

authipay

secure payment processing



Connect

Integration Guide V 1.0

Contents

1	Introduction	3
2	Payment process options	3
2.1	Hosted payment page or using your own payment form	3
2.2	PayOnly Mode	4
2.3	PayPlus Mode	4
2.4	FullPay Mode	4
3	Getting Started	4
3.1	Checklist	4
3.2	ASP Example	5
3.3	PHP Example	6
3.4	Amounts for test transactions	6
4	Mandatory Fields	7
5	Optional Form Fields	9
6	Using your own forms to capture the data	11
6.1	PayOnly Mode	11
6.2	PayPlus Mode	12
6.3	FullPay Mode	12
6.4	Validity checks	13
7	Additional Custom Fields	13
8	3D Secure	14
9	Data Vault	14
10	Recurring Payments	15
11	Transaction Response	16
	Appendix I	19
	Appendix II	20

Getting Support

There are different manuals available for the Authipay gateway. This Integration Guide will be the most helpful for integrating the Connect solution.

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

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 simple way for connecting an online store to Authipay.

Connect manages all of your interactions with credit card processors and financial institutions.

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 web shop.

2 Payment process options

2.1 Hosted payment page or using your own payment form

The Connect 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 AIB Merchant Services 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 AIB Merchant Services. Afterwards the customer will be redirected to your shop again. Your shop system will be notified about the payment result.
- If you prefer your customer never to leave your website, you can create your own payment forms in your individual corporate design. 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 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, Authipay 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, Authipay 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 AIB Merchant Services 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 AIB Merchant Services and the system will collect all other required information. This is the easiest way of integrating your web store Authipay. Optionally you can also use this mode with your own forms.

3 Getting Started

This section provides a simple example on how to integrate your website into Authipay in 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 Authipay, you must have the following items:

- Store Name

This is the ID of the store that was given to you by AIB Merchant Services.
For example : 13123456789

- Shared Secret

This is the shared secret provided to you by AIB Merchant Services.
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 Authipay Gateway in FullPay mode. When the cardholder clicks 'Submit', they are redirected to the AIB Merchant Services 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="GMT"/>
      <input type="hidden" name="txndatetime" value="<%
getDateTime() %>"/>
      <input type="hidden" name="hash" value="<% call
createHash( "13.00","978" ) %>"/>
      <input type="hidden" name="storename" value="13123456789"
/>
      <input type="hidden" name="mode" value="fullpay"/>
      <input type="text" name="chargetotal" value="13.00" />
      <input type="hidden" name="currency" value="978"/>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

The code presented in Appendix I represents the included file ipg-util.asp. It includes code for generating a SHA1 hash as is required by AIB Merchant Services. 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 AIB Merchant Services and you will be provided with the live production URL.

