Connect

Integration Guide

Version 2023-4 (IPG)

Integration Guide Connect

Version 2023-4 (IPG)

Contents

Getting Support

There are different manuals available for Fiserv's eCommerce solutions. This Integration Guide will be the most helpful for integrating hosted payment forms or a Direct Post.

For information about settings, customization, reports and how to process transactions manually (by keying in the information) please refer to the User Guide Virtual Terminal.

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

1. Introduction

The Connect solution provides a quick and easy way to add payment capabilities to your website.

Connect manages the customer redirections that are required in the checkout process of many payment methods or authentication mechanisms and gives you the option to use secure hosted payment pages which can reduce the burden of compliance with the Data Security Standard of the Payment Card Industry (PCI DSS).

This document describes how to integrate your website using Connect and provides step by step instructions on how to quickly start accepting payments from your webshop.

When making decisions on your way of integration, please consider that we do not recommend to use the hosted payment forms inside an iFrame since some Internet browsers do not allow cookies to be sent to the 3rd party hosts, moreover some features (e.g.: 3-D Secure authentications) and some Alternative Payment methods that involve redirections to the 3rd party services (e.g.: iDEAL or PayPal) do not allow displaying their screens within an iFrame. However, if you still plan to embed our hosted payment pages inside an iFrame you must use the 'parentUri' parameter to specify an URL of a page, where the hosted payment page will be embedded.

Depending on your business processes, it can also make sense to additionally integrate our Web Service API solution (see Web Service API Integration Guide).

2. Payment process options

The Connect solution provides several different options for the payment process to support integrations where you handle most of the customer interactions on your own website up to integrations where you use ready-made form pages for the entire payment process.

Hosted Payment Page

If you want to fully outsource the payment process in order not to have any sensitive cardholder data on your systems, you can use our ready-made hosted pages for your customers to enter their payment information.

The most important aspect around the usage of hosted payment page is the security of sensitive cardholder data. When you decide to let your customers enter their credit card details on the page that we provide and host on our servers for this purpose, it facilitates your compliance with the Data Security Standard of the Payment Card Industry (PCI DSS) as the payment processing is completely hosted by Fiserv.

For a standard hosted payment page integration, you should use the checkout option ‘combinedpage’ that consolidates the payment method choice and the typical next step (e.g.: entry of card details or selection of bank) in a single page, which gets automatically optimized for different kinds of user devices (e.g.: PC, smartphone, tablet, etc.).

The hosted page is localized in many languages and can be easily customized with your merchant’s logo, colors, and font types to make it fit to the look and feel of your shop environment (refer to the User Guide Virtual Terminal to learn more). It also shows your merchant’s name (i.e.: legal name) and allows you to display a summary of the purchased items to your customer in the ‘Your Order’ box.

If you do not want to let your customer select the payment method on our hosted page but want to handle that part upfront within your shop environment, you should submit a value for the parameter '[paymentMethod](#)’ in your request to the gateway. In addition, if you do not want to distinguish between different card brands (but just card vs. alternative payment methods), you can send a valid card brand value for the parameter 'paymentMethod’ and your customer will see a hosted page for the card details entry with no card brand logo shown. Please contact your local support team if you want to enable this

feature. This will be managed with a specific setting performed on your account (service configuration) ‘hideCardBrandLogoInCombinedPage’).

If you do not submit a value for the parameter 'paymentMethod’, the gateway will take your customer to a hosted page to choose from the payment methods activated for your store.

If you do not include in your request the fields like e.g.: the card number or the expiry date for a card payment, the gateway will take your customer to a hosted page to collect this information as being mandatory for a transaction processing.

When e.g.: you plan to integrate a specific local alternative payment method i.e.: Local Wallets India, PayLater by ICICI Bank and RuPay, or you require the gateway to collect a full set of billing and/or shipping information, or your consumers use an old operating system with outdated browser versions, please contact your local support team to discuss an alternative hosted payment page integration while using the legacy checkout option ‘classic’.

Direct Post

In the scenarios where you prefer not to use a hosted payment page, you can submit the required customer data directly from your own form to Fiserv, but please be aware that if you store or process sensitive cardholder data within your own application, you must ensure that your system components are compliant with the Data Security Standard of the Payment Card Industry (PCI DSS).

You create the payment form and display it within your website or app. When your customer has entered the card details and presses the "continue button", the customer's device sends the payment information directly to the gateway.

If you choose the Direct Post option and create your own forms, there are additional fields that must be included in your transaction request to the gateway, which are listed in the chapter on using your own forms to capture the data.

3. Getting Started

This section provides a simple example on how to integrate your website using the “combinedpage” checkout option. Examples are provided using ASP and PHP. This section assumes that the developer has a basic understanding of his chosen scripting language.

Checklist

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

- Store Name

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

- Shared Secret

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

ASP Example

The following ASP example demonstrates a simple page that will communicate with the payment gateway.

When the cardholder clicks *Submit*, they are redirected to the Fiserv secure page to enter the card details. After payment has been completed, the user will be redirected to the merchant's receipt page. The location of the receipt page can be configured.

```
<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="Europe/Berlin"/>
    <input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
    <input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
    <input type="hidden" name="hashExtended" value="<% call
    createExtendedHash("13.00","978") %>"/>
    <input type="hidden" name="storename" value="10123456789" />
    <input type="hidden" name="checkoutoption" value="combinedpage"/>
    <input type="hidden" name="paymentMethod" value="M"/>
    <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 II represents the included file ipg-util.asp. It includes code for generating a hash as is required by Fiserv. 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 Fiserv 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 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.

PHP Example

The following PHP example demonstrates a simple page that will communicate with the payment gateway.

When the cardholder clicks *Submit*, they are redirected to the Fiserv secure page to enter the card details. After payment has been completed, the user will be redirected to the merchant's receipt page. The location of the receipt page can be configured.

```
<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="Europe/Berlin"/>
    <input type="hidden" name="txndatetime" value="<?php echo getDateTime() ?>"/>
    <input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
    <input type="hidden" name="hashExtended" value="<?php echo
    createExtendedHash("13.00","978") ?>"/>
    <input type="hidden" name="storename" value="10123456789"/>
    <input type="hidden" name="checkoutoption" value="combinedpage"/>
    <input type="hidden" name="paymentMethod" value="M"/>
    <input type="text" name="chargetotal" value="13.00"/>
    <input type="hidden" name="currency" value="978"/>
```

```
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

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

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

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 authorizations.

We therefore recommend using 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 payment gateway (X = mandatory field). Please refer to this Integration Guide's Appendixes for implementation details in relation to alternative payment methods and the other product options.

Field Name	Description, possible values and format	Sale transaction	PreAuth*	PostAuth*	Void	PayerAuth**
txntype	'sale', 'preauth', 'postauth', 'void' or 'payer_auth' (the transaction type – please note the descriptions of transaction types in the User Guide Virtual Terminal) The possibility to send a 'void' using the Connect interface is restricted. Please contact your local support team if you want to enable this feature.	X (sale)	X (preauth)	X (postauth)	X (void)	X (payer_auth)
timezone	Time zone of the transaction in Area/Location format, e.g. Africa/Johannesburg America/New_York America/Sao_Paulo Asia/Calcutta Australia/Sydney Europe/Amsterdam Europe/Berlin Europe/Dublin Europe/London Europe/Rome	X	X	X	X	X
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X	X
hash_algorithm	This is to indicate the algorithm that you use for hash calculation. The possible values are: • HMACSHA256	X	X	X	X	X

	• HMACSHA384 • HMACSHA512 Only one algorithm value should be used.					
hashExtended	The extended hash needs to be calculated using all non-empty gateway specified request parameters in ascending order of the parameter names, where the upper-case characters come before the lower case (based on ASCII value) and the shared secret must be used as the secret key for calculating the hash value. When you are using Direct Post, there is also an option where you do not need to know the card details (PAN, CVV and Expiry Date) for the hash calculation. This will be managed with a specific setting performed on your store. Please contact your local support team if you want to enable this feature. An example of how to generate a hash is given in Appendix I.	X	X	X	X	X
storename	This is the ID of the store provided by Fiserv.	X	X	X	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 like1,000.01 / 1.000,01 are not allowed.	X	X	X	X	X
checkoutoption	Set the value for this parameter to 'combinedpage' for a standard hosted payment page integration.	X	X			X
currency	The numeric ISO code of the transaction currency, e. g. 978 for Euro (see examples in Appendix IV)	X	X	X	X	X
oid	The order ID of the initial action a PostAuth shall be initiated for.			X	X	
ipgTransactionId *or* merchantTransactionId	Exact identification of a transaction that shall be voided. You receive this value as result parameter, 'ipgTransactionId' of the corresponding transaction. Alternatively, 'merchantTransactionId' can be used for the Void in case the merchant has assigned one.				X	

* The transaction types 'preauth' and 'postauth' only apply to the payment methods credit card, PayPal.
** The transaction type 'payer_auth' is only required if you want to split the 3-D Secure authentication process from the payment transaction (authorization) process. See more information in the 3-D Secure section of this guide.

Please see a list of currencies and their ISO codes in Appendix IV.

5. Optional Form Fields

Field Name	Description, possible values and format
cardFunction	This field allows you to indicate the card function in case of combo cards which provide credit and debit functionality on the same card. It can be set to 'credit' or 'debit'.

<table>
<tr><td></td><td>The field can also be used to validate the card type in a way that transactions where the submitted card function does not match the card's capabilities will be declined. If you e.g.: submit "cardFunction=debit" and the card is a credit card, the transaction will be declined.</td></tr>
<tr><td>comments</td><td>Place any comments here about the transaction.</td></tr>
<tr><td>customerid</td><td>This field allows you to transmit any value, e. g. your ID for the customer.
Please note that for:
Direct Debit transactions, the Customer ID can be submitted to the bank with the maximum length of 32 characters. The minimum length of the Order ID is 32 characters, but it can be longer if the Customer ID is shorter. The maximum number of characters for both Customer ID and Order ID that can be submitted to the bank is 64. Please contact your local support team if you want to enable this feature. Note that this is not applicable when processing Direct Debit through the Fiserv Local Payments offering.
iDEAL transactions, the Customer ID can be submitted in your request filled in with any relevant data which can be populated in a field in the iDEAL TransactionRequest to be displayed on your consumers' bank account statements. Please note that this is not applicable when processing iDEAL through the Fiserv Local Payments offering.</td></tr>
<tr><td>dccInquiryId</td><td>Inquiry ID for a Dynamic Pricing request. Used to send the Inquiry ID you have obtained via a Web Service API call ('RequestMerchantRateForDynamicPricing'). This value will be used to retrieve the currency conversion information (exchange rate, converted amount) for this transaction.</td></tr>
<tr><td>dccSkipOffer</td><td>If the cardholder declines the currency conversion offer within your environment, the request parameter 'dccSkipOffer' can be set to 'true' so that the hosted consumer dialogue will automatically be skipped.</td></tr>
<tr><td>dynamicMerchantName</td><td>The name of the merchant to be displayed on the cardholder's statement. The length of this field should not exceed 25 characters. If you want to use this field, please contact your local support team to verify if this feature is supported in your country.</td></tr>
<tr><td>hideOrderDetails</td><td>Set this parameter to 'true' when you want to hide (remove) the 'Your Order' box from our hosted payment page.</td></tr>
<tr><td>highRiskPurchaseIndicator</td><td>This optional parameter needs to be set to 'true', for transactions handling a cryptocurrency and initiated from a MCC 6051 (Quasi Cash—Merchant) store; or for transactions handling high risk securities initiated from the store with MCC 6211 (Securities—Brokers/ Dealers).</td></tr>
<tr><td>idealIssuerID</td><td>This parameter can be used to submit the iDEAL issuing bank in case you let your customers select the issuer within your shop environment. If you do not pass this value for an iDEAL transaction, a hosted selection form will be displayed to your customer. Please note that this is not applicable when processing iDEAL through the Fiserv Local Payments offering.
<table>
<tr><th>iDEAL issuer</th><th>Value</th></tr>
<tr><td>ABN AMRO</td><td>ABNANL2A</td></tr>
<tr><td>ING</td><td>INGBNL2A</td></tr>
<tr><td>SNS Bank</td><td>SNSBNL2A</td></tr>
<tr><td>Van Lanschot Kempen</td><td>FVLBNL22</td></tr>
<tr><td>Triodos Bank</td><td>TRIONL2U</td></tr>
<tr><td>Knab</td><td>KNABNL2H</td></tr>
<tr><td>Rabobank</td><td>RABONL2U</td></tr>
<tr><td>RegioBank</td><td>RBRBNL21</td></tr>
<tr><td>ASN Bank</td><td>ASNBNL21</td></tr>
<tr><td>Bunq</td><td>BUNQNL2A</td></tr>
<tr><td>Moneyou</td><td>MOYONL21</td></tr>
<tr><td>Revolut</td><td>REVOLT21</td></tr>
<tr><td>Nationale Nederlanden</td><td>NNBANL2G</td></tr>
</table></td></tr>
<tr><td>invoicenumber</td><td>This field allows you to transmit any value, e. g. an invoice number or class of goods. Please note that the maximum length for this parameter is 48 characters.</td></tr>
<tr><td>item1 up to item999</td><td>Line items are regular Connect integration key-value parameters (URL-encoded), where:
the name is a combination of the keyword item and a number, where the number indicates the list position e.g.: item1</td></tr>
</table>

<table>
<tr><td></td><td>

- the value is represented by a semicolon-separated list of values, where the position indicates the meaning of the list item property e.g.: <1>;<2>;<3>;<4>;<5>;<6>;<7>

The ‘item1’ to ‘item999’ parameters allow you to send basket information in the following format:

id;description;quantity;item_total_price;sub_total;vat_tax;shipping;local_tax;category;detailed_category
</td></tr>
<tr><td>language</td><td>This parameter can be used to override the default payment page language configured for your merchant store.
The following values are currently possible:
<table>
<tr><th>Language</th><th>Value</th></tr>
<tr><td>Chinese (simplified)</td><td>zh_CN</td></tr>
<tr><td>Chinese (traditional)</td><td>zh_TW</td></tr>
<tr><td>Czech</td><td>cs_CZ</td></tr>
<tr><td>Danish</td><td>da_DK</td></tr>
<tr><td>Dutch</td><td>nl_NL</td></tr>
<tr><td>English (USA)</td><td>en_US</td></tr>
<tr><td>English (UK)</td><td>en_GB</td></tr>
<tr><td>Finnish</td><td>fi_FI</td></tr>
<tr><td>French</td><td>fr_FR</td></tr>
<tr><td>German</td><td>de_DE</td></tr>
<tr><td>Greek</td><td>el_GR</td></tr>
<tr><td>Hungarian</td><td>hu_HU</td></tr>
<tr><td>Italian</td><td>it_IT</td></tr>
<tr><td>Japanese</td><td>ja_JP</td></tr>
<tr><td>Norwegian (Bokmål)</td><td>nb_NO</td></tr>
<tr><td>Polish</td><td>pl_PL</td></tr>
<tr><td>Portuguese (Brazil)</td><td>pt_BR</td></tr>
<tr><td>Serbian (Serbia)</td><td>sr_RS</td></tr>
<tr><td>Slovak</td><td>sk_SK</td></tr>
<tr><td>Slovenian</td><td>sl_SI</td></tr>
<tr><td>Spanish (Spain)</td><td>es_ES</td></tr>
<tr><td>Spanish (Mexico)</td><td>es_MX</td></tr>
<tr><td>Swedish</td><td>sv_SE</td></tr>
</table></td></tr>
<tr><td>mandateDate</td><td>This field allows you to reference to the date of the original mandate when performing recurring Direct Debit transactions. The date needs to be submitted in format YYYYMMDD.
Please note that this is a mandatory field for recurring Direct Debit transactions.</td></tr>
<tr><td>mandateReference</td><td>This field allows you to transmit a Mandate Reference for Direct Debit payments. Please note the regulatory requisite to keep the Mandate Reference unambiguous.</td></tr>
<tr><td>mandateType</td><td>This field allows you to process Direct Debit transactions that are based on mandates for recurring collections. The mandate type can be set to ‘single’ for single (one-off) debit collections, to ‘firstCollection’ when submitting the initial transaction related to a mandate for recurring Direct Debit collections, to ‘recurringCollection’ for subsequent recurring transactions or to ‘finalCollection’ for the last direct debit in a series of recurring direct debits. Transactions where this parameter is not submitted by the merchant will be flagged as a single debit collection.
Please note that it is mandatory to submit a mandateReference in case of recurring collections.</td></tr>
<tr><td>mandateUrl</td><td>When your store is enabled for SEPA Direct Debit as part of the Local Payments offering, this field allows you to transmit a valid URL of SEPA Direct Debit mandate to enable the Risk and Compliance department to access the details.
Please note that it is mandatory to submit a mandateReference and a mandateDate together with a mandateUrl in case you manage SEPA Direct Debit mandates on your side in the combination with the Local Payments offering.</td></tr>
<tr><td>marketplaceForeignRetailerIndicator</td><td>This field indicates if a marketplace retailer performing domestic transaction is in a different country.
Available values:</td></tr>
</table>

