EUROPEAN
MERCHANT
SERVICES

# EMS E-COMMERCE GATEWAY HOSTED PAYMENT PAGE

TECHNICAL INSTALLATION MANUAL

FEBRUARY 2016

# CONTENTS

### Getting support

There are different manuals available for the EMS e-Commerce solution. This Integration Guide will be the most helpful for integrating the EMS e-Commerce Gateway checkout in your website.

For information about settings, customization, reports and how to process transactions manually (by keying in the information) please refer to the User Guide EMS e-Commerce Gateway Hosted Payment Page.

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

### Introduction

The EMS e-Commerce Gateway solution provides a quick and easy way to add payment capabilities to your website.

This document describes how to integrate your website using EMS e-Commerce Gateway and provides step by step instructions on how to quickly start accepting payments from your webshop.

Not all functions mentioned in this guide are offered through European Merchant Services at the time of printing this manual. Also, non EMS products can be connected to the EMS e-Commerce Gateway at the moment of printing. For more information please contact EMS Customer Service.

In this guide, you will see the EMS e-Commerce Gateway also being referred to as IPG.

## 2 Payment process options

### 2.1 Hosted payment page or using your own payment form

The EMS e-Commerce Gateway solution basically provides two options for integration with your website:

- With the easiest option you use ready-made form pages for the payment process that we provide and host on our servers. In this case your customer will be forwarded to EMS when it comes to payment and can enter the sensitive cardholder data on our SSL-encrypted page.This option facilitates compliance with the Data Security Standard of the Payment Card Industry (PCI DSS) as the payment processing is completely hosted by EMS. Afterwards the customer will be redirected to your shop again. Your shop system will be notified abou the payment result.

If you prefer your customer never to leave your website, you can create your own payment forms in your individual corporate design. Please note that if you store or process 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). To display a secured website (lock symbol in the browser) to your customer, your website needs to provide a SSL-connection via a HTTPS-Server.

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

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

## 2.2 PayOnly Mode

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

## 2.3 PayPlus Mode

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

## 2.4 FullPay Mode

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

## 3 Getting Started

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

### 3.1 Checklist

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

- Store Name

This is the PSPID or USERID of the store that was given to you by EMS.
For example : 10123456789

- Shared Secret

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

### 3.2 ASP Example

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

When the cardholder clicks Submit, they are redirected to the EMS secure pages, where they can enter their billing, shipping and payment information. After payment has been completed, the user will be redirected to the merchants receipt page. The location of the receipt page can be configured.

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


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


 <form method=”post” action=” https://test.ipg-online.com/connect/
gateway/processing “>
       <input type=”hidden” name=”txntype” value=”sale”>
       <input type=”hidden” name=”timezone” value=”Europe/Berlin”/>
       <input type=”hidden” name=”txndatetime” value=”<% getDateTime()
       %>”/>
       <input type=”hidden” name=”hash_algorithm” value=”SHA256”/>
       <input type=”hidden” name=”hash” value=”<% call createHash(
       “13.00”,”978” ) %>”/>
       <input type=”hidden” name=”storename” value=”10123456789” />
       <input type=”hidden” name=”mode” value=”fullpay”/>
       <input type=”text” name=”chargetotal” value=”13.00” />
       <input type=”hidden” name=”currency” value=”978”/>
       <input type=”submit” value=”Submit”>
 </form>
 </body>
</html>
```

The code presented in Appendix I represents the included file ipg-util.asp. It includes code for generating a SHA-256 hash as is required by EMS. 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 EMS 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 SHA-256 hash. This file can be provided on request. To prevent fraudulent transactions, it is recommended that the ‘hash’ is calculated within your server and JavaScript is not used like shown in the samples mentioned.

### 3.3 PHP Example

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

When the cardholder clicks Submit, they are redirected to the EMS secure pages, where they can enter their shipping, billing and payment information. After payment has been completed, the user will be redirected to the merchants receipt page. The location of the receipt page can be configured.

```
<? include(“ipg-util.php”); ?>


<html>

<head><title>IPG Connect Sample for PHP</title></head>

 <body>

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


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

 <input type=”hidden” name=”txntype” value=”sale”>

<input type=”hidden” name=”timezone” value=”Europe/Berlin”/> <input
type=”hidden” name=”txndatetime” value=”<?php echo getDateTime() ?>”/>

 <input type=”hidden” name=”hash_algorithm” value=”SHA256”/>


<input type=”hidden” name=”hash” value=”<?php echo createHash(
“13.00”,”978” ) ?>”/>

 <input type=”hidden” name=”storename” value=”10123456789”/>

<input type=”hidden” name=”mode” value=”fullpay”/>

<input type=”text” name=”chargetotal” value=”13.00”/>
```

```
<input type=”hidden” name=”currency” value=”978”/>


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

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

The code presented in Appendix II represents the included file ipg-util.php. It includes code for generating a SHA-256 hash as is required by EMS. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

### 3.4 Amounts for test transactions

When using our test system for integration, odd amounts (e. g. 13.01 EUR or 13.99 EUR) can cause the transaction to decline as these amounts are sometimes used to simulate unsuccessful authorisations.

