ePosnet

A Fiserv Global Digital Commerce platform

Connect Integration manual LAS

Argentina and Uruguay

Version: 2020-3

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.

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

2. Payment process options

The Connect solution provides two 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.

In the scenarios where you prefer not to use a hosted form, 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).

Checkout option ‘classic’

The checkout option ‘classic’ splits the payment process into multiple pages where you can easily decide, what kind of information you want to get collected by one of the payment platform´s hosted forms or what you want to collect yourself within your webshop environment.

You can e.g. let customers select their preferred payment method within your webshop and submit that payment method in your request to Connect – or if you should prefer not to send the payment method, the Connect solution will automatically show a payment method selection page to your customer where they can choose from all payment methods that are activated for your store.

With three different modes, you can define the range of data that shall be captured by the payment platform:

- 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 platform displays a third page to also collect shipping information

The most important aspect around the usage of hosted payment pages 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.

The hosted pages can be customized with your own logo, colors, and font types in order to make them fit to the look and feel of your webshop. Please refer to the User Guide Virtual Terminal to learn about how to make such customizations.

Checkout option ‘combinedpage’

The checkout option ‘combinedpage’ 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.

This hosted page also shows your merchant name at the top and allows you to display a summary of the purchased items to your customer.

3. Getting Started

This section provides a simple example on how to integrate your website using the “classic” checkout option in payonly Mode. 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 below).

ASP Example

The following ASP example demonstrates a simple page that will communicate with the payment platform in payonly mode.

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 merchants receipt page. The location of the receipt page can be configured.

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


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


  <form method="post" action=" https://test.ipg-
online.com/connect/gateway/processing ">
        <input type="hidden" name="txntype" value="sale">
        <input type="hidden" name="timezone" value="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="mode" value="payonly"/>
        <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 platform in payonly mode.

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 merchants receipt page. The location of the receipt page can be configured.

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

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

<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">
  <input type="hidden" name="txntype" value="sale">
<input type="hidden" name="timezone" value="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="mode" value="payonly"/>
<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 platform (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**	**PreAuth***	**PostAuth***	**Void**
txntype	'sale', 'preauth', 'postauth' or 'void' (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)
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
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X
hash_algorithm	This is to indicate the algorithm that you use for hash calculation. The possible values are: • HMACSHA256 • HMACSHA384 • HMACSHA512 Only one algorithm value should be used.	X	X	X	X

hashExtended	The extended hash needs to be calculated using all request parameters in ascending order of the parameter names. 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
storename	This is the ID of the store provided by Fiserv.	X	X	X	X
mode	'fullpay', 'payonly' or 'payplus' (the chosen mode for the transaction when using the 'classic' checkout option)	X	X		
chargetotal	This is the total amount of the transaction using a dot or comma as decimal separator, e. g. 12.34 for an amount of 12 Euro and 34 Cent. Group separators like 1,000.01 / 1.000,01 are not allowed.	X	X	X	X
currency	The numeric ISO code of the transaction currency, e. g. 978 for Euro (see examples in Appendix IV)	X	X	X	
oid	The order ID of the initial action a PostAuth shall be initiated for.			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

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'. 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.

checkoutoption	This field allows you to set the checkout option to: • 'classic' for a payment process that is split into multiple pages, • '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.
comments	Place any comments here about the transaction.
customerid	This field allows you to transmit any value, e. g. your ID for the customer.
dynamicMerchantName	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.
invoicenumber	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.
item1 up to item999	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 • 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* 'shipping' always has to be set to '0' for single line item. If you want to include a shipping fee for an order, please use the predefined *id* IPG_SHIPPING. For other fees that you may want to add to the total order, you can use the predefined *id* IPG_HANDLING. When you want to apply a discount, you should include an item with a negative amount and change accordingly the total amount of the order. Do not forget to regard the 'quantity' when calculating the values e.g.: subtotal and VAT since they are fixed by items. Examples: A;Product A;1;5;3;2;0 B;Product B;5;10;7;3;0 C;Product C;2;12;10;2;0