	"F" – marketplace retailer is foreign "blank" – marketplace retailer is domestic The parameter must be sent if a transaction is submitted by a marketplace retailer in APAC region.
merchantTransactionId	Allows you to assign a unique ID for the transaction. This ID can be used to reference to this transaction in a PostAuth or Void request ('referencedMerchantTransactionId').
mobileMode	The legacy checkout option specific parameter: If your customer uses a mobile device for shopping at your online store you can submit this parameter with the value 'true', when using the 'classic' checkout option. This will lead your customer to a payment page flow that has been specifically designed for mobile devices.
mode	The legacy checkout option specific parameter: If you are building a payment request for the Sale, PreAuth or PayerAuth transaction, when using the 'classic' checkout option, your request needs to include a value for one of the three different modes to define the range of data that shall be captured by the gateway: • 'payonly' - shows a hosted page to collect the minimum set of information for the transaction (e. g. cardholder name, card number, expiry date and card code for a credit card transaction), • 'payplus' - in addition to the above, the payment gateway collects a full set of billing information on an additional page, • 'fullpay' - in addition to the above, the payment gateway displays a third page to also collect shipping information.
numberOfInstallments	This parameter allows you to set the number of instalments for a Sale transaction if your customer pays the amount in several parts.
installmentsInterest	This parameter allows you to choose, if instalment interest should be applied or not, the values "true" or "false" are currently possible.
installmentDelayMonths	This parameter allows you to delay the first instalment payment for several months, values 2-99 are currently possible.
oid	This field allows you to assign a unique ID for your order. If you choose not to assign an order ID, the Fiserv system will automatically generate one for you. Please note, that only the following characters are allowed, if you are generating oid yourselves: A-Z, a-z, 0-9, "-" Please note that for Direct Debit transactions, a maximum of 78 characters can be submitted to the bank.
parentUri	If you plan to embed our hosted payment pages inside an iFrame you must use this parameter, with the maximum length of 2048 characters, to specify an URL of a page, where the hosted payment page will be embedded. However, note that we do not recommend using the hosted payment forms inside an iFrame since some Internet browsers do not allow cookies to be sent to the 3rd party hosts, moreover some features (e.g.: 3-D Secure authentications) and some Alternative Payment methods that involve redirections to the 3rd party services (e.g.: iDEAL or PayPal) do not allow displaying their screens within an iFrame.
partialApproval	The partial approval feature is particularly useful when the transaction amount exceeds the available funds on the customers card. This feature will allow an approval of the available amount to pay for a portion of the transaction, then the remainder can be paid using another payment method. If you are eligible to use this feature, then you can use this parameter to indicate whether to allow partial approval or not. Valid values: • true • false (default)
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. If you do not submit this parameter, the payment gateway will display a drop-down menu to the customer to choose from the payment methods available for your shop. For valid payment method values please refer to Appendix V.
ponumber	This field allows you to submit a Purchase Order Number with up to 50 characters.
refer	This field describes who referred the customer to your store.
referencedMerchantTransactionID	This field allows to reference to a merchantTransactionId of a

	transaction when performing a Void. This can be used as an alternative to ipgTransactionId if you assigned a merchantTransactionId in the original transaction request.
referencedSchemeTransactionId	Credentials on file (COF) specific parameter. This field allows you to include in your request 'schemeTransactionId' that has been returned in the response of the initial transaction to provide a reference to the original transaction, which stored the credentials for the first time.
responseFailURL	The URL where you wish to direct customers after a declined or unsuccessful transaction (your Sorry URL) – only needed if not setup in Virtual Terminal / Customisation.
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 / Customisation.
shipping	This parameter can be used to submit the shipping fee, in the same format as 'chargetotal'. If you submit 'shipping', the parameters 'subtotal' and 'vattax' have to be submitted as well. Note that the 'chargetotal' has to be equal to 'subtotal' plus 'shipping' plus 'vattax'.
trxOrigin	This parameter allows you to use the secure and hosted payment form capabilities within your own application. Possible values are: • 'MAIL' (for transactions where the payment details are captured manually and provided in written form the Card Code entry is not allowed), • 'PHONE' (for transactions where you have received the order over the phone and enter the payment details yourself the Card Code entry is required), • 'ECI' (for standard usage in an eCommerce environment where your customer enters the payment details).
unscheduledCredentialOnFileType	Credentials on file (COF) specific parameter. This field allows you to flag transactions as unscheduled credential on file type. Currently the valid values are: FIRST, CARDHOLDER_INITIATED or MERCHANT_INITIATED to advise the scenario if the credential is stored on your side.
vattax	This field allows you to submit an amount for Value Added Tax or other taxes, e.g.: GST in Australia. Please ensure the sub total amount plus shipping plus tax equals the charge total.

6. Using your own forms to capture the data

If you decide to create your own forms, i.e.: Direct Post (not to use the ones provided and hosted by Fiserv), there are additional mandatory fields that you need to include. These fields are listed in the following sections.

Using Direct Post allows you to have full control over the look and feel of the form where your customers enter their card details for payment while simultaneously avoiding the need to have sensitive card data within your systems.

It is also important that you check if JavaScript is activated in your customer's browser. If necessary, inform your customer that JavaScript needs to be activated for the payment process.

Capture payment details

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, your form needs to contain the following fields (part of them can be hidden).

For Credit/Debit Card and SEPA Direct Debit fields

Field Name	**Description, possible values and format**	**Credit Card** (+ Visa Debit/Electron/Delta)	**SEPA Direct Debit**	**Maestro**	**Bancontact**	**UnionPay SecurePlus**
cardnumber	Your customer's card number. 12-24 digits.	X		X	X	X
expmonth	The expiry month of the card (2 digits)	X		X	X	(X) mandatory if credit card
expyear	The expiry year of the card (4 digits)	X		X	X	(X) mandatory if credit card
cvm	The card code, in most cases on the backside of the card (3 to 4 digits)	X		X as an optional field "if on card"		(X) mandatory if credit card
iban	Your customer's IBAN - International Bank Account Number (up to 34 digits)		X			
bname	Name of the bank account owner that will be debited (alphanumeric characters, spaces, and dashes limited to 96)		X			
baddr1	Street name and house number of the bank account owner that will be debited (alphanumeric characters, spaces, and dashes limited to 96 characters)		(X) mandatory if IBAN belongs to EFTA and associated country or you have signed a GLV* contract with your service provider			
bcity	City of the bank account owner that will be debited (alphanumeric characters, spaces, and dashes limited to 96 characters)		(X) mandatory if IBAN belongs to EFTA and associated country or you have signed a GLV* contract			
bcountry	Country of the bank account owner that will be debited (2 letter country code)		(X) mandatory if IBAN belongs to EFTA and associated country or you have signed a GLV* contract			
bzip	Zip or postal code of the bank account owner that will be debited (limit of 24 characters incl. spaces)		(X) mandatory if IBAN belongs to EFTA and associated country or you have signed a GLV* contract			

(*) Garantierte Lastschriftverfahren (GLV) is part of the TeleCash from Fiserv offering.

For the Local Payments method specific (mandatory/optional) fields please refer to Appendix X.
For the China Domestic method specific (mandatory/optional) fields please refer to Appendix XII.
For the Korea Domestic method specific (mandatory/optional) fields please refer to Appendix XIV.

Capture billing information

It is possible to additionally transfer billing 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 limited to 96	Customers Company
bname	Alphanumeric characters, spaces, and dashes limited to 96	Customers Name
baddr1	Limit of 96 characters, including spaces	Customers Billing Address 1
baddr2	Limit of 96 characters, including spaces	Customers Billing Address 2
bcity	Limit of 96 characters, including spaces	Billing City
bstate	Limit of 96 characters, including spaces	State, Province or Territory
bcountry	2 Letter Country Code	Country of Billing Address
bzip	Limit of 24 characters, including spaces	Zip or Postal Code
phone	Limit of 32 Characters	Customers Phone Number
fax	Limit of 32 Characters	Customers Fax Number
email	Limit of 254 Characters	Customers Email Address

Capture shipping information

It is possible to additionally transfer shipping information to the payment gateway. The billing information is as specified above. The following table describes the format of the shipping fields:

Field Name	Possible Values	Description
sname	Alphanumeric characters, spaces, and dashes limited to 96	Ship-to Name
saddr1	Limit of 96 characters, including spaces	Shipping Address Line 1
saddr2	Limit of 96 characters, including spaces	Shipping Address Line 2

scity	Limit of 96 characters, including spaces	Shipping City
sstate	Limit of 96 characters, including spaces	State, Province or Territory
scountry	2 letter country code	Country of Shipping Address
szip	Limit of 24 characters, including spaces	Zip or Postal Code
sphnumber	Limit of 32 Characters	Customers' Phone Number
semail	Limit of 254 Characters	Customers Email Address

Validity checks

Prior to the authorization request for a transaction, the payment gateway performs the payment methods' specific validation checks.

For Credit/Debit Card or SEPA Direct Debit transactions the following checks are performed:

- The expiry date of cards needs to be in the future.
- The Card Security Code field must contain 3 or 4 digits.
- The structure of a card number must be correct (LUHN check).
- The name of the account holder for SEPA Direct Debit transactions must be submitted.
- An IBAN for SEPA Direct Debit transactions must be correct and contain up to 34 digits.
- If an IBAN belongs to one of the following countries: Andorra, Switzerland, United Kingdom (incl. Jersey, Guernsey, Isle of Man), Gibraltar, Iceland, Liechtenstein, Monaco, Norway, San Marino, Vatican City, or you have signed a Garantierte Lastschriftverfahren (GLV) contract with your service provider, then the account holder's address details (i.e.: street name and house number, zip code, city, and country) must be submitted.

If the submitted data should not be valid, the payment gateway presents a corresponding data entry page to the customer.

To avoid this hosted page when using your own input forms for the payment process, you can transmit the following additional parameter along with the transaction data:

```
full_bypass=true
```

In that case you get the result of the validity check back in the transaction response and can display your own error page based on this.

Please note, if the transaction is eligible for DCC (your store is configured for DCC and the customer is paying by credit card capable of DCC), your customer will be presented the DCC page despite having full_bypass set to true. This is due to regulatory reasons. You can avoid displaying of DCC choice pages by doing the DCC Inquiry yourself via our Web Service API (RequestMerchantRateForDynamicPricing).

7. Additional Custom Fields

You may want to use further fields to gather additional customer data geared toward your business specialty, or to gather additional customer demographic data which you can then store in your own database for future analysis. You can send as many custom fields to the payment gateway as you wish, and they will get returned along with all other fields to the response URL.

Up to ten custom fields can be submitted in a way that they will be stored within the gateway so that they appear in the Virtual Terminal's Order Detail View as well as in the response to Inquiry Actions that you send through our Web Service API.

Field Name	Description, possible values and format
customParam_key	If you want to use this feature, please send the custom fields in the format customParam_key=value. The maximum length of a custom parameter is 100 characters. Example:`<input type="hidden" name="customParam_color" value="green"/>`

To remain compliant the custom fields are not to be used to submit credit card detail or sensitive card holder information, please use the designated fields defined by the Gateway for this purpose.

8. 3-D Secure

The Connect solution includes the ability to authenticate transactions using Verified by Visa, MasterCard Identity Check, American Express SafeKey, JCB J/Secure and Diners ProtectBuy to provide an additional security layer for online card transactions.

If your store is enabled for 3-D Secure, all Sale or preAuth transactions that you initiate by posting an HTML form will by default go through the 3-D Secure process without the need for you to do anything, i.e.: cardholders with an enrolled card will see a page from the card issuer to perform 2-factor authentication unless the card issuer decides not to check it.

The generic fields to be considered:

Field Name	Description, possible values and format

authenticateTransaction	Optional parameter to be set either to 'true' or 'false' to enable or disable 3-D Secure authentication on a Transaction-by-Transaction basis.
	
	Example for a transaction with 3-D Secure:
	`<input type="hidden" name="authenticateTransaction" value="true"/>`
	
	Example for a transaction without 3-D Secure:
	`<input type="hidden" name="authenticateTransaction" value="false"/>`

threeDSRequestorChallengeIndicator	Optional parameter for EMV 3-D Secure (2.0) to be set toone of the values from the list below in order to indicate the preferred type of authentication. If no specific value is present in the transaction request, default value "01" is used. • 01 - no preference (set as default value) • 02 - no challenge requested • 03 - challenge requested 3DS requestor preference • 04 - challenge requested mandate • 05 - No challenge requested (Transaction Risk Analysis is already performed) • 06 - No challenge requested (Data Share Only) • 07 - No challenge requested (SCA is already performed) • 08 - No challenge requested (Utilize whitelist exemption if no challenge required)09 - Challenge requested (Whitelist prompt requested if challenge required)

threeDSTransType	The parameter for EMV 3-D Secure (2.0) represents the type of purchased item, mandatory for Visa and Brazilian market, otherwise optional. If no specific value is present in the transaction request, default value is used. • 01 - Goods/ Service Purchase (default value) • 03 - Check Acceptance • 10 - Account Funding • 11 - Quasi-Cash Transaction • 28 - Prepaid Activation and Load
scaExemptionIndicator1	Optional parameter to request an exemption from Strong Customer Authentication (SCA) without the need to perform 3-D Secure authentication. Currently available values: • Low Value Exemption • TRA Exemption • Trusted Merchant Exemption • SCP Exemption • Delegated Authentication • Authentication Outage Exception Note this parameter is relevant only for the European merchants impacted by the PSD2 requirements.
skipTRA	This optional parameter allows you to use 3-D Secure even if the transaction has been evaluated as low risk and would be eligible for an exemption. Currently available values: • true • false When your store has been set up with Transaction Risk Analysis (TRA) service, but you do want to force 3-D Secure authentication for a certain transaction, set ‘skipTRA’ to ‘true’. Note this parameter is relevant only for the European merchants impacted by the PSD2 requirements.
oid	Use this optional parameter to assign an identifier for your order; in case you plan to authenticate the transaction using EMV 3DS protocol (aka 3DS 2.1 or 2.2) only the following characters are allowed: • A-Z, a-z, 0-9, "-"
deviceChannel	Use this optional parameter to request a 3DS Requestor Initiated flow to be performed. If no value is submitted in the request a default value “02” is automatically submitted by the Gateway. Currently available values: • 02 – Browser Flow • 03 – 3RI Flow
threeRIInd	In cases you require a 3RI flow to be performed, you must indicate the character of your request. Currently available values: • 01 – Recurring transaction • 02 – Instalment transaction • 03 – Add card • 04 – Maintain card information • 05 – Account verification

	• 06 – Split / delayed shipment • 07 – Top-up • 08 – Mail Order • 09 – Telephone Order • 10 – Whitelist status check • 11 – Other payment
recurringExpiry	This field represents a date after which no further recurring transactions are performed. The field needs to be submitted in cases, where the first recurring or installment transaction is to be performed through 3-D Secure. If no specific value is present in the transaction request, the Gateway calculates the value based on populated recurring parameters.
recurringFrequency	Indicates the minimum number of days between authorisations for a recurring or instalment transaction and should include a numeric value between 1 and 9999. If no specific value is present in the transaction request, the Gateway calculates the value based on populated recurring parameters.

Due to newest data integrity scheme requirements, it is expected to provide also following information in the transaction request, if available.