We therefore recommend 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.

| Field name | Description, possible values and format | „Sale“ transaction | PreAuth* | PostAuth* | Void |
|---|---|---|---|---|---|
| txntype | 'sale’, 'preauth’, 'postauth’ or 'void’<br>(the transaction type – please note the descriptions of transaction types in the User Guide Virtual Terminal & Manager) The possibility to send a ‘void’ using the IPG interface is restricted. Please contact your local support team if you want to enable this feature. | X<br>(sale) | X<br>(preauth) | X<br>(postauth) | X<br>(void) |
| timezone | Timezone of the transaction in Area/ Location format, e.g.<br>Africa/Johannesburg<br>America/New_York<br>America/Sao_Paulo<br>Asia/Calcutta<br>Australia/Sydney<br>Europe/Amsterdam<br>Europe/Berlin<br>Europe/Dublin<br>Europe/London<br>Europe/Rome | X | X | X | X |
| txndatetime | YYYY:MM:DD-hh:mm:ss<br>(exact time of the transaction) | X | X | X | X |
| hash_algorithm | This is to indicate the algorithm that you use for hash calculation. The only possible value at this point is SHA-256. | X | X | X | X |
| hash | This is a SHA hash of the following fields: storename + txndatetime + chargetotal + currency + sharedsecret. Note, that it is important to have the hash generated in this exact order. An example of how to generate a SHA-256 hash is given in Appendix I. | X | X | X | X |

| | | | | | |
|---|---|---|---|---|---|
| storename | This is the PSPID or USERID of the store provided by EMS. | X | X | X | X |
| mode | ‘fullpay’, ‘payonly’ or ‘payplus’<br>(the chosen mode for the transaction) | X | X | | |
| chargetotal | This is the total amount of the transaction using a dot or comma as decimal separator, e. g. 12.34 for an amount of 12 Euro and 34 Cent. Group separators like1,000.01 / 1.000,01 are not allowed. | X | X | X | X |
| currency | The numeric ISO code of the transaction currency, e. g. 978 for Euro (see examples below) | X | X | X | |
| oid | The order ID of the initial action a PostAuth or Void shall be initiated for | | | X | X |
| tdate | Exact identification of a transaction that shall be voided. You receive this value as result parameter ‚tdate’ of the corresponding transaction. | | | | X |

* The transaction types ‘preauth’ and ‘postauth’ only apply to the payment methods credit card, PayPal, ClickandBuy andKlarna.

### Currency code list:

| Currency name | Currency code | Currency number |
|---|---|---|
| Brazilian Real | BRL | 986 |
| Euro | EUR | 978 |
| Indian Rupee | INR | 356 |
| Pound Sterling | GBP | 826 |
| US Dollar | USD | 840 |
| South African Rand | ZAR | 710 |
| Swiss Franc | CHF | 756 |
| Australian Dollar | AUD | 036 |
| Bahrain Dinar | BHD | 048 |
| Canadian Dollar | CAD | 124 |
| Chinese Renmibi | CNY | 156 |
| Croatian Kuna | HRK | 191 |
| Czech Koruna | CZK | 203 |
| Danish Krone | DKK | 208 |
| Hong Kong Dollar | HKD | 344 |

| | | |
|---|---|---|
| Hungarian Forint | HUF | 348 |
| Israeli New Shekel | ISL | 376 |
| Japanese Yen | JPY | 392 |
| Kuwaiti Dinar | KWD | 414 |
| Lithuanian Litas | LTL | 440 |
| Mexican Peso | MXN | 484 |
| New Zealand Dollar | NZD | 554 |
| Norwegian Krone | NOK | 578 |
| Polish Zloty | PLN | 985 |
| Romanian New Leu | RON | 946 |
| Saudi Rihal | SAR | 682 |
| Singapore Dollar | SGD | 702 |
| South Korean Won | KRW | 410 |
| Swedish Krona | SEK | 752 |
| Turkish Lira | TRY | 949 |
| UAE Dirham | AED | 784 |

## 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’. |
| comments | Place any comments here about the transaction. |
| customerid | This field allows you to transmit any value, e. g. your ID for the customer. |
| dccInquiryId | 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. |
| 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. |

| | |
|---|---|
| hashExtended | The extended hash is an optional security feature that allows you to include all parameters of the transaction request. It needs to be calculated using all request parameters in ascending order of the parameter names. |
| item1 *up to* item999 | The 'item1' to 'item999' parameters allow you to send basket information in the following format:<br>*id;description;quantity;item_total_price;sub_total;vat_tax;shipping*<br>'shipping' always has to be set to '0' for single line items. If you want to include a shipping fee for an order, please use the predefined *id* IPG_SHIPPING.<br>For other fees that you may want to add to the total order, you can use the predefined *id* IPG_HANDLING.<br>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:<br>A;Product A;1;5;3;2;0<br>B;Product B;5;10;7;3;0<br>C;Product C;2;12;10;2;0<br>D;Product D;1;-1.0;-0.9;-0.1;0<br>IPG_SHIPPING;Shipping costs;1;6;5;1;0<br>IPG_HANDLING;Transaction fee;1;6.0;6.0;0;0 |