	D;Product D;1;-1.0;-0.9;-0.1;0 IPG_SHIPPING;Shipping costs;1;6;5;1;0 IPG_HANDLING;Transaction fee;1;6.0;6.0;0;0	
language	This parameter can be used to override the default payment page language configured for your merchant store. The following values are currently possible:	
	Language	**Value**
	Chinese (simplified)	zh_CN
	Chinese (traditional)	zh_TW
	Czech	cs_CZ
	Danish	da_DK
	Dutch	nl_NL
	English (USA)	en_US
	English (UK)	en_GB
	Finnish	fi_FI
	French	fr_FR
	German	de_DE
	Greek	el_GR
	Hungarian	hu_HU
	Italian	it_IT
	Japanese	ja_JP
	Norwegian (Bokmål)	nb_NO
	Polish	pl_PL
	Portuguese (Brazil)	pt_BR
	Serbian (Serbia)	sr_RS
	Slovak	sk_SK
	Spanish	es_ES
	Swedish	sv_SE
merchantTransactionId	Allows you to assign a unique ID for the transaction. This ID can be used to reference to this transactions in a PostAuth or Void request (referencedMerchantTransactionId).	
mobileMode	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.	
numberOfInstallments	This parameter allows you to set the number of instalments for a Sale transaction if your customer pays the amount in several parts.	

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 for Direct Debit transactions, a maximum of 78 characters can be submitted to the bank.
paymentMethod	If you let the customer select the payment method (e. g. MasterCard, Visa) 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.
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 in order 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) – mandatory if not setup in Virtual Terminal / Customization.
responseSuccessURL	The URL where you wish to direct customers after a successful transaction (your Thank You URL) – mandatory if not setup in Virtual Terminal / Customization.
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).
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, depending on the mode you choose.

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.

In addition, you should check if JavaScript is activated in your customer's browser and if necessary, inform your customer that JavaScript needs to be activated for the payment process.

payonly Mode

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

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

Field Name	**Description, possible values and format**	**Payment Card**	**Maestro**
cardnumber	Your customer's card number. 12-24 digits.	X	X
expmonth	The expiry month of the card (2 digits)	X	X
expyear	The expiry year of the card (4 digits)	X	X
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"

payplus Mode

Using payplus mode, it is possible to additionally transfer billing information to the payment platform. The following table describes the format of these additional fields:

Field Name	**Possible Values**	**Description**
bcompany	Alphanumeric characters, spaces, and	Customers Company

	dashes limited to 96	
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

fullpay Mode

Using fullpay mode, it is possible to additionally transfer shipping information to the payment platform. 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,	Ship-to Name

	spaces, and dashes limited to 96	
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

Validity checks

Prior to the authorization request for a transaction, the payment platform performs the following validation checks:

- 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)

If the submitted data should not be valid, the payment platform 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.

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 platform 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 payment platform 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"/>`

For Uruguay the key "taxRefundIndicator" must be sent with one of the following values:

Value	Description
NO_TAX_REFUND	No tax refund applies
URY_RETURNS_IVA_LAW_17934	Return of 9% for local debit and credit cards. Return of 22% for foreign debit and credit cards. Applies to the following merchant categories: 12021-12022-12023-12024-12025-12026-12027-12028-12031-12032-12060-12061-31121-32010-32021-32022-32023-32024-32025-32026-32030-32040-32050-33011-33012-33013-33014-43060-43061-43150-43151-43152-43153-43154-43155-43156-43157-43158-43159-51100-51101-4041-82011-82030-82031-82040
URY_RETURNS_IMESI_LAW_18083	Works only on batch. The gateway does not return the tax refund value
URY_RETURNS_AFAM_LAW_18910	Return of 22% for local cards with social benefits only

URY_TAX_REFUND_LAW_18999	Return of 10.5% for foreign debit and credit cards only. Applies only to one merchant category: 74040
URY_RETURNS_IVA_LAW_19210	Merchant should know if the law applies.

If the submitted data does not include this field, the default value "NO_TAX_REFUND" will apply.

8. Data Vault

With the Data Vault product option 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 product option, 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 BIC 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 payment platform 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.

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

9. Recurring Payments

For 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
recurringComments	Limit of 100 characters, including spaces	Any comments about the recurring transaction

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.

10. 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:

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.)
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 note 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. For card payments, the response code '00' is the most common response for an approval.
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

For merchants activated for the MasterCard real-time account updater service:

When your store is enabled for the MasterCard real-time account updater service and you have the payment information vaulted on your side then when applicable the updates are sent as part of the platform 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.
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.
accountUpdaterErrorCode	Error codes that indicate the system/server communication errors.

