First Data

Internet Payment Gateway Integration Guide First Data Connect

Version 2.0 (EMEA)

First Data Internet Payment Gateway

INTEGRATION GUIDE
FIRST DATA CONNECT
VERSION 2.0 (EMEA)

Contents

Getting Support

There are different manuals available for the First Data Internet Payment Gateway. This Integration Guide will be the most helpful for integration issues.
For information about settings, customisation, reports and how to process transactions manually (by keying in the information) please refer to the First Data Virtual Terminal User Guide.

If you have read the documentation and cannot find the answer to your question, please contact your local support team.

Information for merchants with existing First Data Connect integration:

This Integration Guide version contains the following important changes in comparison to versions 1.0 to 1.4:

- Parameters for Request- and Response-Hashes (and php/asp utils) have changed
- Name of field ‘trxCcy’ has changed to ‘currency’
- URL suffix has changed from ‘/emea/connectsh’ to ‘connect/gateway/processing’
- When running 3D Secure transactions, the transaction result will be sent to the Success/Fail URL rather than the responseURL (which does not exist as parameter anymore)
- Response parameters of 3D Secure transactions have been aligned with the result of regular (non-3D Secure) transactions

1 Introduction

First Data Connect is a simple payment solution for connecting an online store to the powerful First Data Internet Payment Gateway.
First Data Connect manages all of your interactions with credit card processors and financial institutions.

This document describes how to integrate your website into First Data Connect and provides step by step instructions on how to quickly start accepting payments from your web shop.

2 Payment process options

2.1 Hosted payment page or using your own payment form

First Data Connect basically provides two options for integration with your website:

- With the easiest option you use ready-made form pages for the payment process that we provide and host on our servers. In this case your customer will be forwarded to First Data when it comes to payment and can enter the sensitive cardholder data on our SSL-encrypted page. Afterwards the customer will be redirected to your shop again. Your shop system will be notified about the payment result.

- If you prefer your customer never to leave your website, you can create your own payment forms in your individual corporate design. Although this form will be hosted on your own servers, the sensitive cardholder data can directly be sent from your customer to the gateway so that you do not need to save any credit card data and therefore can avoid security risks. To display a secured website (lock symbol in the browser) to your customer, your website needs to provide a SSL-connection via a HTTPS-Server.

Also, there are three different modes you can choose from to define the range of data that shall be captured by the gateway. Depending on your individual business process, you can choose a mode that only collects payment data or decide to additionally transmit details for the invoice or shipping address.

Depending on the complexity of your business processes, it can also make sense to additionally integrate the First Data API solution (see Integration Guide First Data API).

2.2 PayOnly Mode

In PayOnly mode, the First Data Internet Payment Gateway collects a minimum set of information for the transaction. When using the hosted payment page, one page is presented to the card holder to enter the payment information (e. g. credit card number, expiry data and card code).

2.3 PayPlus Mode

In PayPlus mode, in addition to the above, the gateway also collects a full set of billing information. When using the hosted payment page, the card holder is presented with two pages, one for the billing information and one for the payment information.

2.4 FullPay Mode

If you want First Data Connect to collect all available information (billing, shipping, and payment information), we recommend using FullPay mode. FullPay mode allows you to send the order total to First Data Connect and the system will collect all other required information. This is the easiest way of integrating your web store into First Data Connect. Optionally you can also use this mode with your own forms.

3 Getting Started

This section provides a simple example on how to integrate your website into the First Data Internet Payment Gateway in FullPay Mode. Examples are provided using ASP and PSP. This section assumes that the developer has a basic understanding of his chosen scripting language.

3.1 Checklist

In order to integrate with the gateway, you must have the following items:

- Store Name

  This is the ID of the store that was given to you by First Data.
  For example : 10123456789

- Shared Secret

  This is the shared secret provided to you by First Data.
  This is used when constructing the hash value (see below).

