Value through Innovation

Worldpay Hosted Payments Integration Quick Start Guide

1. Overview

This document provides basic information for developers integrating hosted payments.

1.1. What Is Hosted Payments?

Hosted Payments allows an existing merchant with WorldPay credentials (**AccountID**, **AccountToken**, **AcceptorID**, **ApplicationID**) to enable their web server/e-commerce solution for payment acceptance by allowing the collection manually-typed credit card information into a secure WorldPay-hosted page. This credit card information can then be processed for a Sale, a Pre-Auth, or it can be tokenized for future payment requests.

1.2. IPS Payment Bridge

To use Hosted Payments, WorldPay must first certify the solution. To enable IPS customers immediate access to Hosted Payments without requiring them to certify their own web solution, IPS has certified a Payment Bridge with WorldPay. IPS customers can utilize the payment bridge to execute certified Hosted Payment requests with WorldPay. This is accomplished by posting transaction requests to **https://dev-paynow.itscocloud.com/execute** with all required and optional parameters, the IPS Payment Bridge will then forward to WorldPay using certified logic.

2. Transaction Parameters

The following list is of all parameters that can be passed to the IPS Payment Bridge. These parameters are defined by WorldPay.

Parameter Names	Type	Notes
AccountID	Required	
AccountToken	Required	
AcceptorID	Required	
License	Conditional	Needed if Live Transactions (isTest=0)
Amount	Required	
ReferenceNumber	Required	
DuplicateCheckDisableFlag	Optional	
DuplicateOverrideFlag	Optional	
MerchantSuppliedTransactionID	Optional	
TerminalID	Required	
LaneNumber	Required	
TransactionID	Conditional	For Completion (Type 10), Void (Type 12), Reverse (Type 11), Create Token (Type 14
PaymentAccountID	Conditional	Needed when updating **PaymentAccount**
PaymentAccountReferenceNumber	Conditional	Needed when creating or updating **PaymentAccount**
CardNumber	Conditional	For Customer Card Credit (Type 13)
ExpirationYear	Conditional	For Customer Card Credit (Type 13)
ExpirationMonth	Conditional	For Customer Card Credit (Type 13)
AddressEditAllowed	Optional	
BillingName	Optional	
BillingEmail	Optional	
BillingPhone	Optional	
BillingAddress1	Optional	Should be included with keyed transactions if AVS desired
BillingAddress2	Optional	
BillingCity	Optional	
BillingState	Optional	
BillingZipcode	Optional	Should be included with keyed transactions if AVS desired
ShippingName	Optional	
ShippingEmail	Optional	
ShippingPhone	Optional	
ShippingAddress1	Optional	
ShippingAddress2	Optional	

Parameter Names	Type	Notes
ShippingCity	Optional	
ShippingState	Optional	
ShippingZipcode	Optional	
CommercialCardCustomerCode	Optional	

TransactionSetup	Required	CreditCardSale = 1
		CreditCardAuthorization = 2
		CreditCardAVSOnly = 3
		PaymentAccountCreate = 7
		PaymentAccountUpdate = 8
		Completion = 10
		Reversal = 11
		Void = 12
		Credit = 13
		PaymentAccountCreateTransID = 14
		PaymentAccountQuery = 15
		PaymentAccountRecordCount= 16
		PaymentAccountRecordTokenReport= 17
		PaymentAccountRecordDelete= 18
		PaymentAccountSale= 19
		PaymentAccountAuthorization= 20

CVVRequired	Optional	
AutoReturn	Optional	
CompanyName	Optional	
LogoURL	Optional	
Tagline	Optional	
WelcomeMessage	Optional	
ReturnURL	Optional	
ReturnURLTitle	Optional	
OrderDetails	Optional	
ProcessTransactionTitle	Optional	
CustomCss	Optional	
isTest	Optional	
TipAmount	Optional	
SalesTaxAmount	Optional	
CashBackAmount	Optional	
Page	Optional	

3. Transaction Setup Types

The **TransactionSetup** parameter is required and can contain a valid value from 1-20. The types of transaction are as follows:

- **CreditCardSale = 1**
  - Credit Card Sale; requires **Amount** > 0.00.
- **CreditCardAuthorization = 2**
  - Credit Card Pre-Auth; requires **Amount** > 0.00.
- **CreditCardAVSOnly = 3**
  - AVS Check Only. Requires **Amount** = 0.00.
- **PaymentAccountCreate = 7**
  - Requests a payment token to be returned from the entered credit card info.
- **PaymentAccountUpdate = 8**
  - Updates a previous payment token.
- **Completion = 10**
  - Completion from a previous pre-auth. Requires **TransactionID**.
- **Reversal = 11**
  - Reversal from a previous transaction. Requires **TransactionID**.
- **Void = 12**
  - Void from a previous transaction. Requires **TransactionID**.
- **Credit = 13**
  - Credit from card. Requires credit card number.
- **PaymentAccountCreateTransID = 14**
  - Create payment token from previous transaction. Requires **TransactionID**.
- **PaymentAccountQuery = 15**
  - Query payment token.
- **PaymentAccountRecordCount = 16**
  - Payment token count.
- **PaymentAccountRecordTokenReport = 17**
  - Payment token report.
- **PaymentAccountRecordDelete = 18**
  - Delete payment token.
- **PaymentAccountSale = 19**
  - Perform sale from a payment token.
- **PaymentAccountAuthorization = 20**
  - Perform a pre-auth from payment token.