Server-to-Server Notification

In addition to the response you receive in hidden fields to your ‘responseSuccessURL’ or ‘responseFailURL’, the platform 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 either on port 80 (http) or 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 calculating the hash with the above hash string.

Such notifications can also be set up for Recurring Payments that get automatically triggered by the gateway. Please contact your local support team to get a Shared Secret agreed for these notifications. You can configure your Recurring Transaction Notification URL in the Virtual Terminal (Customisation/ Store URL Settings).

In case of the recurring transactions the hash parameter is calculated as follows:

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

Shared secret (‘`rcpSharedSecret`’) will be used as a key in HMAC to calculating the hash with the above hash string.

Appendix I – How to generate a hash

Transaction request values used for the hash calculation:

- chargetotal= 13.00
- currency= 978
- paymentMethod=M
- responseFailURL=https://localhost:8643/webshop/response_failure.jsp
- responseSuccessURL=https://localhost:8643/webshop/response_success.jsp
- sharedsecret=sharedsecret
- storename=10123456789
- timezone= Europe/Berlin
- transactionNotificationURL=https://localhost:8643/webshop/transactionNotification
- txndatetime=2020:04:17-17:32:41
- txntype=sale

Step 1. Extended hash needs to be calculated using all request parameters in ascending order of the parameter names. Join the parameters' values to one string with pipe separator (use only parameters' values and not the parameters' names).

stringToExtendedHash =
13.00|978|M|https://localhost:8643/webshop/response_failure.jsp|https://localhost:8643/webshop/response_success.jsp|10123456789|Europe/Berlin|https://localhost:8643/webshop/transactionNotification|2020:04:17-17:32:41|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. Use the value returned by the HMACSHA256 algorithm and submit it to our payment gateway in the given form.

```
8CVD62a88mwr/Nfc+t+CWB+XG0g5cqmSrN8JhFlQJVM=
```

<input type="hidden" name="hashExtended" value="
8CVD62a88mwr/Nfc+t+CWB+XG0g5cqmSrN8JhFlQJVM="/>

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:
       - chargetotal
       - currency
       - paymentMethod
       - responseFailURL
       - responseSuccessURL
       - sharedsecret
       - storename
       - timezone
       - transactionNotificationURL
       - txndatetime
       - txntype
     */

    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|currency|paymentMethod|responseFailURL|responseSuccessURL|storename|timezone|tr
ansactionNotificationURL|txndatetime|txntype;

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

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

Appendix III – ipg-util.php

```
<?php
        // Timezeone needs to be set
        date_default_timezone_set('Europe/Berlin');
        $dateTime = date("Y:m:d-H:i:s");

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

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


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


                $hashSHA256 = CryptoJS.HmacSHA384(hashWithAllStrings, sharedSecret);
                $hash = CryptoJS.enc.Base64.stringify($hashSHA256);


                return $hash;
        }
?>
```

Appendix IV – Currency Code List

Currency name	Currency code	Currency number
Afghan Afghani	AFN	971
Albanian	ALL	008
Algerian Dinar	DZD	012
Argentine Peso	ARS	032
Armenian	AMD	051
Aruban Florin	AWG	533
Australian Dollar	AUD	036
Azerbaijani Manat	AZN	944
Bahamian Dollar	BSD	044
Bahrain Dinar	BHD	048
Bangladeshi taka	BDT	050
Barbados Dollar	BBD	052
Belarusian Ruble	BYR	933
Belize Dollar	BZD	084
Bermudian Dollar	BMD	060
Bolivar	VDF	862
Bolivian Boliviano	BOB	068
Bolívar Soberano	VES	928
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	CVE	132
Cayman Islands Dollar	KYD	136
Central African CFA	XAF	950
CFA franc BCEAO	XOF	952
CFP	XPF	953
Chilean Peso	CLP	152