3.2 ASP Example

The following ASP example demonstrates a simple page that will communicate with the First Data Payment Gateway in FullPay mode. When the cardholder clicks ‘Submit’, they are redirected to the First Data Secure Pages, where they can enter their billing, shipping and payment information. After payment has been completed, the user will be redirected to the merchants receipt page. The location of the receipt page can be configured.

```
<!-- #include file="ipg-util.asp"-->

<html>
 <head><title>IPG Connect Sample for ASP</title></head>
 <body>
 <p><h1>Order Form</h1></p>

 <form method="post" action=" https://test.ipg-
online.com/connect/gateway/processing ">

       <input type="hidden" name="txntype" value="sale">
       <input type="hidden" name="timezone" value="CET"/>
       <input type="hidden" name="txndatetime" value="<%
getDateTime() %>"/>
       <input type="hidden" name="hash" value="<% createHash(
13.00,978 ) %>"/>
       <input type="hidden" name="storename" value="10123456789"
/>
       <input type="hidden" name="mode" value="fullpay"/>
       <input type="text" name="chargetotal" value="13.00" />
       <input type="hidden" name="currency" value="978"/>
       <input type="submit" value="Submit">
 </form>
 </body>
</html>
```

The code presented in Appendix I represents the included file ipg-util.asp. It includes code for generating a SHA1 hash as is required by First Data. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

Note, the POST URL used is for integration testing only. When you are ready to go into production, please contact First Data and you will be provided with the live production URL.

Note, the included file, ipg-util.asp uses a server side JavaScript file to build the SHA1 hash. This file can be provided on request. To prevent fraudulent transactions, it is recommended that the 'hash' is calculated within your server and JavaScript is not used like shown in the samples mentioned.

3.3 PHP Example

The following PHP example demonstrates a simple page that will communicate with the First Data Payment Gateway in FullPay mode. When the cardholder clicks ‘Submit’, they are redirected to the First Data Secure Pages, where they can enter their shipping, billing and payment information. After payment has been completed, the user will be redirected to the merchants receipt page. The location of the receipt page can be configured.

```
<? include("ipg-util.php"); ?>

<html>
<head><title>IPG Connect Sample for PHP</title></head>
 <body>
```

```
  <p><h1>Order Form</h1>

<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">
  <input type="hidden" name="txntype" value="sale">
  <input type="hidden" name="timezone" value="CET"/>
  <input type="hidden" name="txndatetime" value="<?php echo
getDateTime() ?>"/>
  <input type="hidden" name="hash" value="<?php echo createHash(
13.00,978 ) ?>"/>
  <input type="hidden" name="storename" value="10123456789"/>
  <input type="hidden" name="mode" value="fullpay"/>
  <input type="text" name="chargetotal" value="13.00"/>
  <input type="hidden" name="currency" value="978"/>
  <input type="submit" value="Submit">
  </form>
  </body>
</html>
```

Note, the POST URL used is for integration testing only. When you are ready to go into production, please contact First Data and you will be provided with the live production URL.

The code presented in Appendix II represents the included file ipg-util.php. It includes code for generating a SHA1 hash as is required by First Data. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

3.4 Amounts for test transactions

When using our test system for integration, odd amounts (e. g. 13.01 EUR or 13.99 EUR) can cause the transaction to decline as these amounts are sometimes used to simulate unsuccessful authorisations.
We therefore recommend to use even amounts for testing purpose, e. g. 13.00 EUR like in the example above.

4 Mandatory Fields

Depending on the transaction type, the following form fields must be present in the form being submitted to the gateway (X = mandatory field).