<table>
<tr><td rowspan="13">language</td><td colspan="2">This parameter can be used to override the default payment page language configured for your merchant store.<br>The following values are currently possible:</td></tr>
<tr><td>**Language**</td><td>**Value**</td></tr>
<tr><td>Chinese (simplified)</td><td>zh_CN</td></tr>
<tr><td>Chinese (traditional)</td><td>zh_TW</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>Italian</td><td>it_IT</td></tr>
<tr><td>Portuguese (Brazil)</td><td>pt_BR</td></tr>
<tr><td>Spanish</td><td>es_ES</td></tr>
<tr><td>mandateReference</td><td colspan="2">This field allows you to transmit a Mandate Reference for Direct Debit payments</td></tr>
<tr><td>mandateType</td><td colspan="2">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 or to ‘recurringCollection’ when using a mandate for recurring Direct Debit collections. Transactions where this parameter is not submitted by the merchant will be flagged as a single debit collection.</td></tr>
<tr><td>mobileMode</td><td colspan="2">If your customer uses a mobile device for shopping at your online store you can submit this parameter with the value ‘true’. This will lead your customer to a payment page flow that has been specifically designed for mobile devices.</td></tr>
<tr><td>numberOfInstallments</td><td colspan="2">This parameter allows you to set the number of instalments for a Sale transaction if your customer pays the amount in several parts.</td></tr>
<tr><td>oid</td><td colspan="2">This field allows you to assign a unique ID for your order. If you choose not to assign an order ID, the EMS system will automatically generate one for you.</td></tr>
</table>

<table>
<tr><td rowspan="20">paymentMethod</td><td colspan="2">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.<br>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.<br>Valid values are:</td></tr>
<tr><td><b>Payment Method</b></td><td><b>Value</b></td></tr>
<tr><td>MasterCard</td><td>M</td></tr>
<tr><td>Visa (Credit/Debit/Electron/Delta)</td><td>V</td></tr>
<tr><td>American Express</td><td>A</td></tr>
<tr><td>Diners</td><td>C</td></tr>
<tr><td>JCB</td><td>J</td></tr>
<tr><td>Direct Debit Germany</td><td>debitDE</td></tr>
<tr><td>Cabal</td><td>CA</td></tr>
<tr><td>giropay</td><td>giropay</td></tr>
<tr><td>Klarna</td><td>klarna</td></tr>
<tr><td>Maestro</td><td>MA</td></tr>
<tr><td>Maestro UK</td><td>maestroUK</td></tr>
<tr><td>PayPal</td><td>paypal</td></tr>
<tr><td>RuPay</td><td>RU</td></tr>
<tr><td>SOFORT Banking (Überweisung)</td><td>sofort</td></tr>
<tr><td>Sorocred</td><td>SO</td></tr>
<tr><td>iDEAL</td><td>ideal</td></tr>
<tr><td>ClickandBuy</td><td>clickAndBuy</td></tr>
<tr><td>DirektÜberweisung / Direkt.Ident</td><td>direkt</td></tr>
<tr><td>refer</td><td colspan="2">This field describes who referred the customer to your store.</td></tr>
<tr><td>responseFailURL</td><td colspan="2">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.</td></tr>
<tr><td>responseSuccessURL</td><td colspan="2">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.</td></tr>
<tr><td>shipping</td><td colspan="2">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’.</td></tr>
</table>

<table>
<tr><td>trxOrigin</td><td colspan="2">This parameter allows you to use the secure and hosted payment form capabilities within your own application for Mail/Telephone Order (MOTO) payments. Possible values are ‘MOTO‘ (for transactions where you have received the order over the phone or by mail and enter the payment details yourself) and ‘ECI‘ (for standard usage in an eCommerce environment where your customer enters the payment details).</td></tr>
<tr><td>vattax</td><td colspan="2">This field allows you to submit an amount for Value Added Tax or other taxes. Please ensure the sub total amount plus shipping plus tax equals the charge total.</td></tr>
<tr><td rowspan="12">idealIssuerID</td><td colspan="2">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.</td></tr>
<tr><td><b>iDEAL issuer</b></td><td><b>Value</b></td></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</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>
</table>

## 6 Using your own forms to capture the data

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

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.

## 6.1 PayOnly Mode

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

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

| Field name | Description, possible values and format | Credit Card (+ Visa Debit/Electron/ | German Direct Debit | Maestro | giropay | PayPal, SOFORT, ClickandBuy, DirektÜberweisung. iDEAL, Klarna | Maestro UK |
|---|---|---|---|---|---|---|---|
| cardnumber | Your customer's card number. 12-24 digits. | X | | X | | | X |
| expmonth | The expiry month of the card (2 digits) | X | | X | | | X |
| expyear | The expiry year of the card (4 digits) | X | | X | | | X |
| cvm | The card code, in most cases on the backside of the card (3 to 4 digits) | X | | X<br>as an optional field "if on card" | | | (X) |
| iban | Your customer's IBAN - International Bank Account Number (22 digits) | | X | | | | |
| bic | Your customer's BIC – Business Identifier Code (8 or 11 digits) | | | | X | | |
| issuenumber | UK Maestro / Solo card's issue number (1 to 2 digits) | | | | | | (X)<br>mandatory if cvm not set |

