing
IPAY0100195	Transaction denied due to Risk : Maximum credit processing amount
IPAY0100196	Transaction denied due to Risk : Maximum processing amount
IPAY0100197	Transaction denied due to Risk : Maximum debit amount
IPAY0100198	Transaction denied due to Risk : Transaction count limit exceeded for the IP
IPAY0100199	Transaction denied due to previous credit check failure (Validate Original Transaction)
IPAY0100200	Denied by risk : Negative BIN check - Fail
IPAY0100201	Denied by risk: Declined Card check – Fail
IPAY0100203	Problem occurred while doing perform transaction
IPAY0100204	Missing payment details
IPAY0100205	Problem occurred while getting PARES details
IPAY0100206	Problem occurred while getting currency minor digits
IPAY0100207	Bin range not enabled

IPAY0100208	Action not enabled
IPAY0100209	Institution config not enabled
IPAY0100210	Problem occurred during veres process
IPAY0100211	Problem occurred during pareq process
IPAY0100212	Problem occurred while getting veres
IPAY0100213	Problem occurred while processing the hosted transaction request
IPAY0100214	Problem occurred while verifying tranportal id
IPAY0100215	Invalid tranportal id
IPAY0100216	Invalid data received
IPAY0100217	Invalid payment detail
IPAY0100218	Invalid brand id
IPAY0100219	Missing card number
IPAY0100220	Invalid card number
IPAY0100221	Missing card holder name
IPAY0100222	Invalid card holder name
IPAY0100225	Missing card expiry year
IPAY0100226	Invalid card expiry year
IPAY0100227	Missing card expiry month
IPAY0100228	Invalid card expiry month
IPAY0100229	Invalid card expiry day
IPAY0100230	Card expired
IPAY0100231	Invalid user defined field
IPAY0100232	Missing original transaction id
IPAY0100233	Invalid original transaction id
IPAY0100242	RC_UNAVAILABLE
IPAY0100243	NOT SUPPORTED
IPAY0100244	Payment Instrument Not Configured
IPAY0100245	Problem occurred while sending/receiving ISO message
IPAY0100246	Problem occurred while doing perform ip risk check

IPAY0100248	Problem occurred while validating PARES message format
IPAY0100249	Merchant response url is down
IPAY0100250	Payment details verification failed
IPAY0100251	Invalid payment data
IPAY0100253	Problem occurred while cancelling the transaction
IPAY0100254	Merchant not enabled
IPAY0100255	External connection not enabled
IPAY0100256	Payment encryption failed
IPAY0100257	Brand rules not enabled
IPAY0100258	Certification verification failed
IPAY0100259	Problem occurred during merchant hashing process
IPAY0100260	Payment option(s) not enabled
IPAY0100261	Payment hashing failed
IPAY0100263	Transaction details not available
IPAY0200047	Problem occurred while getting details from VPASLOG table for payment id
IPAY0200048	Problem occurred while getting details from VPASLOG table
IPAY0200049	Card number is null. Unable to update risk factors in negative card table & declined card table
IPAY0200050	Problem occurred while updating risk in negative card details
IPAY0200051	Problem occurred while updating risk in declined card table
IPAY0200052	Problem occurred while updating risk factor
IPAY0200053	Problem occurred while updating payment log currency details
IPAY0200054	Problem occurred while inserting currency conversion currency details
IPAY0200055	Problem occurred while updating currency conversion currency details
IPAY0200068	Problem occurred while validating IP address blocking
IPAY0200069	Problem occurred while updating payment log card details
IPAY0200070	Problem occurred while updating ipblock details
IPAY0200071	Problem occurred during authentication
IPAY0200072	Payment log details not available

IPAY0100284	Invalid subsequent transaction, track id is null or empty
IPAY0100285	Transaction denied due to invalid original transaction
IPAY0100287	Terminal Action not enabled for Transaction request, Terminal
IPAY0200056	Problem occurred while getting brand details
IPAY0200057	Problem occurred while getting external connection details
IPAY0200062	Problem occurred while getting brand rules details
IPAY0200063	Problem occurred while updating payment log process code details
IPAY0200064	Problem occurred while updating payment log process code and ip details
IPAY0200065	Problem occurred while updating payment log description details
IPAY0200066	Problem occurred while updating payment log instrument details
IPAY0200067	Problem occurred while updating payment log udf Fields
IPAY0100289	Transaction denied due to Risk: Maximum credit amount
PAY0100283	Problem occurred in determine payment instrument

Chapter 7 Best Practices:

a) The Merchant should mandatory maintain logs for each transaction as mentioned below

a. The parameters before setting the values in the respective variable.
b. Request from the merchant server to Payment Gateway
c. Response that is received from the Payment Gateway in the Merchant Response URL

b) The Merchant should maintain "OWASP" (Open Web Application Security Project) Top 10 recommendation in their web application. (These recommendations are available on www.owasp.org)

c) The Merchant should have the latest SSL security certificate in the payment request and receive webpage, if any. Always ensure that the SSL certificate is valid and has not expired. Such certificates should be as per the approved list of the Acquiring Bank/institution. Self singed certificates are not supported by Payment Gateway in Test and Production Environment.

d) The Merchant should mandatory complete the UAT and ensure all results are in line with the recommended response prior to going LIVE.

e) Any changes in the pages would need to be tested before moving to Production after proper communication to the Bank/institution personnel and receipt of approval. If the pages have a change in logic or transaction flow particularly, the Acquiring Bank/institution’s consent is Mandatory.

f) The transaction request and Response Handling: For ease in integration, “Sample/Demo pages “provided in the integration document are essentially for representation purposes only. The actual pages have to be necessarily developed and implemented by the Merchant’s development team and used in both the Test and Production environment. The Sample demo pages are provided for the logical understanding and transaction flow only. An ideal logical flow for the merchant to process the customer input data is to collect the shopping details of the customer such as transaction amount, merchant track id and other parameters and stored in a secure storage location and validated immediately against the details of shopping cart module.
h) Maintenance of Transaction Logs: It is essential for the transaction logs to be maintained in a secure storage location within the environment. This is crucial in order to trace transaction history in case of a dispute raised by a customer or even internal audit purposes. These logs should ideally include the customer IP address as well apart from the other transaction details