Field name	Description, possible values and format	„Sale“ transaction	PreAuth (credit card only)	PostAuth (credit card only)	Void
txntype	'sale', 'preauth', 'postauth' or 'void' (the transaction type – please note the descriptions of transaction types in the User Guide Virtual Terminal)	X (sale)	X (preauth)	X (postauth)	X (void)
timezone	GMT, CET or EET (timezone of the transaction)	X	X	X	X
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X
hash	This is a SHA1 hash of the following fields : storename + txndatetime + chargetotal + currency + sharedsecret. Note, that it is important to have the hash generated in this exact order. An example of how to generate a SHA1 hash is given below.	X	X	X	X
storename	This is the ID of the store provided by First Data.	X	X	X	X
mode	‘fullpay’, ‘payonly’ or ‘payplus’ (the chosen mode for the transaction)	X	X		
chargetotal	This is the total amount of the transaction using a dot or comma as decimal separator, e. g. 12.34 for an amount of 12 Euro and 34 Cent. Group separators like (1,000.01 / 1.000,01) are not allowed.	X	X	X	X

currency	The numeric ISO code of the transaction currency, e. g. 978 for Euro (see examples below)	X	X	X	
oid	The order ID of the initial action a PostAuth or Void shall be initiated for			X	X
tdate	Exact identification of a transaction that shall be voided. You receive this value as result parameter ‚tdate' of the corresponding transaction.				X

Currency code list:

Currency name	Currency code	Currency number
Euro	EUR	978
Pound Sterling	GBP	826
US Dollar	USD	840
Swiss Franc	CHF	756
Czech Koruna	CZK	203
Danish Krone	DKK	208
Yen	JPY	392
Rand	ZAR	710
Swedish Krona	SEK	752
Canadian Dollar	CAD	124

5 Optional Form Fields

- paymentMethod

  If you let the customer select the payment method (e. g. Mastercard, Visa, Direct Debit) in your shop environment or want to define the payment type yourself, transmit the parameter paymentMethod along with your Sale or PreAuth transaction. Valid values are:

Payment method	Value
MasterCard	M
Visa	V
American Express	A
Diners	C
JCB	J
Lastschrift (Direct Debit)	debitDE
Maestro	MA
giropay	giropay

If you do not submit this parameter, the gateway will display a drop-down menu to the customer to choose from the payment methods available for your shop.

- customerid

  This field allows you to transmit any value, e. g. your ID for the customer

- invoicenumber

  This field allows you to transmit any value, e. g. an invoice number or class of goods

- refer

  This field describes who referred the customer to your store

- comments

  Place any comments here about the transaction

- responseSuccessURL

  The URL where you wish to direct customers after a successful transaction (your Thank You URL) – only needed if not setup in Virtual Terminal / Customising

- responseFailURL – only needed if not setup in Virtual Terminal / Customising

  The URL where you wish to direct customers after a declined or unsuccessful transaction (your Sorry URL)

- language

  This value can be used to override the default payment page language configured for your merchant store. The following values are currently possible:

Language	language
German	de_DE
English (USA)	en_US
English (UK)	en_EN
Italian	en_IT

6 Using your own forms to capture the data

If you decide to create your own forms, i. e. not to use the ones provided and hosted by First Data, there are additional mandatory fields that you need to include. These fields are listed in the following sections, depending on the mode you choose.

6.1 PayOnly Mode

After your customer has decided how to pay, you present a corresponding HTML-page with a form to enter the payment data as well as hidden parameters with additional transaction information.

In addition to the mandatory fields listed above, your form needs to contain the following fields (part of them can be hidden):

Field name	**Description, possible values and format**	**Credit Card**	**German Direct Debit**	**Maestro**	**giropay**	**UK domestic Maestro/ Solo**
cardnumber	Your customer's card number. 12-24 digits.	X		X		X
expmonth	The expiry month of the card (2 digits)	X		X		X
expyear	The expiry year of the card (4 digits)	X		X		X
cvm	The card code, in most cases on the backside of the card (3 to 4 digits)	X				(X)
accountnumber	Your customer's account number (max. 11 digits)		X		X	
bankcode	Your customer's bank code (8 digits)		X		X	
issuenumber	UK Maestro / Solo card's issue number (1 to 2 digits)					(X) mandatory if cvm not set