Field Name	Field Type	Description
bname	Alphanumeric characters, spaces, and dashes limited to 96	Customers Name
baddr1	Limit of 96 characters, including spaces	Customers Billing Address 1
baddr2	Limit of 96 characters, including spaces	Customers Billing Address 2
bcity	Limit of 96 characters, including spaces	Billing City
bstate	Limit of 96 characters, including spaces	State, Province or Territory
bcountry	2 Letter Country Code	Country of Billing Address
bzip	Limit of 96 characters, including spaces	Zip or Postal Code
phone	Limit of 32 Characters	Customer's Phone Number
email	Limit of 254 Characters	Customer's Email Address

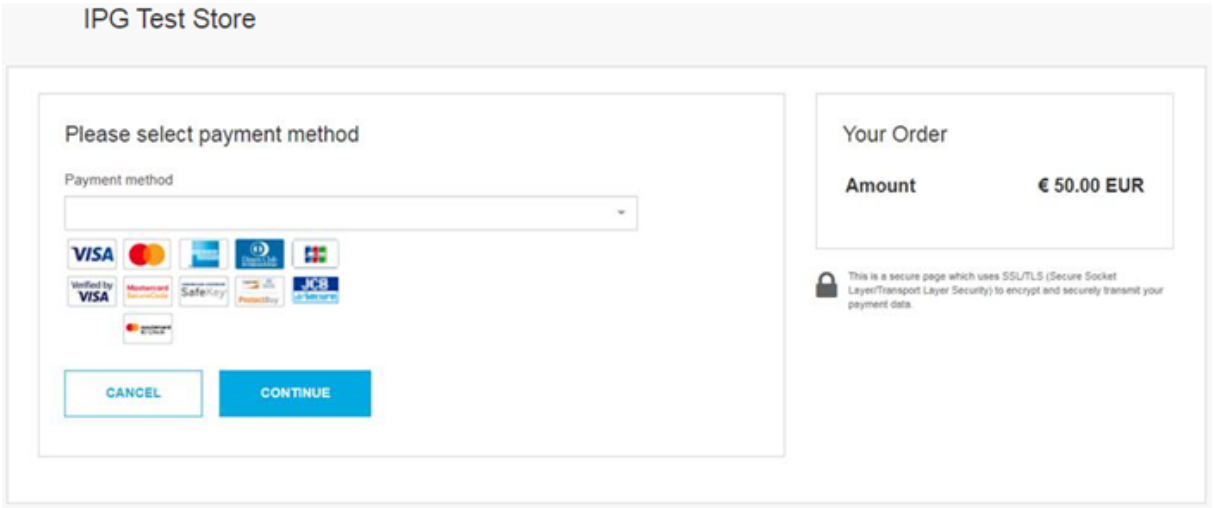
Additional required parameters are captured and populated automatically by the Gateway:

- Browser IP Address
- Browser Screen Height
- Browser Screen Width

In principle, it may occur that 3-D Secure authentications cannot 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 (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.

Credit card transactions with 3-D Secure hold in a pending status while cardholders search for their password or need to activate their card for 3-D Secure during their shopping experience. During this time when the final transaction result of the transaction is not yet determined, the payment gateway sets the Approval Code to „?:waiting 3dsecure". If the session expires before the cardholder returns from the 3-D Secure dialogue with his bank, the transaction will be shown as "N:-5103:Cardholder did not return from ACS".

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



3-D Secure Split Authentication

If your business or technical processes require the cardholder authentication to be separated from the payment transaction (authorization), you can use the transaction type ‘payer_auth’. This transaction type only performs the authentication (and stores the authentication results).

Example of a ‘payer_auth’ request:

```
<!-- #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="payer_auth">
    <input type="hidden" name="timezone" value="Europe/Berlin"/>
    <input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
    <input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
    <input type="hidden" name="hashExtended" value="<% call
createExtendedHash( "13.00","978" ) %>"/>
    <input type="hidden" name="storename" value="10123456789" />
    <input type="hidden" name="checkoutoption" value="combinedpage"/>
    <input type="hidden" name="paymentMethod" value="M"/>
    <input type="text" name="chargetotal" value="13.00" />
    <input type="hidden" name="currency" value="978"/>
    <input type="hidden" name="authenticateTransaction" value="true"/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Example of a ‘payer_auth’ response:

```
{txndate_processed=17/04/20 17:17:32,
ccbin=542606,
timezone=Europe/Berlin,
oid=C-2101f68a-45e9-4f3c-a6da-1337d5574717,
cccountry=N/A,
expmonth=12,
hash_algorithm=HMACSHA256
currency=978,
```

```
chargetotal=13.00,
approval_code=Y:ECI2/5:Authenticated,
hiddenSharedsecret=sharedsecret,
hiddenTxndatetime=2020:04:17-17:32:41,
expyear=2024,
response_hash=LarWYFSNgEToq13HlvyslX6hywi2T/nMn8jMY+1kxkI=,
response_code_3dsecure=1,
hiddenStorename=10123456789,
transactionNotificationURL=https://test.ipg-
online.com/webshop/transactionNotification,
tdate=1491824253,
ignore_refreshTime=on,
ccbrand=MASTERCARD,
txntype=payer_auth,
paymentMethod=M,
txndatetime=2020:04:17-17:32:41,
cardnumber=(MASTERCARD) ... 4979,
ipgTransactionId=84120276797,
status=APPROVED}
```

In a second step, you need to submit a payment transaction (‘sale’ or ‘preauth’) via the IPG Web Service API and reference it to the prior authentication. To review an example of a ‘sale’ transaction that refers to a previous ‘payer_auth’ transaction, please review the chapter 11.4 Split Authentication , in the Web Service API integration guide.

Dynamic 3-D Secure based on the card issuer’s country

With the Dynamic 3-D Secure product option, you can exclude specific card transactions from the 3-D Secure authentication based on a certain country selection (i.e.: issuing country) e.g.: Germany, Switzerland and Austria, while apply the standard 3-D Secure authentication process for other transactions with card from other countries.

You can improve the consumer experience for the cardholders from the selected countries, while the chargeback risk for such transactions is still with you.

If you have ordered this product option, the countries that should be excluded from the 3-D Secure authentication process can be set up for you by your local support team.

In case of some specific high-risk transactions, you can override this setting on transaction level and force the 3-D Secure authentication on a Transaction-by-Transaction basis, even if the card used is issued in a country, which has been defined by you as a country where 3-D Secure authentication should not be applied. To do it, you have to send the parameter ‘override3dsCountryExclusion’ set to “true” then the country setting will be ignored, and the 3-D Secure authentication process applied.

Field Name	Description, possible values and format
override3dsCountryExclusion	Optional parameter to be set either to ‘true’ or ‘false’. Set to ‘true’ if you would like to enforce 3-D Secure authentication, despite this country possibly being exempted from authentication due to the merchant configured list of countries, where 3-D Secure is not required.

9. MCC 6012 Mandate in UK

For UK-based Financial Institutions with Merchant Category Code 6012, Visa and MasterCard have mandated additional information of the primary recipient of the loan to be included in the authorization message.

If you are a UK 6012 merchant use the following parameters for your transaction request:

Field Name	Description, possible values and format
mcc6012BirthDay	Date of birth in format dd.mm.yyyy
mcc6012AccountFirst6	First 6 digits of recipient PAN (where the primary recipient account is a card)
mcc6012AccountLast4	Last 4 digits of recipient PAN (where the primary recipient account is a card)
mcc6012AccountNumber	Recipient account number (where the primary recipient account is not a card)
mcc6012Surname	Surname
mcc6012Zip	Post Code

If you are a UK 6051 and 7299 merchant, you can reuse the MCC 6012 parameters to send the optional data to be included in the authorization message. However, please note that you have to either populate all the parameters or none otherwise the transaction will be declined.

10. Data Vault

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

If you have ordered this feature, 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 IBAN and account holder name 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.

  If you want to assign multiple IDs to the same payment information record, you can submit the parameter 'hosteddataid' several times with different values in the same transaction.

  If you prefer not to assign a token yourself but want to let the gateway do this for you, send the parameter 'assignToken' and set it to 'true'. The gateway will then assign a token and include it in the transaction response as 'hosteddataid'.

  If you have use cases where you need some of the tokens for single transactions only (e.g.: for consumers that check out as a "guest", use the additional parameter 'tokenType' with the values 'ONETIME' (card details will only be stored for a short period of time) or 'MULTIPAY' (card details will be stored for use in future transactions).

- Initiate payment transactions using stored data

  If you stored cardholder information using the Data Vault option, 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 Fiserv'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.

  When using multiple Store IDs, it is possible to access stored card data records of a different Store ID then the one that has been used when storing the record. In that way you can for example use a shared data pool for different distributive channels. To use this feature, submit the Store ID that has been used when storing the record as the additional parameter 'hosteddatastoreid'.

- Avoid duplicate cardholder data for multiple records

To avoid customers using the same cardholder data for multiple user accounts, the additional parameter 'declineHostedDataDuplicates' can be sent along with the request. The valid values for this parameter are 'true'/'false'. If the value for this parameter is set to 'true' and the cardholder data in the request is already found to be associated with another 'hosteddataid', the transaction will be declined. There is no option to check, which existing 'hosteddataid' is holding duplicate cardholder data.

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

11. Recurring Payments

For credit card transactions, it is possible to install recurring payments using Connect. To use this feature, the following additional parameters will have to be submitted in the request:

Field Name	Possible Values	Description
recurringInstallmentCount	Number between 1 and 999	Number of installments to be made including the initial transaction submitted
recurringInstallmentPeriod	day week month year	The periodicity of the recurring payment
recurringInstallmentFrequency	Number between 1 and 99	The time period between installments

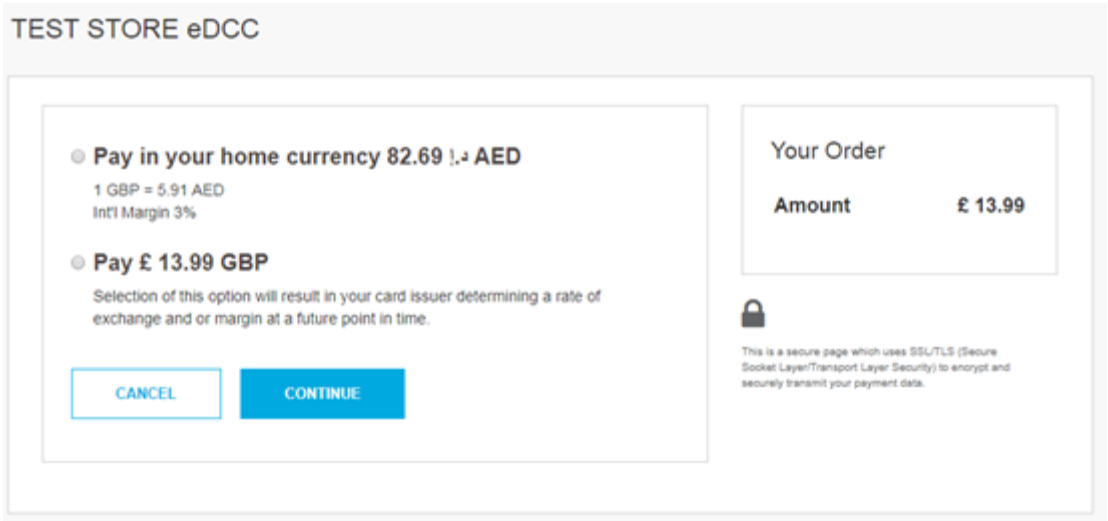
Note that the start date of the recurring payments will be the current date and will be automatically calculated by the system.

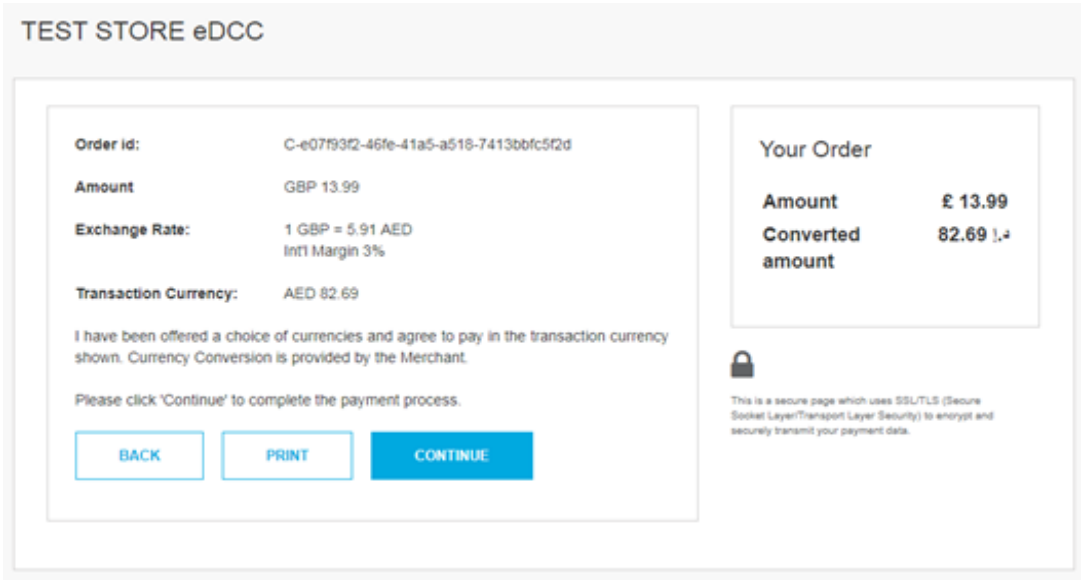
The recurring payments installed using Connect can be modified or cancelled using the Virtual Terminal or Web Service API.

12. Global Choice™ and Dynamic Pricing

With Fiserv's Global Choice™, foreign customers have the choice to pay for goods and services purchased online in their home currency when using their Visa or MasterCard credit card for the payment. The currency conversion is quick and eliminates the need for customers to mentally calculate the estimated cost of the purchase in their home currency. International Visa and MasterCard eCommerce customers can make informed decisions about their online purchases and eradicate any unexpected pricing or foreign exchange conversions on receipt of their monthly statements.

If your Store has been activated for this product option, the Connect solution automatically offers a currency choice to your customers if the card they use has been issued in a country with a currency that is different to your default currency.





Please note that for compliance reasons Fiserv's Global Choice can only be offered on transactions that take place in full at that time (e.g.: Sale, Refund) and not on any delayed settlement (e.g.: pre/post auth, recurring) due to the fluctuation of the rate of exchange.

Another option for your foreign customers is to display all pricing within your online store in their home currency using our Dynamic Pricing solution. This solution removes the need for your company to set pricing in any other currency other than your home currency.

Please see the Integration Guide for our Web Service API for details on how to request the exchange rates.

If your Store has been activated for this product option and you want to submit the payment transaction via our Connect solution, you need to send the DCC Inquiry ID that you have received along with the exchange rate request in the parameter 'dccInquiryId'.

You can also use the 'dccInquiryId' for cases where Global Choice is being offered and handled on your side (e.g.: within a mobile app). If the cardholder declines the currency conversion offer within your environment, the request parameter 'dccSkipOffer' can be set to 'true' so that the hosted consumer dialogue will automatically be skipped.

13. Purchasing Cards

Purchasing Cards offer businesses the ability to allow their employees to purchase items with a credit card while providing additional information on sales tax, customer code etc. When providing specific details on the payment being made with a Purchasing card favourable addendum interchange rates are applied.

There are three levels of details required for Purchasing Cards:

- Level I - The first level is the standard transaction data; no enhanced data is required at this level.
- Level II - The second level requires that data such as tax amount and customer code be supplied in addition to the standard transaction date. (Visa only have a level II option)
- Level III - The third level allows a merchant to pass a detailed accounting of goods and services purchased to the buyer. All the data for Level I and Level II must also be passed to participate in Level III. (Visa and MasterCard).

You can submit Level II and Level III data in your transaction request using the following parameters:

Field Name	Description, possible values and format
pcCustomerReferenceID	Merchant-defined reference for the customer that will appear on the customer's statement.
pcSupplierInvoiceNumber	Merchant-defined reference for the invoice, e.g.: invoice number.
pcSupplierVATRegistrationNumber	The Identification number assigned by the taxing authorities to the merchant.
pcTotalDiscountAmount	The total discount amount applied to a transaction (i.e.: total transaction percentage discounts, fixed transaction amount reductions or summarization of line item discounts).
pcTotalDiscountRate	The rate of the discount for the whole transaction.
pcVatShippingRate	The total freight/shipping amount applied to the transaction. Merchants can choose to deliver the contents of a single transaction in multiple shipments and this field reflects the total cost of those deliveries.
pcVatShippingAmount	The total freight/shipping amount applied to the transaction. Merchants can choose to deliver the contents of a single transaction in multiple shipments and this field reflects the total cost of those deliveries.
pcLineItemsJson	Line Item Details in JSON format. See table below for more information.

Purchasing Cards Line Item Details in JSON format:

Field Name	Description, possible values and format
CommodityCode	A reference to a commodity code used to classify purchased item.
ProductCode	A reference to a merchant product identifier, the Universal Product Code (UPC) of purchased item.
Description	Represents a description of purchased item.
Quantity	Represents a quantity of purchased items.
UnitOfMeasure	Represents a unit of measure of purchased items.
UnitPrice	Represents mandatory data for Level III transactions.
VATAmountAndRate	Represents a rate of the VAT amount, e.g.: 0.09 (means 9%).
DiscountAmountAndRate	Represents a rate of the discount amount, e.g.: 0.09 (means 9%).
LineItemTotal	This field is a calculation of the unit cost multiplied by the quantity and less the discount per line item. The calculation is reflected as: [Unit Cost * Quantity] - Discount per Line Item = Line Item Total.

14. Transaction Response

Response to your Success/Failure URLs

Upon completion, the transaction details will be sent back to the defined 'responseSuccessURL' or 'responseFailURL' as hidden fields. You can define these URLs in your transaction request. Alternatively, you can define them once in the Customisation section of our Virtual Terminal.

Field Name	Description, possible values and format
approval_code	Approval code for the transaction. The first character of this parameter is the most helpful indicator for verification of the transaction result. 'Y' indicates that the transaction has been successful 'N' indicates that the transaction has not been successful "?" indicates that the transaction has been successfully initialized, but a final result is not yet available since the transaction is now in a waiting status. The transaction status will be updated at a later stage.
oid	Order ID
refnumber	Reference number
status	Transaction status, e.g.: 'APPROVED', 'DECLINED' (by authorization endpoint or due to fraud prevention settings), 'FAILED' (wrong transaction message content/parameters, etc.) or 'WAITING' (asynchronous Alternative Payment Methods).
txndate_processed	Time of transaction processing
ipgTransactionId	Transaction identifier assigned by the gateway, e.g.: to be used for a Void
tdate	Identification for the specific transaction
fail_reason	Reason the transaction failed
response_hash	Hash-Value to protect the communication (see more below)
extended_response_hash	Hash-Value to protect the communication, where all response parameters are included in the hash calculation (see more below).
processor_response_code	The response code provided by the backend system. Please note that response codes can be different depending on the used payment type and backend system. While for credit card payments, the response code '00' is the most common response for an approval, the backend for giropay transactions for example returns the response code '4000' for successful transactions.
fail_rc	Internal processing code for failed transactions
terminal_id	Terminal ID used for transaction processing
ccbin	6 digit identifier of the card issuing bank
cccountry	3 letter alphanumeric ISO code of the cardholder's country (e.g.: USA, DEU, ITA, etc.) Filled with "N/A" if the cardholder's country cannot be determined or the payment type is not credit card
ccbrand	Brand of the credit or debit card: MASTERCARD VISA AMEX DINERSCLUB JCB CUP CABAL MAESTRO RUPAY BCMC SOROCRED Filled with "N/A" for any payment method which is not a credit card or debit card
schemeTransactionId	Credentials on file (COF) specific parameter. Returned in the response by a scheme for stored credentials transactions to be used in subsequent transaction request for future reference.

merchantAdviceMessage	Used in the authorization response, for both approved and declined transactions, indicating when consumer non-reloadable prepaid cards and single-use VCNs are recognized.

For 3-D Secure transactions only:

response_code_3dsecure	Return code indicating the classification of the transaction: **1** – Successful authentication (VISA ECI 05, MasterCard ECI 02) **2** – Successful authentication without AVV (VISA ECI 05, MasterCard ECI 02) **3** – Authentication failed (transaction declined by Gateway) **4** – Authentication attempt (VISA ECI 06, MasterCard ECI 01) **5** – Unable to authenticate / Directory Server not responding (VISA ECI 07) **6** – Unable to authenticate / Access Control Server not responding (VISA ECI 07) **7** – Cardholder not enrolled for 3-D Secure (VISA ECI 06) **8** – Invalid 3-D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS) **9** – Challenge requested (Whitelist prompt requested if challenge required) Please see note about blocking ECI 7 transactions in the 3-D Secure section of this document.

For Global Choice™ transactions only:

dcc_foreign_amount	Converted amount in cardholder home currency. Decimal number with dot (.) as a decimal separator.
dcc_foreign_currency	ISO numeric code of the cardholder home currency. This transaction is performed in this currency. String
dcc_margin_rate_percentage	Percent of margin applied to the original amount. Decimal number with dot (.) as a decimal separator.
dcc_rate_source	Name of the exchange rate source (e.g.: Reuters Wholesale Inter Bank). String
dcc_rate	Exchange rate. Decimal number with dot (.) as a decimal separator.
dcc_rate_source_timestamp	Exchange rate origin time. Integer - Unix timestamp (seconds since 1.1.1970).
dcc_accepted	Indicates if the card holder has accepted the conversion offer (response value 'true') or declined the offer (response value 'false').

For iDEAL transactions only:

accountOwnerName	Name of the owner of the bank account that has been used for the iDEAL transaction.
iban	IBAN of the bank account that has been used for the iDEAL transaction.
bic	BIC of the bank account that has been used for the iDEAL transaction.

For Fraud Detect transactions only:

fraudScore	Score returned based on Fraud Detect check.

When your store is enabled for SEPA Direct Debit as part of the TeleCash from Fiserv offering:

bname	Name of the account holder of the bank account that has been used.
iban	IBAN of the bank account that has been used.
bic	BIC is provided only if the German IBAN has been used.
mandateReference	Mandate reference as returned for the first direct debit transaction.
mandateDate	Date of the initial direct debit transaction as returned for the first transaction.

For merchants using the Fiserv Global Merchant Acquiring model only:

associationResponseCode	The raw association value tells exactly how the issuer has responded to the transaction without any mapping done either by the authorization platform or the gateway. It will be returned only for Visa, MasterCard, Amex, and Discover.

For merchants activated for the MasterCard and Visa account updater service:

When your store is enabled for the MasterCard real-time account updater service on the gateway, and you have the payment information vaulted on your side then when applicable the updates are sent as part of the gateway response, and you have to react upon it accordingly i.e.: update the account number for a token when you store PAN and a token on your side.

updatedPAN	Updated primary account number
updatedExpirationDate	Updated expiration date
updatedAccountStatusType	Updated account status with possible values: Account Status: Meaning/Action ACCOUNT_CHANGED: Either the account number or account number along with the expiration date are being updated. Use the new account information going forward. The new account information should also be used in case of authorization reversals. ACCOUNT_CLOSED: Closed account advice. This account has been closed. Try alternate method of payment on subsequent authorization or retries. EXPIRY_CHANGED: Expiration date change. Use the new expiry information going forward. This should also be used in case of authorization reversals.
accountUpdaterErrorCode	Error codes that indicate the system/server communication errors.

For merchants operating on the Fiserv Nashville and activated for the Visa or MasterCard real-time account updater service:

When you are processing on the Fiserv Nashville endpoint and your store is enabled for the Visa real-time account updater service or for the MasterCard real-time account updater service on the gateway then you can expect the updates to be sent as part of the gateway response. When you have the payment information vaulted on your side then you have to react upon it accordingly i.e.: update the account number and the parameter ‘hosteddataid’ for a token when you store PAN and a token on your side.

updatedPAN	Updated primary account number
updatedExpirationDate	Updated expiration date
updatedAccountStatusType	Updated account status with possible values: Account Status: Meaning/Action ACCOUNT_CHANGED: Either the account number or account number along with the expiration date are being updated. Use the new account information going forward. The new account information should also be used in case of authorization reversals.

	ACCOUNT_CLOSED	Closed account advice. This account has been closed. Try alternate method of payment on subsequent authorization or retries.
	EXPIRY_CHANGED	Expiration date change. Use the new expiry information going forward. This should also be used in case of authorization reversals.
	CONTACT_CARDHOLDER	Contact cardholder advice. Account updater cannot provide updates on this account owing to restrictions from cardholder. Use an alternate method of payment or contact customer to get one.
hosteddataid	Returned when the updates have been applied. New (TransArmor) token has to be used in place of the old/previous one. Note that the old/previous token will not be deleted but will be honored by the gateway till the old payment information (account number) will be honored by the scheme (Visa).	
accountUpdaterErrorCode	Error codes that indicate the system/server communication errors.	

For Partial Approval:

partiallyApprovedAmount	Available balance as a partial amount approved.
status	Transaction status: 'PARTIALLY APPROVED'. This unique status allows you to identify this transaction and subtract the partially approved amount from the total transaction amount, and request another form of payment, using split-tender functionality.

Additionally, when using your own error page for negative validity checks (full_bypass=true):

fail_reason_details	Comma separated list of missing or invalid variables. Note that 'fail_reason_details' will not be supported in case of payplus and fullpay mode
invalid_cardholder_data	**true** – if validation of card holder data was negative **false** – if validation of card holder data was positive but transaction has been declined due to other reasons

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

Please consider when integrating that new response parameters may be added from time to time in relation to product enhancements or new functionality.

How to generate a hash for a response

Make sure to use the parameter 'response_hash' to recheck if the received transaction response has really been sent by Fiserv to protect you from fraudulent manipulations. The value is created with a HMAC Hash using the following parameter string:

```
approval_code|chargetotal|currency|txndatetime|storename
```

Shared secret ('`sharedsecret`') will be used as a key in HMAC to calculate the hash with the above hash string. The hash algorithm is the same as the one that you have set in the transaction request.

Please note that you have to implement the response hash validation, when doing so remember to store the 'txndatetime' that you have submitted with the transaction request in order to be able to validate the response hash. Furthermore, you must always use the https-connection (instead of http) to prevent eavesdropping of transaction details.

You can also use the parameter 'extended_response_hash' to include all response parameters in the hash calculation. Please contact your local support team if you want to enable this feature. This will be

managed with a specific setting performed on your account (service configuration 'extendedResponseHashSupported').

Creating the extended response hash

Step 1: Retrieve all non-empty Gateway specified response parameters and then remove the parameter 'extended_response_hash' from your list, so that it will not get included in the hash calculation. Consider also that shared secret will be used as a key in HMAC to calculate the hash and the hash algorithm must be the same as the one that you have set in the transaction request.

Step 2: Sort the response parameters in ascending order of the parameter names, where the upper-case characters come before the lower case (based on ASCII value). Join the parameters' values to one string with pipe separator (use only parameters' values and not the parameters' names).

Step 3: Pass the created string to the HMAC algorithm while using shared secret ('sharedsecret') as a key for calculating the hash value.

Step 5: Encode the result of HMAC with Base64 to generate the extended response hash.
Only HMAC algorithm (i.e.: HMACSHA256, HMACSHA384 or HMACSHA512) is supported for generating the extended response hash.

Server-to-Server Notification

In addition to the response, you receive in hidden fields to your 'responseSuccessURL' or 'responseFailURL', the payment Gateway can send server-to-server notifications with the above result parameters to a defined URL. This is especially useful to keep your systems in synch with the status of a transaction. To use this notification method, you can specify an URL in the Customisation section of the Virtual Terminal or submit the URL in the following additional transaction parameter 'transactionNotificationURL'.

Please note that:

- The Transaction URL is sent as received therefore please don't add additional escaping (e.g.: using %2f for a Slash (/).
- No SSL handshake, verification of SSL certificates will be done in this process.
- The Notification URL needs to listen on port 443 (https) – other ports are not supported.

The response hash parameter for validation (using the same algorithm that you have set in the transaction request) 'notification_hash' is calculated as follows:

```
chargetotal|currency|txndatetime|storename|approval_code
```

Shared secret ('`sharedsecret`') will be used as a key in HMAC to calculate the hash with the above hash string.

Such notifications can also be set up for the recurring payments that get automatically triggered by the gateway. Please contact your local support team to get a shared secret ('rcpSharedSecret') agreed for these notifications. You can configure your Recurring Transaction Notification URL ('rcpTransactionNotificationURL') in the Customisation section of the Virtual Terminal.

In case of the recurring transactions the response hash parameter 'notification_hash' is calculated differently as follows:

```
chargetotal+rcpSharedSecret+currency+txndatetime+storename+approval_code
```

The shared secret ('rcpSharedSecret') is part of the string (it is not used as a key in HMAC to calculate the hash with the hash string). Moreover, the response hash parameter for the recurring transaction notifications is calculated with the SHA256-value (as the default value).

15. Appendix I – How to generate a hash for a request

If you are using an HTML form to initiate a transaction, your request needs to include a security hash for verification of the message integrity.

The hash (parameter 'hashExtended') needs to be calculated using all non-empty gateway specified request parameters in ascending order of the parameter names, where the shared secret (parameter 'sharedsecret') must be used as the secret key for calculating the hash value. The gateway sorts the request parameters in the "natural order". For strings this means the "Lexicographic Order", thus the upper-case characters come before the lower case (based on ASCII value).

The request parameters that are not specified in our solution can still be submitted in your request to the gateway, but they must be excluded from the hash calculation. They will be ignored during processing and returned in the response.

When you are using Direct Post, there is also an option where you do not need to know the card details (PAN, CVV and Expiry Date) for the hash calculation. This will be managed with a specific setting performed on your store. Please contact your local support team if you want to enable this feature.

Creating the hash with all parameters

Transaction request values used for the hash calculation can be considered as a set of mandatory as well as optional gateway specified request parameters depending on the way you decide to build your request. See an example below:

- chargetotal= 13.00
- checkoutoption = combinedpage
- currency= 978
- hash_algorithm=HMACSHA256
- paymentMethod=M
- responseFailURL=https://localhost:8643/webshop/response_failure.jsp
- responseSuccessURL=https://localhost:8643/webshop/response_success.jsp
- storename=10123456789
- timezone= Europe/Berlin
- transactionNotificationURL=https://localhost:8643/webshop/transactionNotification
- txndatetime= 2021:09:06-16:43:04
- txntype=sale
- sharedsecret=sharedsecret (to be used as the secret key for calculating the hash value)

The steps below provide the guidelines on how to calculate a hash, while using the values from our example.

Step 1. Extended hash needs to be calculated using all non-empty gateway specified request parameters in ascending order of the parameter names, where the upper-case characters come before the lower case (based on ASCII value). Join the parameters' values to one string with pipe separator (use only parameters' values and not the parameters' names).

stringToExtendedHash =
13.00|combinedpage|978|HMACSHA256|M|https://localhost:8643/webshop/response_failure.jsp|https://localhost:8643/webshop/response_success.jsp|10123456789|Europe/Berlin|https://localhost:8643/webshop/transactionNotification|2021:09:06-16:43:04|sale

Corresponding hash string does not include 'sharedsecret', which has to be used as the secret key for the HMAC instead.

Step 2. Pass the created string to the HMACSHA256 algorithm and using shared secret as a key for calculating the hash value.

HmacSHA256(stringToExtendedHash, sharedsecret)

Step 3. Encode the result of HMACSHA256 with Base64 and pass it to the gateway as part of your request.

Base64:
EapafBqqOF6N/kch8USkHPGh+fwSko24h6FpQnQHfQ8=

```
<input type="hidden" name="hashExtended" value="
EapafBqqOF6N/kch8USkHPGh+fwSko24h6FpQnQHfQ8="/>
```

Only HMAC algorithm (i.e.: HMACSHA256, HMACSHA384 or HMACSHA512) is supported for generating the extended request hash.

16. Appendix II – ipg-util.asp

```
<!-- google CryptoJS for HMAC -->
<script LANGUAGE=JScript RUNAT=Server src="script/cryptoJS/crypto-js.min.js"></script>
<script LANGUAGE=JScript RUNAT=Server src="script/cryptoJS/enc-base64.min.js"></script>
<script LANGUAGE=JScript RUNAT=Server>
    var today = new Date();
    var txndatetime = today.formatDate("Y:m:d-H:i:s");

    /*
        Function that calculates the hash of the following parameters as an example:
       - chargetotal
       - checkoutoption
       - currency
       - hash_algorithm
       - paymentMethod
       - responseFailURL
       - responseSuccessURL
       - storename
       - timezone
       - transactionNotificationURL
       - txndatetime
       - txntype
       - and sharedsecret as the secret key for calculating the hash value
     */

    function createExtendedHash(chargetotal, currency) {
        // Please change the storename to your individual Store Name
        var storename = "10123456789";
        // NOTE: Please DO NOT hardcode the secret in that script. For example read it from
a database.
        var stringToExtendedHash =
chargetotal|checkoutoption|currency|hash_algorithm|paymentMethod|responseFailURL|responseSu
ccessURL|storename|timezone|transactionNotificationURL|txndatetime|txntype;

        var hashHMACSHA256 = CryptoJS.HmacSHA256(stringToExtendedHash, sharedSecret);
        var extendedhash = CryptoJS.enc.Base64.stringify(hashHMACSHA256);

        Response.Write(extendedhash);
    }

    function getDateTime() {
        Response.Write(txndatetime);
    }
</script>
```

17. Appendix III – ipg-util.php