## 6.2 PayPlus Mode

Using PayPlus mode, 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 | Customers Company |
| bname | Alphanumeric characters, spaces, and dashes | Customers Name |
| baddr1 | Limit of 30 characters, including spaces | Customers Billing Address 1 |
| baddr2 | Limit of 30 characters, including spaces | Customers Billing Address 2 |
| bcity | Limit of 30 characters, including spaces | Billing City |
| bstate | Limit of 30 characters, including spaces | State, Province or Territory |
| bcountry | 2 Letter Country Code | Country of Billing Address |
| bzip | International Postal Code | Zip or Postal Code |
| phone | Limit of 20 Characters | Customers Phone Number |
| fax | Limit of 20 Characters | Customers Fax Number |
| email | Limit of 45 Characters | Customers Email Address |

### 6.3 FullPay Mode

Using FullPay mode, it is possible to additionally transfer shipping information to the payment gateway. The 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 | Ship-to Name |
| saddr1 | Limit of 30 characters, including spaces | Shipping Address Line 1 |
| saddr2 | Limit of 30 characters, including spaces | Shipping Address Line 2 |
| scity | Limit of 30 characters, including spaces | Shipping City |
| sstate | Limit of 30 characters, including spaces | State, Province or Territory |
| scountry | 2 letter country code | Country of Shipping Address |
| szip | International Postal Code | Zip or Postal Code |

## 6.4 Validity checks

Prior to the authorisation request for a transaction, the payment gateway 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)
- An IBAN must contain 22 digits
- A BIC needs to contain 8 or 11 digits

If the submitted data should not be valid, the payment gateway presents a corresponding error 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).

Note, if you implement the payment method Klarna in (Full-)ByPass mode, you will need to follow certain rules for allowing item handling in your request. For more information please refer to Appendix V.

# 7 Additional Custom Fields

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

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

## 8 3D Secure

The EMS e-Commerce Gateway solution includes the ability to authenticate transactions using Verified by Visa, MasterCard SecureCode and American Express SafeKey. If your credit card agreement includes 3D Secure and your Merchant ID has been activated to use this service, you do not need to modify your payment page.

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

Example for a transaction without 3D Secure:

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

In principle, it may occur that 3D Secure authentications 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 (GICC ECI 7). **A liability shift to the card issuer for possible chargebacks is not warranted in this case.** If you prefer that such transactions shall not be processed at all, our technical support team can block them for your store on request.

Credit card transactions with 3D Secure hold in a pending status while cardholders search for their password or need to activate their card for 3D 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 3D 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 3D Secure transactions differs in some points compared to a normal transaction flow. If you already have an existing shop integration and plan to activate 3D Secure subsequently, we recommend performing some test transactions on our test environment.

## 9 MCC 6012 Visa Mandate

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

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

| Field Name | Description |
|---|---|
| mcc6012BirthDay | Date of birth in format MM/DD/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 |

## 10 Data Vault

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

If you have ordered this product, the EMS e-Commerce Gateway solution offers you the following functions:

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

- **Initiate payment transactions using stored data**
  If you stored cardholder information using the Data Vault 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 EMS'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.

## 11 Solvency Information from Bürgel

The EMS e-Commerce Gateway solution is integrated with Bürgel Wirtschaftsinformationen, a leading company in the field of business information.

This integration allows you to select the payment methods you offer to an individual customer based on Bürgel's information on the non-payment risk. Please see information on setting options in the User Guide Virtual Terminal & Online Portal.

If you have a contract with Bürgel and have ordered this product option, use the following parameters for your transaction requests:

| Field Name | Description | Mandatory |
|---|---|---|
| valueaddedservices | Buergel | Please submit this parameter for all transactions where you want to use this feature |
| bfirstname, blastname, bname | Customer name | Yes, bfirstname and blastname or bname |
| baddr1 | Customer address | Yes, format must be street and house number |
| bzip | Customer ZIP or Postal Code | Yes |
| bcity | Customer city | Yes |
| bcountry | Customer country | Yes, in the ISO alpha code format, e.g. DE |
| bbirthday | Customer birthday | Not mandatory.<br>Format: DD.MM.YYYY |

## 12 Recurring Payments

For credit card, Direct Debit and PayPal transactions, it is possible to install recurring payments using EMS e-Commerce Gateway. 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<br>week<br>month<br>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 |