6.2 PayPlus Mode

Using PayPlus mode, it is possible to additionally transfer shipping information to the payment gateway. The following table describes the format of these additional fields:

Field Name	Possible Values	Description
bcompany	Alphanumeric characters, spaces, and dashes	Customers Company
bname	Alphanumeric characters, spaces, and dashes	Customers Name
baddr1	Limit of 30 characters, including spaces	Customers Billing Address 1
baddr2	Limit of 30	Customers Billing Address 2

	characters, including spaces	
bcity	Limit of 30 characters, including spaces	Billing City
bstate	2 Letter State Code	US State for shipping to the USA
bstate2	Limit of 30 characters, including spaces	Province or Territory
bcountry	2 Letter Country Code	Country of Billing Address
bzip	International Postal Code	Zip or Postal Code
phone	Limit of 20 Characters	Customers Phone Number
fax	Limit of 20 Characters	Customers Fax Number
email	Limit of 45 Characters	Customers Email Address

6.3 FullPay Mode

Using FullPay mode, it is possible to additionally transfer shipping information to the payment gateway. The shipping information is as specified above. The following table describes the format of the billing fields:

Field Name	Possible Values	Description
sname	Alphanumeric characters, spaces, and dashes	Ship-to Name
saddr1	Limit of 30 characters, including spaces	Shipping Address Line 1
saddr2	Limit of 30 characters, including spaces	Shipping Address Line 2
scity	Limit of 30 characters, including spaces	Shipping City
sstate	2 letter state code	US State for shipping to the USA
sstate2	Limit of 30 characters, including spaces	Province or Territory
scountry	2 letter country code	Country of Shipping Address

szip	International Postal Code	Zip or Postal Code

7 Additional Custom Fields

You may send as many custom fields to the gateway as you wish. Custom field values are returned along with all other fields to the response URL.

It is also possible to document up to fifteen custom fields in your store configuration. You may use these fields to gather additional customer data geared toward your business specialty, or you may use them to gather additional customer demographic data which you can then store in your own database for future analysis.

8 3D Secure

The First Data Internet Payment Gateway includes the ability to authenticate transactions using Verified by Visa and MasterCard SecureCode. If your credit card agreement includes 3D Secure and your Merchant ID has been activated to use this service, you do not need to modify your payment page.

(Modification of Payment Page not required anymore!; deleted section about responseURL as this does not exist anymore)

If you are enabled to submit 3DSecure transactions but for any reason want to submit specific transactions without using the 3D Secure protocol, you can use the additional parameter *authenticateTransaction* and set it to either “true” or “false”.

Example for a transaction without 3D Secure:

```
<input type="hidden" name="authenticateTransaction" value="false"/>
```

In principle, it may occur that 3D Secure authentications can not be processed successfully for technical reasons. If one of the systems involved in the authentication process is temporarily not responding, the payment transaction will be processed as a “regular” eCommerce transaction (GICC ECI 7). **A liability shift to the card issuer for possible chargebacks is not warranted in this case.** If you prefer that such transactions shall not be processed at all, our technical support team can block them for your store on request.

Please note that the technical process of 3D Secure transactions differs in some points compared to a normal transaction flow. If you already have an existing shop integration and plan to activate 3D Secure subsequently, we recommend performing some test transactions on our test environment.

9 Data Vault

With the Data Vault product you can store sensitive cardholder data in an encrypted database in First Data's data centre to use it for subsequent transactions without the need to store this data within your own systems.

If you have ordered this product, the Connect solution offers you the following functions:

- **Store or update payment information when performing a transaction**
  Additionally send the parameter ***hosteddataid*** together with the transaction data as a unique identification for the payment information in this transaction. Depending on the payment type, credit card number and expiry date or account number and bank code will be stored under this ID if the transaction has been successful. In cases where the submitted 'hosteddataid' already exists for your store, the stored payment information will be updated.