```
<!DOCTYPE HTML>
<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">


<fieldset>
        <legend>IPG Connect Request Details</legend>
        <p>
            <label for="storename">Store ID:</label>
            <input type="text" name="storename" value="10123456789" readonly="readonly" />
        </p>
        <p>
            <label for="timezone">Timezone:</label>
            <input type="text" name="timezone" value="Europe/London"  readonly="readonly"/>
        </p>
        <p>
            <label for="chargetotal">Transaction Type:</label>
            <input type="text" name="txntype" value="sale" readonly="readonly" />
        </p>
        <p>
            <label for="chargetotal">Transaction Amount:</label>
            <input type="text" name="chargetotal" value="13.00" readonly="readonly" />
        </p>
        <p>
            <label for="currency">Currency (see ISO4217):</label>
            <input type="text" name="currency" value="978" readonly="readonly" />
        </p>
        <p>
            <label for="txndatetime">Transaction DateTime:</label>
            <input type="text" name="txndatetime" value="<?php echo getDateTime(); ?>"/>
        </p>
        <p>
            <label for="hashExtended">Hash Extended:</label>
               <input type="text" name="hashExtended" value="<?php echo
createExtendedHash('13.00', '978'); ?>"  readonly="readonly" />
        </p>
        <p>
            <label for="hashExtended">Hash Algorithm :</label>
            <input type="text" name="hash_algorithm" value="HMACSHA256" readonly="readonly"
/>
        </p>
        <p>
            <label for="hashExtended">Checkout option :</label>
            <input type="text" name="checkoutoption" value="combinedpage"
readonly="readonly" />
        </p>
        <p>
            <input type="submit" id="submit" value="Submit" />
        </p>
    </fieldset>

</form>

<?php

function getDateTime() {
```

```
    return date("Y:m:d-H:i:s");
}

function createExtendedHash($chargetotal, $currency) {
// Please change the store Id to your individual Store ID
// NOTE: Please DO NOT hardcode the secret in that script. For example read it from a
database.
$sharedSecret = "sharedsecret";
$separator = "|";
$storeId= "10123456789";
$timezone= "Europe/London";
$txntype= "sale";
$checkoutoption = "combinedpage";

$stringToHash = $chargetotal . $separator . $checkoutoption . $separator . $currency .
$separator . "HMACSHA256" . $separator . $storeId . $separator . $timezone. $separator .
date("Y:m:d-H:i:s") . $separator . $txntype;

$hash = base64_encode(hash_hmac('sha256', $stringToHash, $sharedSecret, true));
return $hash;
}

?>
</body>
</html>
```

The above is the working PHP example, to run it you can copy the above and paste it on https://www.w3schools.com/php/phptryit.asp?filename=tryphp_function1

18. Appendix IV – Currency Code List

Currency name	Currency code	Currency number
CFA Franc BCEAO	XOF	952
Afghan Afghani	AFN	971
Albanian	ALL	008
Algerian Dinar	DZD	012
Argentine Peso	ARS	032
Armenian Dram	AMD	051
Aruban Florin	AWG	533
Australian Dollar	AUD	036
Azerbaijanian Manat	AZN	944
Bahamian Dollar	BSD	044
Bahrain Dinar	BHD	048
Bangladeshi Taka	BDT	050
Barbados Dollar	BBD	052
Belarussian Ruble	BYN	933
Belize Dollar	BZD	084
Bermudian Dollar	BMD	060
Bolívar Soberano	VES	928
Bolivian Boliviano	BOB	068
Bosnian Convertible	BAM	977
Botswana Pula	BWP	072
Brazilian Real	BRL	986
British Pound	GBP	826
Bruneian Dollar	BND	096
Bulgarian Lev	BGN	975
Burundi Franc	BIF	108
Cambodian Riel	KHR	116
Canadian Dollar	CAD	124
Cape Verdean (Cabo Verde Escudo)	CVE	132
Cayman Islands Dollar	KYD	136
Central African CFA	XAF	950
CFP	XPF	953
Chilean Peso	CLP	152
Chinese Renminbi	CNY	156
Colombian Peso	COP	170
Comorian Franc	KMF	174
Congolese Franc	CDF	976
Costa Rican Colon	CRC	188
Croatian Kuna	HRK	191
Cuban Peso	CUP	192
Czech Koruna	CZK	203
Danish Krone	DKK	208
Djiboutian Franc	DJF	262
Dobra	STN	930
Dominican Peso	DOP	214
East Caribbean Dollar	XCD	951
Egyptian Pound	EGP	818
Ethiopian Birr	ETB	230
Euro	EUR	978
Falkland Islands Pound	FKP	238
Fijian Dollar	FJD	242
Gambian Dalasi	GMD	270
Georgian Lari	GEL	981
Gibraltar Pound	GIP	292
Gourde	HTG	332
Guatemalan Quetzal	GTQ	320
Guinea Franc	GNF	324
Guyanese Dollar	GYD	328
Honduran Lempira	HNL	340

Hong Kong Dollar	HKD	344
Hungarian Forint	HUF	348
Iceland Krona	ISK	352
Indian Rupee	INR	356
Indonesian Rupiah	IDR	360
Iranian Rial	IRR	364
Iraqi Dinar	IQD	368
Israeli New Shekel	ILS	376
Jamaican Dollar	JMD	388
Japanese Yen	JPY	392
Jordanian Dinar	JOD	400
Kazakhstani Tenge	KZT	398
Kenyan Shilling	KES	404
Kuwaiti Dinar	KWD	414
Kwanza	AOA	973
Laotian Kip	LAK	418
Lebanese Pound	LBP	422
Liberian Dollar	LRD	430
Libyan Dinar	LYD	434
Lilangeni	SZL	748
Loti	LSL	426
Macau Pataca	MOP	446
Macedonian Denar	MKD	807
Malagasy Ariary	MGA	969
Malawian Kwacha	MWK	454
Malaysian Ringgit	MYR	458
Maldivian Rufiyaa	MVR	462
Mauritian Rupee	MUR	480
Mexican Peso	MXN	484
Moldovan Leu	MDL	498
Mongolian Tugrik	MNT	496
Moroccan Dirham	MAD	504
Mozambique Metical	MZN	943
Mvdol	BOV	984
Myanmar Kyat	MMK	104
Nakfa	ERN	232
Namibia Dollar	NAD	516
Nepalese Rupee	NPR	524
Netherlands Antillean Guilder	ANG	532
New Zealand Dollar	NZD	554
Ngultrum	BTN	064
Nicaraguan Cordoba Oro	NIO	558
Nigerian Naira	NGN	566
Norwegian Krone	NOK	578
Omani Rial	OMR	512
Ouguiya	MRU	929
Pakistani Rupee	PKR	586
Panamanian Balboa	PAB	590
Papua New Guinean Kina	PGK	598
Paraguayan Guarani	PYG	600
Peruvian Nuevo Sol	PEN	604
Philippine Peso	PHP	608
Polish Zloty	PLN	985
Qatari Rial	QAR	634
Romanian New Leu	RON	946
Russian Ruble	RUB	643
Rwandan Franc	RWF	646
Saint Helena Pound	SHP	654
Salvador Colon	SVC	222
Samoan Tala	WST	882
Saudi Rihal	SAR	682
Serbian Dinar	RSD	941

Seychelles Rupee	SCR	690
Sierra Leonean	SLL	694
Singapore Dollar	SGD	702
Solomon Islands Dollar	SBD	090
Som	KGS	417
Somali Shilling	SOS	706
South African Rand	ZAR	710
South Korean Won	KRW	410
South Sudanese Pound	SSP	728
Sri Lanka Rupee	LKR	144
Sudanese Pound	SDG	938
Surinamese Dollar	SRD	968
Swedish Krona	SEK	752
Swiss Franc	CHF	756
Syrian Pound	SYP	760
Taiwan Dollar	TWD	901
Tajikistani Somoni	TJS	972
Tanzanian Shilling	TZS	834
Thai Baht	THB	764
Tongan Pa'anga	TOP	776
Trinidad and Tobago Dollar	TTD	780
Tunisian Dinar	TND	788
Turkish Lira	TRY	949
Turkmenistan New Manat	TMT	934
Uganda Shilling	UGX	800
Ukrainian Hryvnia	UAH	980
UAE Dirham	AED	784
US Dollar	USD	840
Uruguayan Peso	UYU	858
Uzbekistan Sum	UZS	860
Vanuatu Vatu	VUV	548
Vietnamese Dong	VND	704
Yemeni Rial	YER	886
Zambian Kwacha	ZMW	967
Zimbabwe Dollar	ZWL	932

19. Appendix V – Payment Method List

If you let your consumer select the payment method in your website or want to define the payment method yourself, submit the parameter 'paymentMethod' in your transaction request. If you do not submit this parameter, the gateway will display a hosted page to the consumer to choose from the payment methods that are enabled for your store and supported for the combination of the consumer's country and the transaction currency.

Payment Method	**Value**
Adancard (local Argentinian brand)	ADANCARD
Alipay*	aliPay
Alipay (China Domestic)	aliPay_domestic
American Express	A
Apple Pay on the web	applePay
Argencard (local Argentinian brand)	ARGENCARD
Asian local payment methods via Razer Merchant Services	asian_apm
Automatica (local Argentinian brand)	AUTOMATICA
Bancontact	BCMC
BBPS (local Argentinian brand)	BBPS
BLIK	blik_amp_ng
Cabal	CA
Cabal (local Argentinian brand)	CABAL_ARGENTINA
Cetelem (local Argentinian brand)	CETELEM
CFSA (local Argentinian brand)	CFSA
Clarin 365 (local Argentinian brand)	CLARIN_365
Club del Este (local Argentinian brand)	CLUB_ESTE
Club la Nacion (local Argentinian brand)	CLUB_LA_NACION
Confiable (local Argentinian brand)	CONFIABLE
Consumax (local Argentinian brand)	CONSUMAX
Coopeplus (local Argentinian brand)	COOPEPLUS
Credimas (local Argentinian brand)	CREDIMAS
Crediguia (local Argentinian brand)	CREDIGUIA
Dina Card (local Serbian brand)	DI
Diners	C
DuitNow	duitnow
Eftpos (local Australian brand)	EFTPOS
Elebar (local Argentinian brand)	ELEBAR
ELO (local Brazilian brand)	EL
eps*	eps
Equated Monthly Installments (EMI)	emi
Falabella CMR (local Argentinian brand)	FALABELLA_CMR
Faster Payment System (FPS)	fasterPaymentSystem
Favacard (local Argentinian brand)	FAVACARD
FinanYa (local Argentinian brand)	FINANYA
Giropay	giropay
Google Pay on the web	googlePay
GrabPay	grabPay
Grupar (local Argentinian brand)	GRUPAR
Hiper (local Brazilian brand)	hiper
HiperCard (local Brazilian brand)	hipercard
iDEAL	ideal
Italcred (local Argentinian brand)	ITALCRED
Ired (local Argentinian brand)	IRED
JCB	J
Kadicard (local Argentinian brand)	KADICARD
Korean Payment Service (Korea Domestic)	kps
Local Wallets India	indiawallet
Local Wallets (Japan Domestic)	sbps_other_payments
Maestro	MA
Maestro UK	maestroUK
MasterCard	M
Mercury	mercury

Mira (local Argentinian brand)	MIRA
MU.DO.N (local Argentinian brand)	MUDON
Multibanco*	multibanco
MyBank*	mybank
Naranja (local Argentinian brand)	NARANJA
Nativa (local Argentinian brand)	NATIVA
Netbanking (India)	netbanking
Nevada (local Argentinian brand)	NEVADA
Payconiq	payconiq
Payit (a payment solution using Open Banking technology)	natwest_payit
PayLater by ICICI Bank	payLater
PayPal	paypal
PayNow	paynow
Payit (UAE's digital wallet)	fab_payit
Patagonia 365 (local Argentinian brand)	PATAGONIA365
Paysafecard*	paySafeCard
PostFinance Card	postfinance_card
PostFinance E-Finance	postfinance
PostFinance Pay	postfinance_pay
Przelewy24 (P24)*	przelewy24
Pyme Nacion (local Argentinian brand)	PYME_NACION
Qida (local Argentinian brand)	QIDA
RuPay	RU
SafetyPay*	safetypay
Samsung Pay	samsung_pay_wallet
SEPA Direct Debit	debitDE
SEPA Direct Debit*	direct_debit-apm
Sorocred	SO
Su Crédito (local Argentinian brand)	SU_CREDITO
Sidecreer (local Argentinian brand)	SIDECREER
Tarjeta Shopping (local Argentinian brand)	TARJETA_SHOPPING
Tarjeta Sol (local Argentinian brand)	TARJETA_SOL
Token Banking (Open banking)	token_banking
Trustly*	trustly
Tuya (local Argentinian brand)	TUYA
Ultra (local Argentinian brand)	MAXIULTRA
Unired (local Argentinian brand)	UNIRED
UnionPay	CUP
UnionPay (China Domestic)	CUP_domestic
UnionPay (Japan Domestic)	sbps_other_payments
Visa (Credit/Debit/Electron/Delta)	V
Visa Mobile	visa_mobile_wallet

(*) Only supported in a collecting model through the Fiserv Local Payments offering.

20. Appendix VI – PayPal Legacy

Refer to the following information when integrating PayPal as a payment method.

Transaction types mapping

Connect Transaction Type (txntype)	PayPal operation
Sale	SetExpressCheckoutPayment (sets *PaymentAction* to *Authorization* in *SetExpressCheckout* and *DoExpressCheckoutPayment* requests)
Preauth	GetExpressCheckoutDetails
sale – with additional parameters for installing a Recurring Payment	DoExpressCheckoutPayment*
Postauth	DoCapture (,DoReauthorization)
Void	DoVoid

Address handling

If you pass a complete set of address values within your request to Connect (name, address1, zip, city and country within billing and/or shipping address), these values will be forwarded to PayPal, setting the PayPal parameter 'addressOverride' to '1'.

Please note that it is an eligibility requirement for PayPal's Seller Protection that the shipping address will be submitted to PayPal.

If you submit no or incomplete address data within the Connect request, no address data will be forwarded to PayPal and the PayPal parameter 'addressOverride' will not be set.

Regardless of that logic, the payment gateway will always store the shipTo address fields received from PayPal in the GetDetails request in the ShippingAddress fields, possibly overwriting values passed in the request to Connect (such overwriting depends on the above logic).

* If you want to use PayPal's Reference Transactions feature for recurring payments, please contact PayPal upfront to verify if your PayPal account meets their requirements for this feature.

Recurring Payment Transaction

You have to submit a SALE transaction request with the corresponding parameters to install the recurring payments. The first transaction is always conducted immediately along with the request.

The subsequent transactions are executed by the Gateway's scheduler, via the API Web Service, as defined during the initial SALE transaction with the installation.

21. Appendix VII – PayPal Checkout

Please note, in case you are integrated with PayPal Legacy solution, you are required to migrate to PayPal Checkout solution upon earliest convenience. Please reach out to your customer service team for assistance.

New PayPal framework adheres to the latest compliance and security standards and offers smoother checkout process with the lightbox.

In addition to mandatory fields, it is also recommended to include PayPal specific fields, as listed below.

Field Name	Type *	Description
checkoutoption	M	Set the value for this parameter to 'combinedpage'.
customParam_client-meta-id	C	Represents a specific risk tracking id (Risk Session Correlation ID / Client Metadata ID). Example: `<input type="hidden" name="customParam_client-meta-id" value="123456"/>` To be sent together with "`customParam_stc-content`" parameter.
customParam_stc-content	C	Use this parameter to provide content for the PayPal Set Transaction Context (STC) to be used for the Risk Services analysis. Example: `<input type="hidden" name="customParam_stc-content" value="{"name":"jan", "id":"1234"}"/>`
item1 - item999	O	The line items are regular Connect integration key-value parameters (URL-encoded) that allow you to send basket information in the following format: *id;description;quantity;item_total_price;sub_total;vat_tax;shipping;* As part of a line item, you can also submit your VAT and shipping cost (as a delivery/shipping fee). Example: `<book>;<The Hobbit>;<1>;<4.50>;<2.92>;<0.58>;<1.00>`
invoicenumber	C	Represents PayPal invoice ID, the field must be populated if available during request submission. This value will be visible in your PayPal seller account while reviewing a transaction.
paymentMethod	O	Optional parameter, you can submit a value 'paypal' directly. If you do not submit this parameter, the Gateway will display a Hosted Payment Page to choose from all the payment methods activated for your store.
sname	O	Ship-to Name Alphanumeric characters, spaces, and dashes limited to 96
saddr1	O	Shipping City Limit of 96 characters, including spaces
scountry	O	Country of Shipping Address 2 letter country code in the ISO alpha code format (e.g.: DE)
szip	O	Zip or Postal Code Limit of 24 characters, including spaces
shippingPreference	O	Use this parameter to indicate the way you would like to handle your customer's shipping address. Available values:

		• 'SET_PROVIDED_ADDRESS': Customer can see the address details inside the wallet but cannot change it. The address details are provided as part of the Gateway response. • 'NO_SHIPPING': Customer cannot see the address details inside the wallet nor can change it. No address details are provided as part of the Gateway response. Example: `<input type="hidden"name="shippingPreference"value="NO_SHIPPING"/>`
txntype	M	Supported transaction types: • 'sale' • 'preauth' • 'void' • 'postauth'

*(M)=Mandatory (O)=Optional (C)=Conditional

Recurring payments

If you want to use PayPal's Reference Transaction feature, please contact PayPal to verify if your PayPal business account meets the PayPal requirements for this feature.

Your account must have set up Data Vault / Hosted Data service and you need to submit the parameter 'chargetotal' with a zero value in your preauthorization request. Such transaction request is always used to create a Billing Agreement ID on the PayPal side, returned to you in the field 'hostedDataId'.

Transaction Response

Response parameter	Description
approval_code	Transaction approval code. Initial character indicates the transaction status. 'Y' – Transaction approved 'N' – Transaction declined '?:waiting PAYPAL' – Transaction has been initiated but a final result is not yet available
status	Transaction status, e.g.'APPROVED', 'DECLINED', 'FAILED' or 'WAITING'
hosteddataid	Stores "Billing Agreement ID"
sname	Ship-to Name
saddr1	Shipping Address
scity	Shipping City
scountry	Country of Shipping Address
szip	Zip or Postal Code
fail_reason	Indicates why the transaction was declined.
ipgTransactionId	It is mapped to the PayPal Custom. You will be able to see this value in your PayPal seller account while reviewing a transaction.
refnumber	Represents the PayPal Transaction ID. The value will be visible in your PayPal seller account while reviewing a transaction.

Initiating a Return transaction

To initiate a Return transaction using the Gateway's Virtual Terminal or REST API.

22. Appendix VIII – Fraud Detect

Refer to the following information when you are signed up to Fiserv's Fraud Detect product to have card transactions reviewed for a fraud scoring.

You can submit a payment transaction to the gateway, which routes it to the appropriate authorization front-end. The gateway receives the authorization response. If an approval is received, the gateway submits the transaction to Fraud Detect including authorization response details (e.g.: AVS/Card Code match).

In case you use the Fraud Detect product and want to pass the details for the scoring, you need to pass the following parameter for:

Mobile device details:

- customParam_deviceRiskId
- customParam_deviceRiskAPIKey
- customParam_deviceRiskHost

Device intelligence:

- customParam_deviceIntelligenceVendor
- customParam_deviceIntelligenceSessionID

Whether the payment was made inside or outside the store (e.g.: pay at pump or in petrol station):

- customParam_inStoreOutStore

Pump number used at a petrol station:

- customParam_pumpNumber

Customer type (eg: Retail, Restaurant, Grocery, Mobile etc.):

- customParam_customerType

Purchase type (eg: gift card reload, gift card purchase etc.):

- customParam_purchaseType

Example:

```
<input type="hidden" name="customParam_deviceRiskId" value="*****"/>
```

These fields are handled in the same way as other optional request parameters. The gateway stores these parameters and passes them on to Fraud Detect. These parameters have no impact on the transaction processing flow.

In the response from the gateway (parameter 'fraudScore') you receive the score returned based on the Fraud Detect check performed.

23. Appendix IX – Local Payments

Refer to the following information when you have ordered this product option and your store is enabled for the Local Payments offering.

The Local Payments solution offers a unique combination of global coverage, a single contracting and integration experience, and a broad and expanding portfolio of local payment methods.

Local Payments, also often referred to as Alternative Payment Methods, are defined as payment transactions where neither credit/debit cards or paper currencies are used as the form of payment. These payment methods are primarily used in eCommerce and mCommerce transactions, although some solutions are making a push for adoption at point of sale locations. In many markets, they are more commonly used than credit/debit cards.

Local Payments differ from card/association processing in a number ways. They are generally designed to meet local needs and used in one or a limited number of markets. Unlike traditional credit/debit card processing, pricing across these payment methods is not uniform and retail pricing depend on local costs and merchant industries (e.g.: high-risk vs. low-risk). Local Payments offerings and user experiences also vary greatly, though most are quite different from debit/credit user experiences.

Consumer demand and preference are driving the growth in new methods of payment across the globe. In fact, local payment methods are growing more rapidly than major card schemes, and merchant demand for non-card (credit/debit) methods of payment is on the rise. These new payment methods deliver many benefits to both merchants and consumers.

Local Payments help you reach and securely process payments from a broader base of consumers in each local market, reduce shopping cart abandonment/improve conversion and improve customer experiences. They enable more consumers to easily and confidently shop online (i.e.: provide easy access to secure payment methods for those that are unbanked and/or without credit or debit cards), expand their ability to access international merchants and enable them to 'pay their way,' all of which improve their shopping experiences and overall satisfaction.

Payment methods supported in a collecting model through the Fiserv Local Payments offering

Payment Name	Payment Type	Customer's Country / Region	Timeouts
Alipay	eWallet	China	240 minutes (4 hours)
Bancontact	Local Card Brand	Belgium	60 minutes (1 hour)
eps	Bank Payment	Austria	60 minutes (1 hour)
giropay	Bank Payment	Germany	5 minutes
iDEAL	Bank Payment	Netherlands	30 minutes
Multibanco	Bank Payment	Portugal	10 080 minutes (7 days)
MyBank	Bank Payment	Italy	30 minutes
paysafecard	Prepaid Voucher	Austria, Australia, Belgium, Bulgaria, Canada, Croatia, Cyprus, Czech Republic, Denmark, Finland, France, Georgia, Germany, Gibraltar, Greece, Hungary, Ireland, Italy, Liechtenstein, Lithuania, Luxembourg, Malta, Mexico, Netherlands, New Zealand, Norway, Peru, Poland, Portugal, Romania,	120 minutes (2 hours)

		Slovakia, Slovenia, Spain, Sweden, Switzerland, U.K., Uruguay	
Przelewy24	Bank Payment	Poland	60 minutes (1 hour)
SEPA Direct Debit	Direct Debit	European Economic Area (E.E.A.)	Model A: 1 440 minutes (24 hours) Model C: N/A
SOFORT Banking / Klarna Pay Now	Bank Payment	Austria, Belgium, Germany, Italy, Netherlands, Spain	60 minutes (1 hour)
Trustly	Bank Payment	Denmark, Estonia, Finland, Germany, Italy, Malta, Netherlands, Norway, Poland, Spain, Sweden, U.K.	10 080 minutes (7 days)
SafetyPay	Cash & Bank Payments	Brazil, Mexico & Peru (more than 50 Local Payment Methods)	1 800 minutes (30 hours) Customizable in the payment request

Initiating a Sale transaction

A Sale transaction for most Local Payments requires a direct interaction with the consumer who needs to be redirected to the payment method's screens (e.g.: the login page of the consumer's bank or a wallet provider) and back to your website after all required steps are completed.

As we handle all the required redirections to the various stakeholders for you, all you need to do is to post a form to a URL with the parameters and values required for the transaction.

URL for Test Transactions