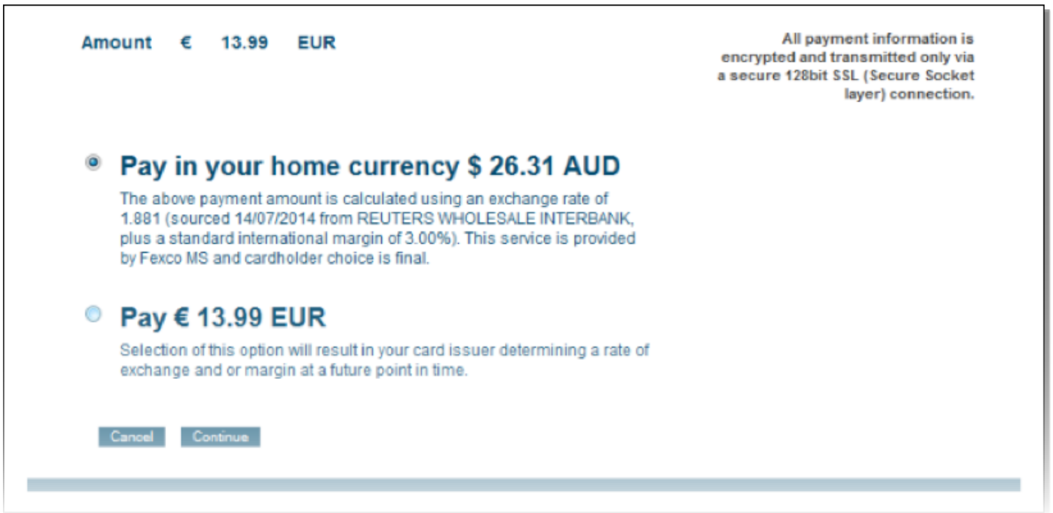
Note that the start date of the recurring payments will be the current date and will be automatically calculated by the system.

The recurring payments installed using EMS e-Commerce Gateway can be modified or cancelled using the Virtual Terminal or Web Service API.

## 13 EMS e-DCC and Dynamic Pricing

With EMS e-DCC,  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 EMS e-Commerce Gateway 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.



Amount € 13.99 EUR

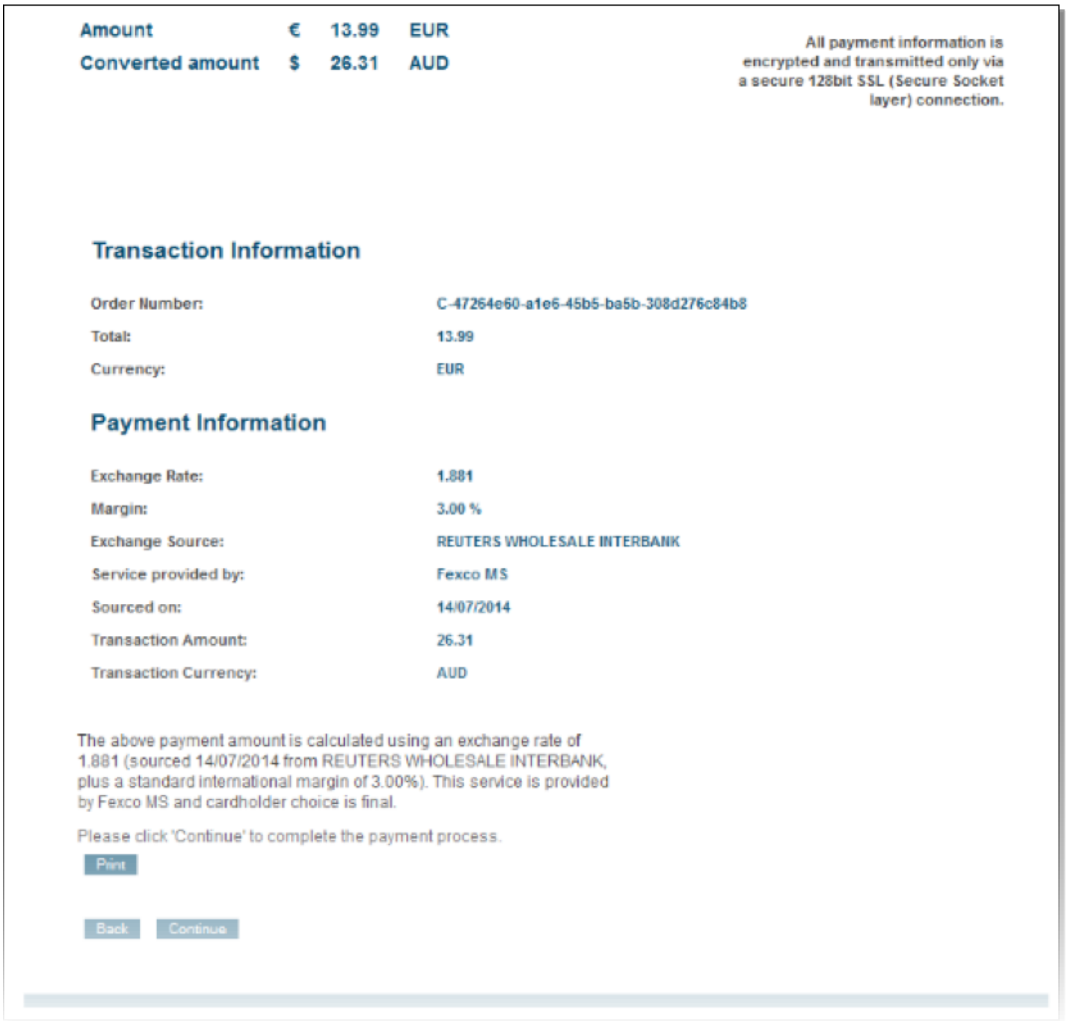
All payment information is encrypted and transmitted only via a secure 128bit SSL (Secure Socket layer) connection.