- **Initiate payment transactions using stored data**
  If you stored cardholder information using the Data Vault product, you can perform transactions using the 'hosteddataid' without the need to pass the credit card or bank account data again.
  Please note that it is not allowed to store the card code (in most cases on the back of the card) so that for credit card transactions, the cardholder still needs to enter this value. If you use First Data's hosted payment forms, the cardholder will see the last four digits of the stored credit card number, the expiry date and a field to enter the card code.

See further possibilities with the Data Vault product in the Integration Guide for the First Data API.

10 Transaction Response

Upon completion, the transaction details will be sent back to the defined responseSuccessURL or responseFailURL as hidden fields:

Field name	Description
approval_code	Approval code for the transaction
oid	Order ID
refnumber	Reference number
status	Transaction status
txndate_processed	Time of transaction processing
tdate	Identification for the specific transaction, e. g. to be used for a Void
fail_reason	Reason the transaction failed
response_hash	Hash-Value to protect the communication (see note below)
processor_response_code	The response code provided by the backend system

For 3D Secure transactions only:

response_code_3dsecure	Return code indicating the classification of the transaction: **1** – Successful authentication (GICC ECI 11/10) **2** – Successful authentication without AVV (GICC ECI 11/10) **3** – Authentication failed / incorrect password (transaction declined) **4** – Authentication attempt (GICC ECI 13/12) **5** – Unable to authenticate / Directory Server not responding (GICC ECI 7) **6** – Unable to authenticate / Access Control Server not responding (GICC ECI 7) **7** – Cardholder not enrolled for 3D Secure (GICC ECI 13/12) **8** – Merchant not enabled for 3D Secure (transaction declined) Please see note about blocking GICC ECI 7 transactions in the 3D Secure section of this document.

In addition, your custom fields and billing/shipping fields will also be sent back to the specific URL.

The parameter *response_hash* allows you to recheck if the received transaction response has really been sent by First Data and can therefore protect you from fraudulent manipulations.
The value is created with a SHA 1 Hash using the following parameter string:
`sharedsecret + approvalcode + chargetotal + currency + txndatetime + storename`

Appendix I

ipg-util.asp

```
<Script LANGUAGE=JScript RUNAT=Server src="sha1.js">
</SCRIPT>
<Script LANGUAGE=JScript RUNAT=Server>
      var today = new Date();
      var formattedDate = today.formatDate("Y:m:d-H:i:s");

      /*
            Function that calculates the hash of the following
            parameters:
             - Store Id
             - Date/Time(see $dateTime above)
             - chargetotal
             - shared secret
             - currency (numeric ISO value)
      */
      function createHash(chargetotal, currency) {
            // Please change the store Id to your individual Store ID
            var storeId = "10123456789;
            // NOTE: Please DO NOT hardcode the secret in that
script. For example read it from a database.
            var sharedSecret = "sharedsecret";

            var stringToHash = storeId + formattedDate + currency +
chargetotal + sharedSecret;

            var ascii = getHexFromChars(stringToHash);

            var hash = calcSHA1(ascii);

            Response.Write(hash);
      }
      function getHexFromChars(value) {
            var char_str = value;
            var hex_str = "";
            var i, n;
            for(i=0; i < char_str.length; i++) {
                  n = charToByte(char_str.charAt(i));
                  if(n != 0) {
                        hex_str += byteToHex(n);
                  }
            }
            return hex_str.toLowerCase();
      }

      function getDateTime() {
            Response.Write(formattedDate);
      }
</SCRIPT>
```

Appendix II

ipg-util.php

```
<?php
      $dateTime = date("Y:m:d-H:i:s");

      function getDateTime() {
            global $dateTime;
            return $dateTime;
      }

      function createHash($chargetotal, $currency) {
            $storeId = "10123456789";
$sharedSecret = "sharedsecret";

            $stringToHash = $storeId . getDateTime() . $chargetotal .
$currency . $sharedSecret;

            $ascii = bin2hex($stringToHash);

            return sha1($ascii);
      }

?
```