4. Request Format

The request format is as follows:

```
https://dev-
paynow.itscocloud.com/execute?<param1=val>&<param2=val>&<paramX=val>
```

The parameters should include, but not limited to:

Parameter	Description
TransactionSetup	**TransactionType** values 1-20.
ReturnURL	The web page that will receive the results from the transaction.
AutoReturn	**1** if the results are automatically returned to the **ReturnURL**, omit or set to **0** if results are displayed from WorldPay and another button is required to be clicked to execute the **ReturnURL**.
isTest	**1** if running test transactions, omit or set to **0** if running live transactions.
Credentials	Provide **AccountID**, **AccountToken**, **AcceptorID**, **ApplicationID** (test or live, depending on **isTest**).
License	Provide if live transactions.
ReferenceNumber	Required unique value. Omit and a random value will be generated.
LaneNumber	Required value. Does not have to be unique.
TerminalID	Required value. Does not have to be unique.
Amount	Required value. Amount to charge.
CVVRequired	**1** if CVV must be provided, omit or **0** if CVV is not required to be provided.
BillingAddress1	If AVS street address checking is desired, enter value for **BillingAddress1**.
BillingZipCode	If AVS ZIP code checking is desired, enter a value for **BillingZipCode**.

The **ReturnURL** must be defined as a page on the merchant web service that will process the response. The response will be as follows.

```
<ReturnURL>?param1=val>&<param2=val>&…<paramX=val>
```

5. Transaction Example

Example Credit Card Sale transaction for $16.00, IPS test credentials, requiring CVV and will use AVS, with an auto-return url defined as Google:

```
https://dev-
paynow.itscocloud.com/execute?isTest=1&ReturnURL=https://www.google.com
&AutoReturn=1&AccountToken=E1EB3EFB049DFB599F1CB454E1CFC4FD14BF90BCE744
56AE9E9490D7D609B466C81A3801&AcceptorID=364798674&AccountID=1188346&Tra
nsactionSetup=1&Amount=16.00&ReferenceNumber=001&TerminalID=002&LaneNum
ber=003&BillingAddress1=100&BillingZipcode= 33606&CVVRequired=1
```

Sample response to above, when using Visa 4445222299990007, exp 12/22, CVV 382:

```
https://www.google.com/?HostedPaymentStatus=Complete&TransactionSetupID
=AAEE184C-6779-4572-
89B7838D20086146&TransactionID=159599897&ExpressResponseCode=0&ExpressR
esponseMessage=Approved&AVSResponseCode=Y&CVVResponseCode=M&ApprovalNum
ber=289337&LastFour=0007&ValidationCode=BD46141F887E4653&CardLogo=Visa&
ApprovedAmount=16.00&BillingAddress1=100&BillingZipcode=33606&Bin=44452
2&Entry=Manual&NetTranID=221860112425222&TranDT=2022-07-05%2011:24:25
```

6. Mobile Payment Implementation

The Hosted Payments Solution requires the integrator to have a Return URL address to receive the results of the payment. In a mobile payment application, the mobile device is not capable of receiving the Return URL information, as this requires a publicly available web server for WorldPay to send the results to.

To facilitate a mobile payments integration, the Hosted Payments Bridge may be used as the Return URL address, and when the Hosted Payments Bridge receives the response from WorldPay, it will retain those results for 5 minutes, allowing the mobile payments app sufficient time to retrieve those results from the Hosted Payments Bridge.

6.1. Operation

- Mobile Application constructs a valid Hosted Payments post string that will include the Hosted Payments Bridge as the return URL, along with a unique identifier the Hosted Payments Bridge can use to reference the transaction.
- Mobile Application posts the string in a Web View. This will post to Hosted Payments Bridge, which will then forward to WorldPay and display the credit card input form.
- Mobile Application needs to start another process that is asking the Hosted Payments Server if the results are ready. This can best be executed at a regular interval (like 1 second), with an appropriate timeout in place.
- Customer enters credit card information. WorldPay sends results to Hosted Payments Bridge. Hosted Payments Bridge stores the response by Mobile Application provided unique ID.
- Mobile Application process requesting results receives the results.
- Mobile Application locally processes results and continues with app payment logic.

6.2. Return URL

The Return URL constructed in the original post string must be as follows:

```
ReturnURL=https://dev-paynow.itscocloud.com/Response?IPS=<UNIQUE_ID>
```

Example:

```
https://dev-paynow.itscocloud.com/Response?IPS=123
```

6.3. HTTP Request for Transaction Results

The HTTP Request must be as follows:

```
https://dev-paynow.itscocloud.com/Response?IPS=<UNIQUE_ID>&Response=Y
```

Example:

```
https://dev-paynow.itscocloud.com/Response?IPS=123&Response=Y
```

This returns a web page with the results enclosed with brackets ({}). If results are not available yet, the page will contain the text string {NO DATA}. Otherwise the page will contain the text string

{<TRANSACTION_RESULTS>}, where **<TRANSACTION_RESULTS>** is the name-value pair string with transaction results.