● Pay in your home currency $ 26.31 AUD

The above payment amount is calculated using an exchange rate of 1.881 (sourced 14/07/2014 from REUTERS WHOLESALE INTERBANK, plus a standard international margin of 3.00%). This service is provided by Fexco MS and cardholder choice is final.

○ Pay € 13.99 EUR

Selection of this option will result in your card issuer determining a rate of exchange and or margin at a future point in time.



| | | | |
|---|---|---|---|
| Amount | € | 13.99 | EUR |
| Converted amount | $ | 26.31 | AUD |

All payment information is encrypted and transmitted only via a secure 128bit SSL (Secure Socket layer) connection.

## Transaction Information

| | |
|---|---|
| Order Number: | C-47264e60-a1e6-45b5-ba5b-308d276c84b8 |
| Total: | 13.99 |
| Currency: | EUR |

## Payment Information

| | |
|---|---|
| Exchange Rate: | 1.881 |
| Margin: | 3.00 % |
| Exchange Source: | REUTERS WHOLESALE INTERBANK |
| Service provided by: | Fexco MS |
| Sourced on: | 14/07/2014 |
| Transaction Amount: | 26.31 |
| Transaction Currency: | AUD |

The above payment amount is calculated using an exchange rate of 1.881 (sourced 14/07/2014 from REUTERS WHOLESALE INTERBANK, plus a standard international margin of 3.00%). This service is provided by Fexco MS and cardholder choice is final.

Please click 'Continue' to complete the payment process.

Please note that for compliance reasons EMS e-DCC 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 EMS e-Commerce Gateway solution, you need to send the DCC Inquiry ID that you have received along with the exchange rate request in the parameter 'dccInquiryId'.

## 14 Transaction Response

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

| Field name | Description |
|---|---|
| approval_code | Approval code for the transaction. The first character of this parameter is the most helpful indicator for verification of the transaction result.<br>‘Y’ indicates that the transaction has been successful<br>‘N’ indicates that the transaction has not been successful<br>“?” indicates that the transaction has been successfully initialised, but a final result is not yet available since the transaction is now in a waiting. The transaction will be updated at a later stage. |
| oid | Order ID |
| refnumber | Reference number |
| status | Transaction status |
| txndate_processed | Time of transaction processing |
| tdate | Identification for the specific transaction, e. g. to be used for a Void |
| fail_reason | Reason the transaction failed |
| response_hash | Hash-Value to protect the communication (see note below) |
| processor_response_code | The response code provided by the backend system.<br>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 succesful 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.)<br>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:<br>MC<br>VISA<br>AMEX<br>DINERS/DISCOVER<br>JCB<br>UNIONPAY<br>MAESTRO<br>Filled with “N/A” for any payment method which is not a credit card or debit card |

## For 3D Secure transactions only:

| | |
|---|---|
| response_code_3dsecure | Return code indicating the classification of the transaction:<br>**1** – Successful authentication (GICC ECI 11/10)<br>**2** – Successful authentication without AVV (GICC ECI 11/10)<br>**3** – Authentication failed / incorrect password (transaction declined)<br>**4** – Authentication attempt (GICC ECI 13/12)<br>**5** – Unable to authenticate / Directory Server not responding (GICC ECI 7)<br>**6** – Unable to authenticate / Access Control Server not responding (GICC ECI 7)<br>**7** – Cardholder not enrolled for 3D Secure (GICC ECI 13/12)<br>**8** – Invalid 3D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS)<br>Please see note about blocking GICC ECI 7 transactions in the 3D Secure section of this document. |

## For e-DCC 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). |

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.<br>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<br>**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.

The parameter 'response_hash' allows you to recheck if the received transaction response has really been sent by EMS and can therefore protect you from fraudulent manipulations. The value is created with a SHA Hash using the following parameter string:

```
sharedsecret + approval_code  + chargetotal + currency + txndatetime +
storename
```

The hash algorithm is the same as the one that you have set in the transaction request.

Please note that if you want to use this feature, you have to store the 'txndatetime' that you have submitted with the transaction request in order to be able to validate the response hash.