```
https://test.ipg-online.com/connect/gateway/processing
```

You will get the production URL with your production account credentials.

When building a request, independently of the payment method, there are some mandatory fields that need to be included in every request for a Sale transaction.

Example of a form with the minimum number of fields:

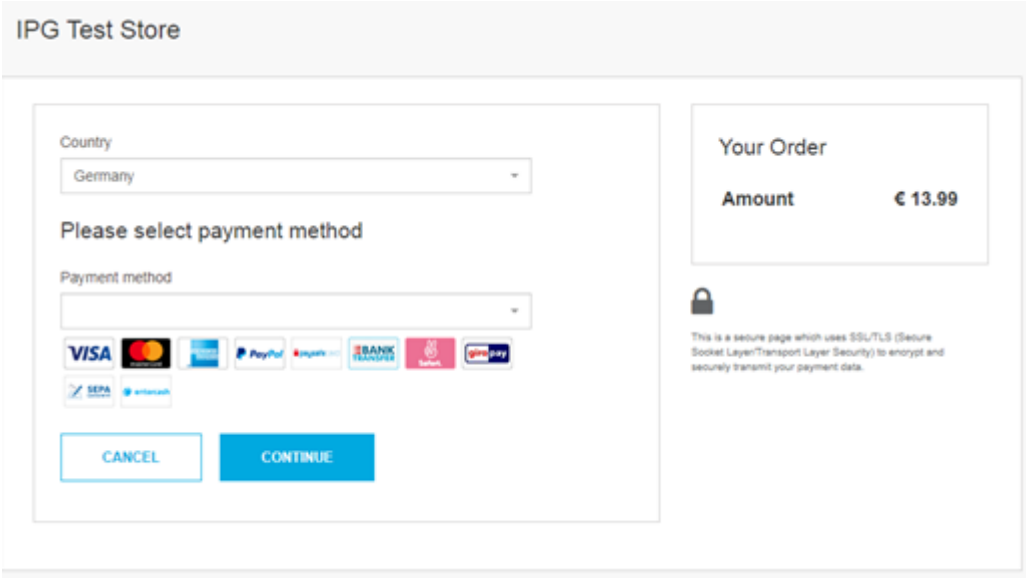
```
<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">
<input type="hidden" name="txntype" value="sale">
<input type="hidden" name="timezone" value="America/New_York"/>
<input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
<input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
<input type="hidden" name="hashExtended" value="<% call createExtendedHash (
"13.00","840" ) %>"/>
<input type="hidden" name="storename" value="541234567" />
<input type="hidden" name="checkoutoption" value="combinedpage"/>
<input type="text" name="chargetotal" value="13.00" />
<input type="hidden" name="currency" value="840"/>
<input type="submit" value="Submit">
</form>
```

Other generic fields to be considered
(M)=Mandatory (O)=Optional

Field Name	Type	Description, possible values and format
checkoutoption	M	Set the value for this parameter to 'combinedpage' for a payment process where the payment method choice and the typical next step (e.g.: entry of card details or selection of bank) in consolidated in a single page.
paymentMethod	O	You can submit the parameter 'paymentMethod' in your transaction request relevant for a selected local payment method, as defined in Appendix V. If you do not submit this parameter, gateway will display a page to your consumer to choose from the payment methods that are supported for the combination of the consumer's country and the transaction currency.
bname	M	The consumer's name, e.g.: Albert Einstein. This is required for all Local Payments transactions. It is required for all Local Payments transactions, so we recommend including it in every Sale transaction request. If you do not submit this field, a hosted page will be displayed to the consumer to capture the name.
bcountry	M	The consumer's country in 2 Letter Country Code format, e.g.: US for the United States or DE for Germany. It is required for all Local Payments transactions, so we recommend to include it in every Sale transaction request. If you do not submit this field and the payment method requires it, a hosted page will be displayed with the country that we have identified based on IP address and the option to change the country, if not appropriate.

Many of the payment methods are available for customers coming from a certain country. In the scenarios where you use the hosted payment page for payment selection, the gateway can display to your consumers a hosted page with only these payment methods that are set up for your store and supported for the combination of the consumer's country and the transaction currency. This validation is done either based on the submitted billing country ('bcountry') or the customer's IP address.

See below an example of a hosted payment page in the checkout option 'combinedpage', where the country is pre-set to 'Germany' based on the customer's IP address but still it can be changed via a dedicated drop-down, where else the payment methods are limited based on the combination country/currency.



When building a request for a specific payment method, apart from the mandatory fields required for Sale transaction and some generic fields to be considered, you might also have to include some specific fields in your transaction request.

Payment method specific fields to be considered
(M)=Mandatory (O)=Optional (C)=Conditional

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.

<table>
<tr><th>Field Name</th><th>Relevant for</th><th>Description, possible values and format</th></tr>
<tr><td>apmbic</td><td>iDEAL (O)</td><td>The value for an iDEAL issuer's bank that can be used for bypassing the iDEAL hosted payment page and redirecting to a pre-selected banking page. Submit a valid BIC value:
<table>
<tr><td>Bank Name</td><td>BIC</td></tr>
<tr><td>Rabobank</td><td>RABONL2U</td></tr>
<tr><td>ABN AMRO</td><td>ABNANL2A</td></tr>
<tr><td>Van Lanschot Kempen</td><td>FVLBNL22</td></tr>
<tr><td>Triodos Bank</td><td>TRIONL2U</td></tr>
<tr><td>ING Bank</td><td>INGBNL2A</td></tr>
<tr><td>SNS Bank</td><td>SNSBNL2A</td></tr>
<tr><td>ASN</td><td>ASNBNL21</td></tr>
<tr><td>RegioBank</td><td>RBRBNL21</td></tr>
<tr><td>Knab</td><td>KNABNL2H</td></tr>
<tr><td>Bunq</td><td>BUNQNL2A</td></tr>
<tr><td>Revolut</td><td>REVOLT21</td></tr>
<tr><td>Nationale Nederlanden</td><td>NNBANL2G</td></tr>
</table>
Please note, you must ensure to display the official iDEAL logo on the Hosted Payments Page. This informs your customers that they can pay with iDEAL.

See iDEALs logos here:
https://www.ideal.nl/en/businesses/logos/</td></tr>
<tr><td>apmPaymentMethod</td><td>SafetyPay (O)</td><td>Enables the desired payment channel. Submit a value as:
• 'cash' - for the cash payments
• 'online' - for the online bank transfer payments
If empty/null, all bank and cash payment methods will be made available.</td></tr>
<tr><td>timeout</td><td>SafetyPay (O)</td><td>You can submit a value for a transaction timeout in minutes to encourage your consumer to pay within a certain timeframe.</td></tr>
<tr><td>customerid</td><td>Trustly (M)</td><td>Unique reference to identify the consumer (or transaction) for example from your CRM system.</td></tr>
<tr><td>beneficiaryid</td><td>Trustly (C)</td><td>When asked for compliance purposes, submit ID, username, hash, or anything uniquely identifying the ultimate beneficiary.</td></tr>
<tr><td>beneficiaryname</td><td>Trustly (C)</td><td>When asked for compliance purposes, submit the ultimate beneficiary's full name.</td></tr>
<tr><td>beneficiarycountrycode</td><td>Trustly (C)</td><td>When asked for compliance purposes, submit the ultimate beneficiary's country of residence (two-letter ISO code). Example: ES</td></tr>
<tr><td>beneficiaryaddress</td><td>Trustly (C)</td><td>When asked for compliance purposes, submit the ultimate beneficiary's street address (street, zip code, city), excluding the country.
Example: Main street 1, 12345, Barcelona</td></tr>
<tr><td>termsaccepted</td><td>Przelewy24 (P24) (O)</td><td>Determines whether the GDPR page is displayed before the payment page. Submit a values as:
• '0' - the GDPR page is shown
• '1' - the GDPR page is not shown</td></tr>
<tr><td>p24method</td><td>Przelewy24 (P24) (O)</td><td>The numeric identifier of a bank. Used for bypassing the P24 hosted payment page and redirecting to a pre-selected banking page. Submit a valid numeric value:
<table>
<tr><td>Bank Name</td><td>Value</td></tr>
</table></td></tr>
</table>

		<table><tr><td>BLIK - PSP</td><td>154</td></tr><tr><td>Euro Bank</td><td>94</td></tr><tr><td>mBank - mTransfer</td><td>25</td></tr><tr><td>Płacę z IKO</td><td>135</td></tr><tr><td>Płacę z Orange</td><td>146</td></tr><tr><td>Przekaz tradycyjny</td><td>178</td></tr><tr><td>Raiffeisen Bank PBL</td><td>102</td></tr><tr><td>Użyj przedpłaty</td><td>177</td></tr><tr><td>Przekaz/Przelew tradycyjny</td><td>1000</td></tr></table> Example: Sending ‘p24method’ set to ’25’ as part of your request redirects your consumer to the bank page. If an invalid value is sent for |p24method’ you consumer will be redirected to the P24 bank selection page. Due to the General Data Protection, P24 requires that the consumers accept their terms and conditions as an intermediate redirect before redirecting to the bank page.
email	Przelewy24 (P24) (M) SEPA Direct Debit (M)	Consumer’s email address. For SEPA Direct Debit this parameter is mandatory only when you want to use the *out-of-the-box* solution offered by Fiserv.
iban	SEPA Direct Debit (M)	Consumer’s IBAN - International Bank Account Number (up to 34 digits)
mandateDate	SEPA Direct Debit (M)	To be populated with the initial mandate signature date. It is a mandatory to submit a mandateDate in case of recurring collections.
mandateReference	SEPA Direct Debit (M)	To be populated with the mandate reference It is mandatory to submit a mandateReference in case of recurring collections.
mandateType	SEPA Direct Debit (M)	Sequence type of Direct Debit, defaults to ‘single’. Values: • single - Direct Debit is executed once • firstCollection - First Direct Debit in a series of recurring • recurringCollection – Follow-up Direct Debit in a series of recurring • finalCollection – Last Direct Debit in a series of recurring
mandateUrl	SEPA Direct Debit (M)	This parameter is mandatory only when you want to manage the SEPA Direct Debit mandates on your side. To be populated with the valid URL of the SEPA mandate to enable the Risk and Compliance department to access the details.
bname	SEPA Direct Debit (M)	Name of the bank account owner that will be debited.
baddr1	SEPA Direct Debit (C)	Mandatory if IBAN belongs to EFTA and associated country. Street name and house number of the bank account owner that will be debited.
bcity	SEPA Direct Debit (C)	Mandatory if IBAN belongs to EFTA and associated country. City of the bank account owner that will be debited.
bcountry	SEPA Direct Debit (C)	Mandatory if IBAN belongs to EFTA and associated country. Country of the bank account owner that will be debited (2 letter country code).
bzip	SEPA Direct Debit (C)	Mandatory if IBAN belongs to EFTA and associated country. Zip or postal code of the bank account owner that will be debited.
mobileMode	Alipay (O)	You can submit this parameter with the value ‘true’ to enable Alipay for mobile web i.e.: the mobile enabled variant of Alipay.
appToAppURL	Bancontact (O)	Custom app URL for redirecting the consumer back to the app, which triggered the payment.

		Alternatively, consider usage of the universal links.
language	Bancontact (O)	The hosted payment page features 4 languages. Supported languages are: • 'en_US' - English • 'nl_NL' - Dutch • 'de_DE' - German • 'fr_FR' – French (default when no parameter specified)

Initiating a Return transaction

When Return is supported for a selected local payment, you can initiate a Return transaction with a reference to the Transaction ID of the original Sale transaction to the API Web Service. Please see details in the Integration Guide for the Web Service API, chapter Generic Transaction Type for Voids and Returns.

There is the limit for the amount of Return transaction to a maximum of 100 000 either EUR or USD, which are the only currencies that are applicable for this limit. Returns using other currencies will not be limited.

Options for SEPA Direct Debit

When you manage SEPA Direct Debit mandates on your side you can use these in combination with the Local Payments offering by submitting the reference and date of the mandate as well as a link to the mandate itself. This is especially useful in cases where you have a large number of mandates on file from previously used solutions and want to continue to collect the mandates and generate the pre-notification emails yourself.

Single payment or recurring payment

Field Name	M/O/C	Description
email	O	Consumer's email address to generate the pre-notification emails by yourself.
iban	M	Consumer's IBAN - International Bank Account Number (up to 34 digits).
bname	M	Name of the bank account owner that will be debited.
baddr1	C	Mandatory if IBAN belongs to EFTA and associated country. Street name and house number of the bank account owner that will be debited.
bcity	C	Mandatory if IBAN belongs to EFTA and associated country. City of the bank account owner that will be debited.
bzip	C	Mandatory if IBAN belongs to EFTA and associated country. Zip or postal code of the bank account owner that will be debited.
mandateType	O	Sequence type of Direct Debit, defaults to 'single'. Values: single - Direct Debit is executed once firstCollection - First Direct Debit in a series of recurring recurringCollection – Follow-up Direct Debit in a series of recurring finalCollection – Last Direct Debit in a series of recurring
mandateReference	M	To be populated with the mandate reference.
mandateDate	M	To be populated with the initial mandate signature date.
mandateUrl	M	To be populated with the valid URL of the SEPA mandate to enable the Risk and Compliance department to access the details.

When you do not want to manage the SEPA Direct Debit mandates on your side, you can instead use the *out-of-the-box* solution offered by Fiserv, where we collect the mandates and generate the pre-notification emails. Upon receiving the valid transaction request, the gateway displays a hosted page to

your customer with the mandate text and assigned mandate reference. It gives your consumer the option to consent or reject this mandate. As part of the gateway's response, you receive the mandate reference and mandate date, which have to be used in case of the subsequent payments under this mandate.

Single payment or First payment in recurring series

Field Name	M/O/C	Description
email	M	Consumer's email address to generate the pre-notification emails by the gateway.
iban	M	Consumer's IBAN - International Bank Account Number (up to 34 digits).
bname	M	Name of the bank account owner that will be debited.
baddr1	C	Mandatory if IBAN belongs to EFTA and associated country. Street name and house number of the bank account owner that will be debited.
bcity	C	Mandatory if IBAN belongs to EFTA and associated country. City of the bank account owner that will be debited.
bzip	C	Mandatory if IBAN belongs to EFTA and associated country. Zip or postal code of the bank account owner that will be debited.
mandateType	O	Sequence type of Direct Debit, defaults to 'single'. Values: single - Direct Debit is executed once firstCollection - First Direct Debit in a series of recurring

Follow-up payments in recurring series

Field Name	M/O	Description
email	M	Consumer's email address to generate the pre-notification emails by the gateway.
iban	M	Consumer's IBAN - International Bank Account Number (up to 34 digits).
bname	M	Name of the bank account owner that will be debited.
baddr1	C	Mandatory if IBAN belongs to EFTA and associated country. Street name and house number of the bank account owner that will be debited.
bcity	C	Mandatory if IBAN belongs to EFTA and associated country. City of the bank account owner that will be debited.
bzip	C	Mandatory if IBAN belongs to EFTA and associated country. Zip or postal code of the bank account owner that will be debited.
mandateType	M	Sequence type of Direct Debit. Values: recurringCollection – Follow-up Direct Debit in a series of recurring finalCollection – Last Direct Debit in a series of recurring
mandateReference	M	To be populated with the mandate reference from the response.
mandateDate	M	To be populated with the initial mandate signature date from the response.

Transaction response

Among all the details sent back to the defined response URLs as the transaction result, you might especially consider:

approval_code	Approval code for the transaction. The first character of this parameter is the most helpful indicator for verification of the transaction result. 'Y' indicates that the transaction has been successful 'N' indicates that the transaction has not been successful

	"?" indicates that the transaction has been successfully initialized, but a final result is not yet available since the transaction is now in a waiting status. The transaction status will be updated at a later stage.
status	Transaction status, e.g.: 'APPROVED', 'DECLINED' (by authorization endpoint or due to fraud prevention settings), 'FAILED' (wrong transaction message content/parameters, etc.) or 'WAITING' (asynchronous Alternative Payment Methods).
fail_reason	Reason the transaction failed. Only if 'status' is 'DECLINED' possible values are: • INPUT_DATA - There was a problem in the data passed/submitted • LOCAL_ERROR - Local system error • LOCAL_DECLINE - The transaction has been declined by the authorization endpoint • REMOTE_ERROR - There was a remote processing error • REMOTE_DECLINE - The transaction has been declined by a remote system (e.g.: payment process authentication failed) • TIMEOUT - There was a timeout while waiting for the transaction result • UNKNOWN - Transaction failed for unknown reasons (also default in reporting in case of succeeded transactions) • USER_ABORT - The user aborted the payment process

Specific response parameters when your store is enabled for SEPA Direct Debit

mandateReference	Mandate reference as returned for the first direct debit transaction.
mandateDate	Date of the initial direct debit transaction as returned for the first transaction.
plannedDueDate	When you manage SEPA Direct Debit mandates on your side, you receive a UTC date (YYYY-MM-DD) with the planned due date (the earliest day that the funds will be debited from the consumer's account).

Transaction status

You need to be aware that, based on how many alternative payment methods work, there is always a chance for a transaction to get approved (succeed) after it has been initially marked as declined (failed) by the gateway. The approved status on the other hand is final. This logic is applicable only for the Sale transactions, but not for the Returns transactions, which are immediately getting either approved or declined.

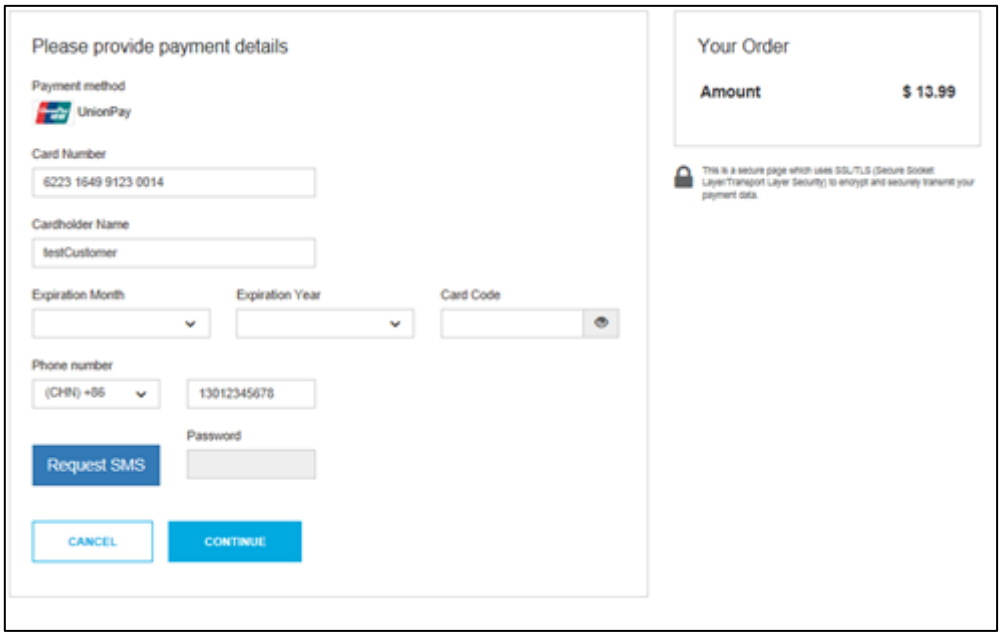
You will receive a notification in case approved-after-declined (succeed-after-failed) happens. The fail_reason will be overwritten once the succeeded notification has been sent.

24. Appendix X – UnionPay SecurePlus

Refer to the following information when your store is enabled for UnionPay SecurePlus.

SecurePlus is a part of the UnionPay Online Payments (UPOP) eCommerce payment solution designed by UnionPay for merchants who want to reduce the risk of fraudulent transactions, similar to 3-D Secure.

When enabling your store for UnionPay SecurePlus, you would have to provide the UPOP MID specific in order request UnionPay to verify enrolment and to send a SMS code to your customers (as the card holders). However, you can also decide to allow your consumers to skip authentication, if you are happy to hold the responsibility of transactions without the security check.



The generic fields to be considered:

Field Name	Description, possible values and format
paymentMethod	You can submit the parameter 'paymentMethod' in your transaction request for UnionPay as defined in Appendix V. If you do not submit this parameter, gateway will display a page to your consumer to choose from the payment methods activated for your store.
bname	You can submit the consumer's name (cardholder's name) in your transaction request. In some cases, when integration the checkout option 'combinedpage', the consumer's name might be required as a mandatory parameter. If you do not submit this field, gateway will display a page to your consumer to capture the name.
phone	You can submit the consumer's phone number in your transaction request only as digits limited to: 4-15 digits and without the phone country code extension, which is set to +86 by default. If you do not submit this field, a hosted page will be displayed to the consumer to capture the phone and allow him to change the phone country code extension, when applicable. The phone is mandatory when going through security check since it is the phone number that is checked against the card number unless you are happy to hold the responsibility of transactions without the security check and your store is configured accordingly to skip this authentication then your customer would be able to perform a credit card transaction, where no phone number would be needed.

You can also consider integrating UnionPay SecurePlus via the gateway's Web Service API. See the further information in the Integration Guide for the Web Service API.

25. Appendix XI – China Domestic

Refer to the following information when your store is enabled for China Domestic processing.

The China Domestic solution includes: China UnionPay and Alipay with a redirection of the consumer to pages in Chinese language providing your customers with a familiar shopping experience.

Initiating a Sale transaction

A Sale transaction requires a direct interaction with the consumer who needs to be redirected to the payment method's screens and back to your website after all required steps are completed.

As the gateway handles all the required redirections, all you need to do is to post a form to a URL with the parameters and values required for a Sale transaction.

When building a request for China Domestic apart from the mandatory fields you will also need to include some specific fields in your transaction request.

The payment method specific fields to be considered:
(M)=Mandatory (O)=Optional

<table>
<tr><th>Field Name</th><th>Relevant for</th><th>Description, possible values and format</th></tr>
<tr><td>item1</td><td>aliPay_domestic (M)
CUP_domestic (M)</td><td>Submit exactly one line item parameter with four (4) property values in the following format:

id;description;quantity;item_total_price

Transaction request without a line item or with multiple line items will be declined.

Example: 100018;The Hobbit;1;3.50

<table>
<tr><th>Position</th><th>Property</th><th>Description</th></tr>
<tr><td>1</td><td>id</td><td>Product code (编码Code) from "Product category list"

Productcatelog_3.3.xlsx</td></tr>
<tr><td>2</td><td>description</td><td>Product name</td></tr>
<tr><td>3</td><td>quantity</td><td>Quantity of product(s)</td></tr>
<tr><td>4</td><td>item_total_price</td><td>Product price</td></tr>
</table>
</td></tr>
<tr><td>customerid</td><td>CUP_domestic (M)</td><td>Unique reference to identify the consumer.</td></tr>
<tr><td>custom_domesticBankId</td><td>CUP_domestic (M)</td><td>Submit a bank identifier for reporting purpose in relation to promotions with local Chinese banks. Max length 8.</td></tr>
</table>

26. Appendix XII – Korea Domestic

Refer to the following information when your store is enabled for Korea Domestic (Korean Payment System) processing.

Initiating a Sale transaction

As the gateway handles all the required redirections, all you need to do is to post a form to a URL with the parameters and values required for a Sale transaction.

When building a request for Korea Domestic apart from the mandatory fields you will also need to include some custom fields in your transaction request.

The payment method specific fields to be considered:
(M)=Mandatory (O)=Optional

Field Name	M/O	Description, possible values and format
paymentMethod	O	Set the value for this parameter to 'kps'.
checkoutoption	M	Set the value for this parameter to 'combinedpage'.
oid	M	Unique order ID, alphanumeric string (32 max).
mobileMode	M	Set the value for this parameter to 'true'. This will lead your customer to a payment page flow that has been specifically designed for mobile devices.
numberOfInstallments	O	Optional parameter to set the installment options that will be offered for that transaction. Possible values are: • '00': lumpsum (No Installments) • '02'~'60': 2 months ~60 months • Otherwise by default: 0~ 12 months (If installment is configured) You need to be configured for installments during boarding.
localTax	O	Optional parameter to submit an amount for Local Tax. Please ensure the sub total amount plus local tax equals the charge total.
subTotal	O	Optional parameter to submit a tax free amount. Please ensure the sub total amount plus local tax equals the charge total.
customParam_kps_ItemInfo	M	Type of purchased item, alphanumeric string (1 max). Possible values are: • '1': Goods • '2': Online content
customParam_kps_CcProdDesc	M	Description of purchased items to be displayed on the KPS payment page, alphanumeric string (256 max).
customParam_kps_VAExpireDate	O	The Effective/Valid Time for Virtual Bank Acct. Optionally required for Virtual Account payment method. Format: YYYYMMDDHH24MISS
customParam_kps_SelectPayment	O	Optional parameter to restrict available payment methods for a transaction. Possible values are: • 'ALL': All (Default) • 'CRDT': Card • 'HP': Mobile carrier billing • 'ACCT': Bank account transfer • 'VACT': Virtual bank account transfer • 'IC': CashCard • 'SPAY': e-Wallet (e.g.: Samsung Pay, 11Pay, Payco)
customParam_kps_LangType	O	Optional parameter to choose the Payment Page Display Language. Possible values are: • 'HAN': Korean (Default)

		• 'ENG': English
customParam_kps_BillType	O	Optional parameter to choose the Billing Type for the transaction. Possible values are: • '00': Taxable (Default) • '01': Duty free
customParam_kps_CardSelect	O	Optional parameter to restrict available issuers for a transaction. All issuers are available if this parameter is not set. Possible values are: • '00' - All Issuers • '01' - BC • '07' - KB • '02' - Shinhan • '03' - Samsung • '05' - Lotte • '12' - NH • '27' - Hana • '04' - Hyundai • '13' - CITI • '22' - Jeju • '14' - Woori • '11' - Suhyup • '24' - Jeonbok • '23' - Kwangju • '17' - Shinhyup • '09' - All of International Issuers
customParam_kps_escrowYn	O	Optional parameter to manage ESCROW options. Applicable only for Bank Account Transfer and Virtual Bank Account Transfer Payment methods. Possible values are: • 'S': Customer can choose to use ESCROW (Default) • 'Y': ESCROW is required regardless of customer's choice • 'N': Customer cannot user ESCROW Escrow is a legal arrangement in which a third party temporarily holds large sums of money or property until a particular condition has been met (such as the fulfillment of a purchase agreement).
customParam_kps_VAPhoneNumber	O	Optional parameter to receive SMS related to Virtual Account. If not set, Customer has the option to enter it in the payment screen. Applicable only for Virtual Account.
customParam_kps_cashYn	O	Optional parameter to enable Cash receipt feature. Possible values are: • 'Y': Yes • 'N': No • 'M': Mandatory Issue
customParam_kps_ SupportDate	O	Optional parameter to enable customers to know date until which the listed price is supported.
customParam_kps_FXFlag	O	Foreign exchange payment flag. This parameter in mandatory only when the transaction must happen in any currency other than KRW. If the transaction should happen in KRW, then don't use this parameter.

After a transaction request is submitted to the gateway, the consumer will be redirected to the Korean Payment System selection page, where the payment can be completed. Upon completion, you will receive a response from the gateway including specific details related to this payment method that carry out additional payment information, which you can pass to your customer.

Please note, that highlighted parameters such as ‘kpsPaymentBrand’ and ‘kpsPaymentMode’ carry out additional payment information.