If the customer cancelled the transaction by clicking the cancel link on the WorldPay payments page, the **<TRANSCTION_RESULTS>** will contain the string **HostedPaymentStatus=Cancelled**.

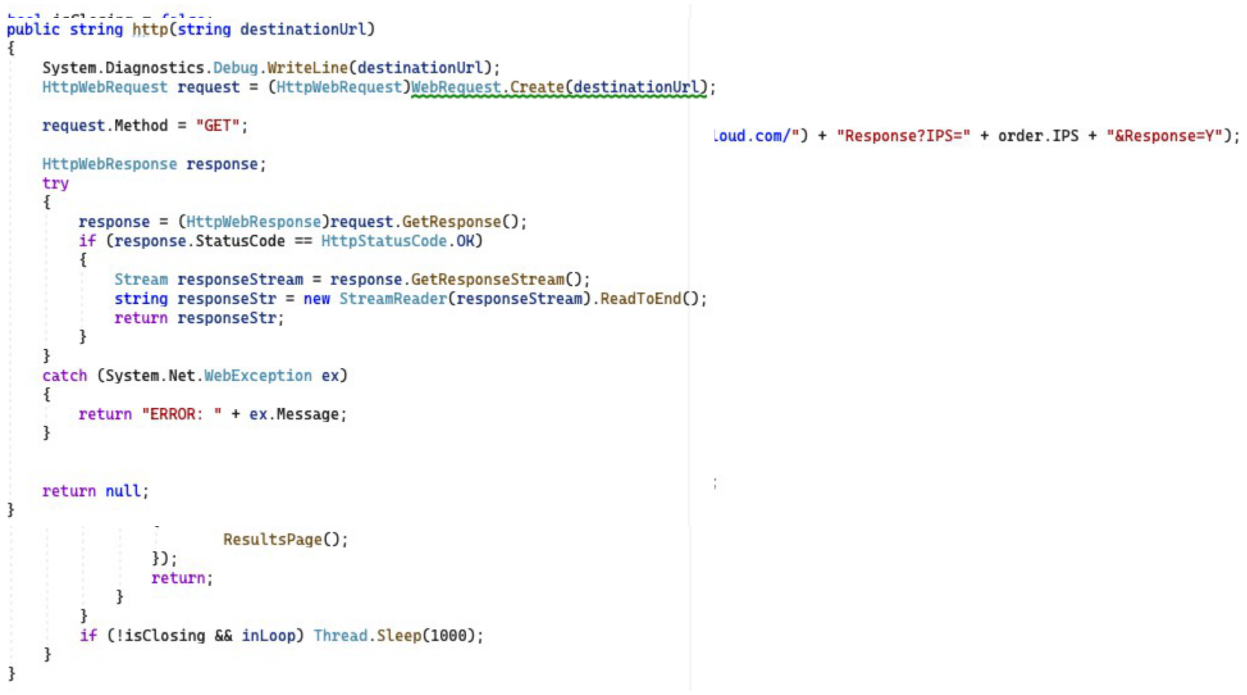
6.4. Example Request

Example Credit Card Sale transaction for $16.00, our test credentials, requiring CVV and will use AVS, with an auto-return url defined as the same hosted payment bridge with ID 123:

```
https://dev-
paynow.itscocloud.com/execute?isTest=1&ReturnURL=https://dev-
paynow.itscocloud.com/Response?IPS=123&AutoReturn=1&License=&AccountTok
en=C8E33719BEC41A4EA07B9E09681ED1323B9074E1651209203BE4C24452E884B4DD84
8601&AcceptorID=364800780&AccountID=1217395&TransactionSetup=1&Amount=1
6.00&ReferenceNumber=001&TerminalID=002&LaneNumber=003&BillingAddress1=
100&BillingZipcode=33606&CVVRequired=1
```

6.5. Reference Polling Code

Below is reference C# code that polls the server in 1 second intervals until results are received. The method waitForResponse is expected to run on a background thread, and then when results are received (either customer cancelled or card results), the code redirects to methods back on the main UI thread. The code looks for "{" and "}" in the response and captures the data between those brackets



```
public string http(string destinationUrl)
{
    System.Diagnostics.Debug.WriteLine(destinationUrl);
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(destinationUrl);

    request.Method = "GET";                                          loud.com/") + "Response?IPS=" + order.IPS + "&Response=Y");

    HttpWebResponse response;
    try
    {
        response = (HttpWebResponse)request.GetResponse();
        if (response.StatusCode == HttpStatusCode.OK)
        {
            Stream responseStream = response.GetResponseStream();
            string responseStr = new StreamReader(responseStream).ReadToEnd();
            return responseStr;
        }
    }
    catch (System.Net.WebException ex)
    {
        return "ERROR: " + ex.Message;
    }


    return null;
}
                        ResultsPage();
                });
                return;
            }
        }
        if (!isClosing && inLoop) Thread.Sleep(1000);
    }
}
```