In addition, it is possible that the payment gateway sends the above result parameters to a defined URL. 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:

- 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 + sharedsecret + currency + txndatetime + storename
+ approval_code.
```

## 15 Appendix I – How to generate a SHA-256 Hash

1. Concatenate the storename, txndatetime, chargetotal, currency and sharedsecret

2. Convert each character of the concatenated value to its ascii hexadecimal representation

3. Pass the ascii hexadecimal representation of the concatenated string to the SHA-256 algorithm

4. Use the value returned by the SHA-256 algorithm in the form that is submitted to our payment gateway

## Example

- storename = 98765432101
- txndatetime = 2013:07:16-09:57:08
- chargetotal = 1.00
- currency = 826
- sharedsecret = TopSecret

Step 1. Concatenate the storename, txndatetime, chargetotal, currency and sharedsecret

987654321012013:07:16-09:57:081.00826TopSecret

Step 2. Convert each character of the concatenated value to its ascii hexadecimal representation

3938373635343332313031323031333a30373a31362d30393a35373a3038312e3030
383236546f70536563726574

Step 3. Pass the ascii hexadecimal representation of the concatenated string to the SHA-256 algorithm

SHA256(3938373635343332313031323031333a30373a31362d30393a35373a3038312e303038323654
6f70536563726574)

Step 4. Use the value returned by the SHA-256 algorithm in the form that is submitted to our payment gateway

fe9450b8dea3b9b2b211674d68485699df557efeb7afef47bea829089d875a36

<input type="hidden" name="hash" value="
fe9450b8dea3b9b2b211674d68485699df557efeb7afef47bea829089d875a36
"/>

## 16 Appendix II – ipg-util.asp

```
<Script LANGUAGE=JScript RUNAT=Server src="sha256.js">
</SCRIPT>
<Script LANGUAGE=JScript RUNAT=Server>
      var today = new Date();
      var formattedDate = today.formatDate("Y:m:d-H:i:s");

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

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

            var ascii = getHexFromChars(stringToHash);

            var hash = calcSHA256(ascii);

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

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

## 17 Appendix III – ipg-util.php

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

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

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

            $stringToHash = $storename . getDateTime() . $chargetotal .
$currency . $sharedSecret;
```

```
            $ascii = bin2hex($stringToHash);