```
sending parameters: {kpsPaymentBrand=KB,
txndate_processed=13/01/22 06:54:33,
timezone=Asia/Calcutta,
number_of_installments=00,
oid=XXXXXXXXXXX,
kpsPaymentMode=CreditCard,
cccountry=N/A,
endpointTransactionId=XXXXXX,
currency=410,
processor_response_code=0000,
chargetotal=1004,
approval_code=Y:XXXXXX:4390898371:PPX :84390898371,
hiddenSharedsecret=XXXXXXXXXX,
hiddenTxndatetime=2022:01:13-11:24:22,
response_hash=b0cac9d86b758cca5dd6f2ea03d3585fa41a8f0c,
hiddenStorename=8101000003,
transactionNotificationURL=https://test.ipg-
online.com/webshop/transactionNotification, ignore_deploymentType=JBoss,
tdate=1642053273,
installments_interest=false,
ignore_refreshTime=on,
ccbrand=N/A,
txntype=sale,
paymentMethod=kps,
txndatetime=2022:01:13-11:24:22,
ipgTransactionId=84390898371,
status=APPROVED}
```

Initiating a Return transaction

Transaction type Return allows you to return funds to a customer’s card against an existing order on the gateway. To perform a return of Korean Domestic transaction, you will need the order id and additional KPS specific parameters to be submitted in your Return request to our gateway.

Full or partial Returns can be done via our Web Service AP or the Virtual Terminal interface.

27. Appendix XIII - Debit Disbursement

Refer to the following information only when you are operating in US and your store is enabled to allow credit transaction processing.

Debit Disbursement (Visa OCT, MasterCard MoneySend) allows businesses to disburse funds in real-time, directly to a debit card. Faster payouts can increase loyalty and satisfaction, reduce costs for businesses. The Debit Disbursement solution is cheaper, faster, more convenient and more traceable than traditional payment methods. It facilitates payments and transfers such as:

- Fund disbursements by e-commerce marketplaces
- Government disbursements (such as VAT refunds)
- Forex and binary option trade payouts
- Affiliate and contractor payouts
- Expense reimbursements
- Corporate and manufacturing rebates
- Insurance claims

The functionality for disbursements can be used with Direct Post and hosted payment page integrations. It is also available for REST API originated transactions.

The funding source may be a credit card, debit card, prepaid card, or bank account, but the receiving account must be a debit card. Note currently only Visa and MasterCard brand debit cards can be used as the recipient for debit disbursements.

For person-to-person payments (P2P) and P2PBankInit - Person to Person Bank Initiated, the merchant must perform the operation as two individual transactions, one for funding (Pull transaction to debit funds from sender) and one for disbursement (Push transaction to receive funds by receiver).

Disbursement types supported:

- P2P - Person to Person
- P2PBankInit - Person to Person Bank Initiated
- MerchDisb - Merchant Disbursement
- FundsDisb - Funds Disbursement
- Pay Roll Pension Disbursement
- MerchInitMT - Money Transfer – Merch Initiate

Pull transactions for getting funds from the sender can be done using the transaction type ‘sale’, while Push transactions for the disbursement to the receiver using the transaction type ‘credit’.

When building a request for Pull transaction apart from the mandatory fields required for Sale transaction, you can also need to include some custom fields in your transaction request.

The payment method specific fields to be considered:
(M)=Mandatory (O)=Optional

Field Name	M/O	Description, possible values and format
		Sender Information
sdrName	O	Customer’s Name
sdrAccount	O	Account Number

sdrReference	O	Reference Number
sdrAddr	O	Address
sdrCity	O	City
sdrState	O	State
sdrCountry	O	Country
sdrZip	O	Zip
sdrPhone	O	Phone
sdrBirthDate	O	Birthdate

When building a request for Push transaction apart from the mandatory fields required for Credit transaction, you will also need to include some fields in your transaction request. Note that the possibility to send 'credit' using the Connect interface is restricted and needs to be enabled for your store.

Field Name	**M/O**	**Description, possible values and format**
		Billing Information
bname	M	Customer's Name
		Receiver Information
rAccountNumber	O	Account Number
rReferenceNumber	O	Reference Number

The transactions will be presented in the Virtual Terminal Reports as 'sale' for Pull transactions and as 'return' for Push transactions.

28. Appendix XIV – Digital Wallets

Refer to the following information only when you are integrating Google Pay or/and Apple Pay on the web as a payment method.

Google Pay on the web
Google Pay is a digital wallet solution provided by participating banks and supported by Google.
It allows users to store cards from participating banks. To learn more about Google Pay, please visit https://pay.google.com/about/.

Initiating a transaction (Checkout Process)
The checkout process for Google Pay can be initiated with a “Google Pay” button that you place on your website either as a specifically alternative checkout option or next to other payment methods that you offer.

When consumers click this button, you construct a Sale or PreAuth transaction request, with the required parameters including the payment method parameter. This will take your customers to a hosted page from where they can be redirected to the Google Pay payment screen, with list of cards added to customer Google Pay wallet. Selecting the card by customers from the list and clicking the ‘Pay’ button would complete the payment.

Alternatively, you can let your customer select the payment method on the gateway’s hosted payment method selection page. If you prefer that option, simply do not submit the payment method parameter.

Apple Pay on the web
Apple Pay on the web allows making purchases on the web in Safari on your iPhone, iPad, or Mac, you can use Apple Pay without having to create an account or fill out lengthy forms. Moreover, with Touch ID on MacBook Air and MacBook Pro, paying takes just a touch and is quicker, easier, and more secure than ever before. To learn more about Apple Pay on the web, please visit https://developer.apple.com/documentation/apple_pay_on_the_web.

Initiating a transaction (Checkout Process)
The checkout process for Apple Pay on web can be initiated in Safari browser with “Apple Pay” button that you place on your website either as a specifically alternative checkout option or next to other payment methods that you offer.

When consumers click this button, you construct a Sale or PreAuth transaction request, with the required parameters including the payment method parameter. This will take your customers to a hosted page from where they can be redirected directly to the Apple Pay payment screen, with list of cards added to customers’ Apple Pay wallet. Selecting the card by customers from the list and authenticate using Touch id/Face id on Apple device would complete the payment.

Alternatively, you can let your customer select the payment method on the gateway’s hosted payment method selection page. If you prefer that option, simply do not submit the payment method parameter.

Apple Pay on the web transaction can only be initiated with Apple’s Safari browser and authorization from an iOS device like iPhone, Apple Watch or MacBook.

The generic fields to be considered:

Field Name	M/O	Description, possible values and format
checkoutoption	M	Set the value for this parameter to ‘combinedpage’
paymentMethod	O	Set the value for this parameter to ‘googlePay’ or ‘applePay’ If you do not submit this parameter, gateway will display a page to your consumer to choose from the payment methods activated for your store.

29. Appendix XV – Network Tokenisation

One you retrieved Network Token from your provider, you can use our Connect ‘combinedmode’ to submit its value in ‘cardnumber’ field together with ‘tokenCryptogram’.

The following represents an example of a ‘Sale’ transaction request including ‘tokenCryptogram’ and Network Token value filled out in the ‘cardnumber’ field:

```
<!-- #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="checkoutoption" value="combinedpage">
    <input type="hidden" name="timezone" value="Europe/Berlin"/>
    <input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
    <input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
    <input type="hidden" name="hashExtended" value="<% call createExtendedHash(
"13.00","978" ) %>"/>
    <input type="hidden" name="storename" value="1109950006" />
    <input type="hidden" name="mode" value="payonly"/>
    <input type="hidden" name="paymentMethod" value="M"/>
    <input type="text" name="chargetotal" value="130.00" />
    <input type="hidden" name="currency" value="978"/>
    <input type="hidden" name="authenticateTransaction" value="true"/>
    <input type="hidden" name="threeDSRequestorChallengeIndicator" value=”1”/>
    <input type="hidden" name="tokenCryptogram"
value=”AGX1lvbYlypcAAV22IGgADFA==”/>
    <input type="text" name="cardnumber" value="540215******2355">
    <input type="text" name="expmonth" value="12">
    <input type="text" name="expyear" value="24">
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

30. Appendix XVI – Visa AFT & Mastercard MoneySend

The Account Funding Transaction (AFT) is a transaction used to pull funds from a card account in order to fund a push OCT to a different account, which can be either the cardholder's or another person's account.

Visa AFT

Field Name	M/O	Description, possible values and format
BusinessApplication Identifier	M	Represents the identity of the merchant participating in AFT program, available values: • ACCOUNT_TO_ACCOUNT • BANK_INITIATED_TRANSFER • BUSINESS_TO_BUSINESS • CARD_BILL_PAYMENT • FUNDS_DISBURSEMENT • FUND_TRANSFER • GAMBLING_PAYOUT • GENERAL_FUNDS_DISBURSEMENT • GOVERNMENT_DISBURSEMENT • LOYALTY_PAYMENTS • MERCHANT_DISBURSEMENT • MERCHANT_PAYMENT • NON_CARD_BILL_PAYMENT • ONLINE_GAMBLING_PAYOUT • PAYROLL_OR_PENSION_DISBURSEMENT • PERSON_TO_PERSON • TOPUP_FOR_ENHANCED_PREPAID_LOADS • TOP_OFF • WALLET_TRANSFER
sdrName	M	Sender's Name
sdrAccount	M	Sender's Account Number
sdrReference	O	Sender Reference Number; contains a transaction reference number that is provided by the originator and can be used to uniquely identify the sender
sdrAddr	M	Sender's Address
sdrCity	O	Sender's City
sdrState	O	Sender's State
sdrPhone	O	Sender's Phone Number
sdrCountry	M	Sender's Country
rName	M	Recipient's Name
rAccountNumber	M	Recipient's Account Number
rReferenceNumber	M	Recipient's Reference Number

The following represents an example of a Visa AFT transaction including mandatory and optional set of elements:

```
<!-- #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="checkoutoption" value="combinedpage">
<input type="hidden" name="timezone" value="Europe/Berlin"/>
<input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
<input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
```

```
<input type="hidden" name="hashExtended" value="<% call createExtendedHash(
"13.00","978" ) %>"/>
<input type="hidden" name="storename" value="1109950006" />
<input type="hidden" name="mode" value="payonly"/>
<input type="hidden" name="paymentMethod" value="M"/>
<input type="text" name="chargetotal" value="130.00" />
<input type="hidden" name="currency" value="978"/>
<input type="hidden" name="authenticateTransaction" value="true"/>
<input type="hidden" name="threeDSRequestorChallengeIndicator" value="1"/>
<input type="text" name="cardnumber" value="540215******2355">
<input type="text" name="expmonth" value="12">
<input type="text" name="expyear" value="24">
<input type="text" name="businessApplicationIdentifier"
value="PAYROLL_OR_PENSION_DISBURSEMENT">
<input type="text" name="sdrName" value="Sender Name">
<input type="text" name="sdrAccount" value="1234567890">
<input type="text" name="sdrReference" value="sendRefNo123">
<input type="text" name="sdrAddr" value="Sender Address">
<input type="text" name="sdrCity" value="Sender City">
<input type="text" name="sdrState" value="Sender State">
<input type="text" name="sdrPhone" value="123456456789">
<input type="text" name="sdrCountry" value="Sender Country">
<input type="text" name="rName" value="Recipient Name">
<input type="text" name="rAccountNumber" value="Recipient Account Number">
<input type="text" name="rReferenceNumber" value="receiverReferenceNumber108">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Mastercard MoneySend

Field Name	M/O	Description, possible values and format
TransactionTypeIdentifier	M	Represents the identity of the merchant participating in AFT program, available values: • BUSINESS_DISBURSEMENT_MONEY_SEND • BUSINESS_DISBURSEMENT_MONEY_TRANSFER • BUSINESS_TO_BUSINESS_MONEY_SEND • BUSINESS_TO_BUSINESS_MONEY_TRANSFER • CARD_BILL_PAYMENT_MONEY_SEND • CARD_BILL_PAYMENT_MONEY_TRANSFER • GOVERNMENT_DISBURSEMENT_NONPROFIT • OWN_ACCOUNT_MONEY_SEND • OWN_ACCOUNT_MONEY_TRANSFER • OWN_DEBIT_PREPAID_TRANSFER • OWN_WALLET_TRANSFER • PERSON_TO_PERSON_CARD_ACCOUNT • PERSON_TO_PERSON_MONEY_SEND • PERSON_TO_PERSON_MONEY_TRANSFER • RAPID_MERCHANT_SETTLEMENT
sdrName	M	Sender's Name
sdrAccount	O	Sender's Account Number
sdrReference	O	Sender Reference Number; contains a transaction reference number that is provided by the originator and can be used to uniquely identify the sender
sdrAddr	O	Sender's Address
sdrCity	O	Sender's City
sdrState	O	Sender's State
sdrPhone	O	Sender's Phone Number
sdrCountry	O	Sender's Country
rName	M	Recipient's Name
rAccountNumber	M	Recipient's Account Number

rReferenceNumber	O	Recipient's Reference Number
rCountry	M	Recipient's Country

The following represents an example of a Mastercard funding transaction including mandatory and optional set of elements:

```
<!-- #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="checkoutoption" value="combinedpage">
<input type="hidden" name="timezone" value="Europe/Berlin"/>
<input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
<input type="hidden" name="hash_algorithm" value="HMACSHA256"/>
<input type="hidden" name="hashExtended" value="<% call createExtendedHash(
"13.00","978" ) %>"/>
<input type="hidden" name="storename" value="1109950006" />
<input type="hidden" name="mode" value="payonly"/>
<input type="hidden" name="paymentMethod" value="M"/>
<input type="text" name="chargetotal" value="130.00" />
<input type="hidden" name="currency" value="978"/>
<input type="hidden" name="authenticateTransaction" value="true"/>
<input type="hidden" name="threeDSRequestorChallengeIndicator" value="1"/>
<input type="text" name="cardnumber" value="540215******2355">
<input type="text" name="expmonth" value="12">
<input type="text" name="expyear" value="24">
<input type="text" name="transactionTypeIdentifier" value="
BUSINESS_DISBURSEMENT_MONEY_SEND">
<input type="text" name="sdrName" value="Sender Name">
<input type="text" name="sdrAccount" value="1234567890">
<input type="text" name="sdrReference" value="sendRefNo123">
<input type="text" name="sdrAddr" value="Sender Address">
<input type="text" name="sdrCity" value="Sender City">
<input type="text" name="sdrState" value="Sender State">
<input type="text" name="sdrPhone" value="123456456789">
<input type="text" name="sdrCountry" value="Sender Country">
<input type="text" name="rName" value="receiverReferenceNumber108">
<input type="text" name="rAccountNumber" value="Recipient Account Number">
<input type="text" name="rCountry" value="Recipient Country">
<input type="submit" value="Submit">
</form>
</body>
</html>
```