Chinese Renmibi	CNY	156
Colombian Peso	COP	170
Congolose	CDF	976
Costa Rican Colon	CRC	188
Croatian Kuna	HRK	191
Cuban	CUP	192
Czech Koruna	CZK	203
Danish Krone	DKK	208
Djiboutian	DJF	262
Dominican Peso	DOP	214
Dutch Antilles Gulden	ANG	532
East Caribbean Dollar	XCD	951
Eestonia Krona	EEK	233
Egyptian Pound	EGP	818
Ethiopian	ETB	230
European Euro	EUR	978
Fijan Dollar	FJD	242
Gambian	GMD	270
Georgian	GEL	981
Ghanaian	GHS	288
Gibraltar	GIP	292
Guatemalan quetzal	GTQ	320
Guyanese Dollar	GYD	328
Honduran	HNL	340
Hong Kong Dollar	HKD	344
Hungarian Forint	HUF	348
Icelandic 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
Laotian Kip	LAK	418
Lebanese Pound	LBP	422
Libyan	LYD	434
Lithuanian Litas	LTL	440
Macau Pataca	MOP	446
Macedonian	MKD	807
Malagasy	MGA	969
Malawian	MWK	454
Malaysian Ringgit	MYR	458
Maldivian	MVR	462
Mauritian Rupee	MUR	480
Mexican Peso	MXN	484
Moldovan Leu	MDL	498
Mongolian	MNT	496
Moroccan Dirham	MAD	504
Myanmar Kyat	MMK	104
Namibia Dollar	NAD	516
Nepalese Rupee	NPR	524
Netherlands Antillean Guilder	ANG	532
New Zealand Dollar	NZD	554
Ni-Vanuatu	VUV	548
Nicaraguan	NIO	558
Nigerian naira	NGN	566
Norwegian Krone	NOK	578
Omani Rial	OMR	512
Pakistani Rupee	PKR	586
Panamenian	PAB	590
Papua New Guinean	PGK	598
Paraguayan Guarani	PYG	600
Peruvian sol	PEN	604
Philippine Peso	PHP	608
Polish Zloty	PLN	985
Qatari riyal	QAR	634
Romanian New Leu	RON	946

Russian Ruble	RUB	643
Rwandan	RWF	646
Saint Helenian	SHP	654
Salvadoran	SVC	222
Samoan	WST	882
Sao Tomean	STD	678
Saudi Riyal	SAR	682
Serbian Dinar	RSD	941
Seychellois Rupee	SCR	690
Sierra Leonean	SLL	694
Singapore Dollar	SGD	702
Solomon Islander	SBD	090
Somali	SOS	706
South African Rand	ZAR	710
South Korean Won	KRW	410
Sri Lankan Rupee	LKR	144
Surinamese Dollar	SRD	968
Swedish Krona	SEK	752
Swiss Franc	CHF	756
Syrian	SYP	760
Taiwan Dollar	TWD	901
Tajikistani	TJS	972
Tanzanian Shilling	TZS	834
Thai baht	THB	764
Tongan	TOP	776
Trinidad and Tobago Dollar	TTD	780
Tunisian Dollar	TND	788
Turkish Lira	TRY	949
UAE Dirham	AED	784
Uganda Shilling	UGX	800
Ukrainian Hryvnia	UAH	980
US Dollar	USD	840
Uruguayan Peso	UYU	858
Uzbekistani	UZS	860
Yemeni	YER	886
Zambian	ZMW	894

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 platform 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**
American Express	A
Argencard	ARGENCARD
Automatica	AUTOMATICA
BBPS	BBPS
Cabal Argentina	CABAL_ARGENTINA
Cetelem	CETELEM
Clarin 365	CLARIN_365
Club la Nacion	CLUB_LA_NACION
Consumax	CONSUMAX
Coopeplus	COOPEPLUS
Crediguia	CREDIGUIA
Credimas	CREDIMAS
Diners	C
Elebar	ELEBAR
Falabella CMR	FALABELLA_CMR
Favacard	FAVACARD
Grupar	GRUPAR
Italcred	ITALCRED
JCB	J
Kadicard	KADICARD
Maestro	MA
MasterCard	M
Mira	MIRA
Naranja	NARANJA
Nativa	NATIVA
Nevada	NEVADA
Patagonia 365	PATAGONIA365
Pyme Nacion	PYME_NACION

Qida	QIDA
Tarjeta Shopping	TARJETA_SHOPPING
Tarjeta Sol	TARJETA_SOL
Tuya	TUYA
Visa	V

© 2020 Fiserv, Inc. All rights reserved. All trademarks, service marks and trade names used in this material are the property of their respective owners.