            return hash('sha256',$ascii);
        }

?>
```

## 18 Appendix IV – PayPal

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

### Transaction types mapping

| IPG<br>Transaction Type (txntype) | PayPal operation |
|---|---|
| sale | SetExpressCheckoutPayment<br>(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 IPG (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 IPG 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 IPG (such overwriting depends on the above logic).

## 19 Appendix V – Klarna

Refer to the following information when integrating Klarna Invoice and Part Pay as payment methods.

### Prepare your website

You can choose to integrate your website in such a way that your customers will be redirected to hosted payment forms for the Klarna checkout process. Using this integration method, the EMS e-Commerce Gateway solution is providing all required input forms for you.

For Klarna, all of EMS e-Commerce Gateway's payment modes (PayOnly, PayPlus, FullPay) behave in the same way. Since Klarna does not allow the shipping address to be different from the billing address, no separate entry form for a shipping address will be displayed to your customer. Instead, Klarna's specific billing form (customer details form) will be shown to your customers in order to capture their details.

If you prefer your customers never to leave your website, you can create your own specific payment forms for Klarna by modifying your website accordingly to the guidelines presented on Klarna's website (http://developers.klarna.com/en). In that case you will be able to collect all data required for the transaction on your side and send it over to the payment gateway as a part of the transaction request. You could also decide to send a subset of the data needed for the transaction. In that case the IPG solution will only display selected hosted forms to your customers in order to collect the mandatory data that has not been submitted by you.

### Activate Klarna for your test store

- Obtain test credentials from Klarna via https://developers.klarna.com/en/se+php/kpm/apply-for-test-account
- Make sure your payment gateway test Store ID has been enabled for Klarna
- Activate Klarna as a payment method in your test store, via the Klarna Setup page in the Virtual Terminal's Customisation section.

### Order Process with Klarna

The process begins with the customer selecting the goods in your web shop and placing the order. To allow your customers to pay by Klarna you have to submit a PreAuth transaction, which is used to create the order. If it is more suitable for your processes, you can alternatively use the Sale transaction type which will then be automatically translated into a PreAuth transaction. Klarna PreAuth transaction requires a number of mandatory and additional parameters which are described in more detail below.

When the order is submitted, Klarna will run fraud and credit checks on the consumer and tell in return if the purchase is approved. Once the purchase is approved, you should start preparing the goods for shipment.

When the goods are ready for shipment, a Completion needs to be submitted by you to the gateway since it activates the order on Klarna side and allows you to provide the customer with the invoice.

**Required parameters for Klarna PreAuth transactions**

**Basket information (Line items)** - Klarna requires the list of items in order to approve the purchase. Therefore it is mandatory to send line items with all PreAuth transactions.
The basket information has to be sent in request parameters: ‘item1’, ‘item2’ up to ‘item999’, in the following format: id;description;quantity;item_total_price;sub_total;vat_tax;shipping.
Transactions without line items will be declined.

**Customer information details** - Depending on the country selection (please refer to http://developers.klarna.com/en), Klarna requires different consumer information details in order to approve the purchase such as e.g.:

- klarnaPersonalNumber
- klarnaBirthDate
- klarnaClientGender
- klarnaFirstname
- klarnaLastname
- klarnaStreetName
- klarnaHouseNumber
- klarnaHouseNumberExtension
- klarnaCellPhoneNumber
- klarnaPClassID
- klarnaCity
- klarnaCountry
- klarnaZip
- klarnaPhone
- klarnaEmail

Taking into account the country selection as well as the chosen integration mode, you can send all customer information details in your transaction request or only a subset of it.

If you send billing information (general parameters starting with "b" e.g.: bname, bcity, bcountry, etc.), the Klarna customer details form, displayed to the customer, will already be  prefilled with the details based on the following fields:

- bcity -> klarnaCity
- bcountry -> klarnaCountry
- bzip -> klarnaZip
- phone -> klarnaPhone
- email -> klarnaEmail

**Currency** – The currency of a PreAuth transaction for Klarna has to correspond to the currency of the customer’s country (the buyer’s country). You should submit currency in the parameter 'currency' as a numeric ISO code. See examples below.

| Country name | Currency name | Currency code | Currency number |
|---|---|---|---|
| Austria<br>Germany<br>Netherlands<br>Norway | Euro | EUR | 978 |
| Denmark | Danish Krone | DKK | 208 |
| Norway | Norwegian Krone | NOK | 578 |
| Sweden | Swedish Krona | SEK | 752 |

**Additional data** - As part of the order, you may provide additional data such as:

- Shipping fee. This data is not mandatory for Klarna. There are several options to specify it:

  1. The preferred option is to include the shipping fee to the parameter ‘shipping’, in the same format as ‘chargetotal’. ‘subtotal’ and ‘vaxtax’ have to be submitted as well in this case. Note that the ‘chargetotal’ has to be equal to ‘subtotal’ plus ‘shipping’ plus ‘vattax’. At the same time, the sum of the ‘item_total_price’ has to be equal to ‘chargetotal’ without ‘shipping’. Do not forget to regard the ‘quantity’ when calculating the values.

See examples below:

*chargetotal=89.00*
*subtotal=58.00*

*shipping=10.00*

*vattax=21.00*

*A;Product A;1;5.0;3.0;2.0;0*

*B;Product B;5;10.0;7.0;3.0;0*

*C;Product C;2;12.0;10.0;2.0;0*

2. The second option is to include the shipping fee as a separate line item with a specific id: ‘IPG_SHIPPING’. Note that the sum of the ‘item_total_price’ has to be equal to the ‘chargetotal’. Do not forget to regard the ‘quantity’ when calculating the values. See examples below:

*chargetotal=85.00*

*subtotal=63.00*

*shipping=0.0*

*vattax=22.00*

*A;Product A;1;5.0;3.0;2.0;0*

*B;Product B;5;10.0;7.0;3.0;0*

*C;Product C;2;12.0;10.0;2.0;0*

*IPG_SHIPPING;Shipping costs;1;6.0;5.0;1.0;0*

- Discounts. This data is not mandatory for Klarna.
  You can define the discount by submitting the item with a negative amount. See examples below:

  *chargetotal=84.00*

  *subtotal=62.10*

  *shipping=0.0*

  *vattax=21.90*

  *A;Product A;1;5.0;3.0;2.0;0*

  *B;Product B;5;10.0;7.0;3.0;0*

  *C;Product C;2;12.0;10.0;2.0;0*

  *D;Basic clipboard;1;-1;-0.9;-0.1;0*

  *IPG_SHIPPING;Shipping costs;1;6.0;5.0;1.0;0*

- Handling fee (Charge fee=Invoice fee).
  This data is required only for Klarna Invoice and is set to '0' by default it. You can however decide to set the invoice fee to a different extend, via the Klarna Setup page in the Virtual Terminal's Customisation section.

  Note that when a customer will select Invoice as a payment method in the normal IPG flow, the handling fee is automatically added to the order as an additional item with the specific id: 'IPG_HANDLING'.

  In case you use (Full)ByPass mode you have to follow certain rules to allow the handling item in your request by submitting:

  - an item with id: 'IPG_HANDLING'
    Example: IPG_HANDLING;Transaction fee;1;6.0;6.0;0;0
  - klarnaPClassID = -1
  - klarnaCountry filled in

All other transactions containing 'IPG_HANDLING' item will be declined.

### Transaction types mapping

| IPG<br>Transaction Type (txntype) | Klarna operation |
|---|---|
| sale | Not applicable / will be translated into preauth |
| preauth | reserveAmount |
| postauth *(full and partial)* | Activate |
| void | Cancel_Reservation<br>Credit_Invoice |

The basket information is required for all PreAuth transactions. PreAuth transactions without items will be declined. The full PostAuth transaction can be submitted without items but the order amount has to be the same as the remaining value of the original transaction.
Void can be used for PreAuth and PostAuth transactions with the restriction that PostAuth transactions can only be voided during the same day.

Note that a PreAuth transaction for Klarna could return the transaction result: WAITING. This status indicates that the transaction is being reviewed by Klarna and will be updated at a later point.

**Do you have any questions about our payment services?**

Call us at +31 (0)20 - 660 30 40

E-mail us at contact@emscard.com

Or visit our website at www.emscard.com

**Follow us on Twitter and LinkedIn**

twitter.com/emscard

in www.linkedin.com/company/european-merchant-services

European Merchant Services BV - Postbus 22764 - 1100 DG Amsterdam Nederland - KvK: 34226533 - Telefoon: +31 (0)20 660 31 20