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

3.3 PHP Example

The following PHP example demonstrates a simple page that will communicate with the Authipay Gateway in FullPay mode. When the cardholder clicks 'Submit', they are redirected to the AIB Merchant Services secure pages, where they can enter their shipping, billing and payment information. After payment has been completed, the user will be redirected to the merchant's receipt page. The location of the receipt page can be configured.

```
<? 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="GMT"/>
      <input type="hidden" name="txndatetime" value="<?php echo
getDateTIme() ?>"/>
      <input type="hidden" name="hash" value="<?php echo createHash(
"13.00", "978" ) ?>"/>
      <input type="hidden" name="storename" value="13123456789"/>
      <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 AIB Merchant Services and you will be provided with the live production URL.

The code presented in Appendix II represents the included file ipg-util.php. It includes code for generating a SHA1 hash as is required by AIB Merchant Services. 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 gateway (X = mandatory field).

Field name	Description, possible values and format	„Sale“ transaction	PreAuth (credit card only)	PostAuth (credit card only)	Void
txntype	'sale', 'preauth', 'postauth' or 'void' (the transaction type – please note the descriptions of transaction types in the User Guide) 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	GMT, CET or EET (timezone of the transaction)	X	X	X	X
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X
hash	This is a SHA1 hash of the following fields : storename + txndatetime + chargetotal + currency + sharedsecret. Note, that it is important to have the hash generated in this exact order. An example of how to generate a SHA1 hash is given below.	X	X	X	X
storename	This is the ID of the store provided by AIB Merchant Services.	X	X	X	X
mode	'fullpay', 'payonly' or 'payplus' (the chosen mode for the transaction)	X	X		
chargetotal	This is the total amount of the transaction using a dot or comma as decimal separator, e. g. 12.34 for an amount of 12 Euro and 34 Cent. Group separators like (1,000.01 / 1.000,01) are not allowed.	X	X	X	X
currency	The numeric ISO code of the transaction currency, e. g. 978 for Euro (see examples on the next page)	X	X	X	

oid	The order ID of the initial action a PostAuth or Void shall be initiated for			X	X
tdate	Exact identification of a transaction that shall be voided. You receive this value as result parameter ,tdate' of the corresponding transaction.				X

Currency code list:

Currency name	Currency code	Currency number
Euro	EUR	978
Pound Sterling	GBP	826
US Dollar	USD	840
Swiss Franc	CHF	756
Australian Dollar	AUD	036
Bahrain Dinar	BHD	048
Canadian Dollar	CAD	124
Chinese Renimbi	CNY	156
Croatian Kuna	HRK	191
Czech Koruna	CZK	203
Danish Krone	DKK	208
Hong Kong Dollar	HKD	344
Hungarian Forint	HUF	348
Indian Rupee	INR	356
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 African Rand	ZAR	710
South Korean Won	KRW	410
Swedish Krona	SEK	752
Turkish Lira	TRY	949
UAE Dirham	AED	784

5 Optional Form Fields

- [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. Valid values are:

Payment method	Value
MasterCard	M
Visa (Credit/Debit/Electron/Delta)	V
Laser	L
American Express	A
Diners	C
giropay	giropay
Maestro	MA
PayPal	Paypal

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

- [oid](#)

This field allows you to assign a unique ID for your order. If you choose not to assign an order ID, the Authipay system will automatically generate one for you.

- [customerid](#)

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

- [invoicenumber](#)

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

- [refer](#)

This field describes who referred the customer to your store

- [comments](#)

Place any comments here about the transaction

- [responseSuccessURL](#)

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

- [responseFailURL](#)

The URL where you wish to direct customers after a declined or unsuccessful transaction (your Sorry URL) – only needed if not setup in Virtual Terminal / Customisation

- [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.

- [language](#)

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

Language	language
English (USA)	en_US
English (UK/Ireland)	en_GB
French	fr_FR
German	de_DE
Italian	it_IT

- [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.

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 AIB Merchant Services, 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/Delta)	Maestro	giropay	PayPal,
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"		
accountnumber	Your customer's account number (max. 11 digits)			X	
bankcode	Your customer's bank code (8 digits)			X	

6.2 PayPlus Mode

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

Field Name	Possible Values	Description
sname	Alphanumeric characters, spaces, and dashes	Ship-to Name
saddr1	Limit of 30 characters, including spaces	Shipping Address Line 1

	spaces	
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, Authipay 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 account number must not contain more than 10 digits
- A bank code needs to contain 8 digits

If the submitted data should not be valid, the payment gateway presents a corresponding error page to the card holder.

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

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

8 3D Secure

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

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.

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 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 Data Vault

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

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

- **Store or update payment information when performing a transaction**
Additionally send the parameter *hosteddataid* together with the transaction data as a unique identification for the payment information in this transaction. Depending on the payment type, credit card number and expiry date or account number and bank code will be stored under this ID if the transaction has been successful. In cases where the submitted ‘hosteddataid’ already exists for your store, the stored payment information will be updated.
- **Initiate payment transactions using stored data**
If you stored cardholder information using the Data Vault 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 AIB Merchant Services’ 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 than 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.

10 Recurring Payments

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

Field Name	Possible Values	Description
recurringInstallmentCount	Number between 1 and 999	Number of instalments 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 instalments
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.

11 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. ‘Y’ indicates that the transaction has been successful ‘N’ indicates that the transaction has not been successful “?” indicates that the transaction has been successfully initialised, but a final result is not yet available. 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. 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
cccountr	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	<p>Brand of the credit or debit card:</p> <table border="1" data-bbox="805 257 1181 533"> <tr><td>MC</td></tr> <tr><td>VISA</td></tr> <tr><td>AMEX</td></tr> <tr><td>DINERS/DISCOVER</td></tr> <tr><td>UNIONPAY</td></tr> <tr><td>MAESTRO</td></tr> <tr><td>LASER</td></tr> </table> <p>Filled with "N/A" for any payment method which is not a credit card or debit card</p>	MC	VISA	AMEX	DINERS/DISCOVER	UNIONPAY	MAESTRO	LASER
MC								
VISA								
AMEX								
DINERS/DISCOVER								
UNIONPAY								
MAESTRO								
LASER								

For 3D Secure transactions only:

Field name	Description
response_code_3dsecure	<p>Return code indicating the classification of the transaction:</p> <ul style="list-style-type: none"> 1 – Successful authentication (GICC ECI 11/10) 2 – Successful authentication without AVV (GICC ECI 11/10) 3 – Authentication failed / incorrect password (transaction declined) 4 – Authentication attempt (GICC ECI 13/12) 5 – Unable to authenticate / Directory Server not responding (GICC ECI 7) 6 – Unable to authenticate / Access Control Server not responding (GICC ECI 7) 7 – Cardholder not enrolled for 3D Secure (GICC ECI 13/12) 8 – Invalid 3D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS) <p>Please see note about blocking GICC ECI 7 transactions in the 3D Secure section of this document.</p>

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

Field name	Description
fail_reason_details	Comma separated list of missing or invalid variables
invalid_cardholder_data	<p>true – if validation of card holder data was negative</p> <p>false – if validation of card holder data was positive but transaction has been declined due to other reasons</p>

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 AIB Merchant Services and can therefore protect you from fraudulent manipulations.

The value is created with a SHA 1 Hash using the following parameter string:

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

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 gateway sends the above result parameters to a defined URL before showing the result page to the card holder. 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 SHA1 algorithm) 'notification_hash' is calculated as follows:

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

ipg-util.asp

```

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

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

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

        var ascii = getHexFromChars(stringToHash);

        var hash = calcSHA1(ascii);

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

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

```

ipg-util.php

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

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

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

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

        $ascii = bin2hex($stringToHash);

        return sha1($ascii);
    }
?>
```

Authipay is brought to you by AIB Merchant Services.

Contact Us

If you require any further assistance, please contact us:

ROI:

Telephone: **1850 200 417** *Option 4*

NI:

Telephone: **0871 200 1437** *Option 4*

UK:

Telephone: **0871 200 1436** *Option 4*

authipay@aibms.com

www.authipay.com

If you would like to find out more about our services, please contact us:

6 Belfield Office Park
Beaver Row
Clonskeagh
Dublin 4
Ireland

ROI:

Telephone: **01 218 2100**

Fax: **01 218 2199**

UK & NI:

Telephone: Locall **0845 301 5407**

Fax: Locall **0845 301 541**

sales@aibms.com

www.aibms.com

www.authipay.com

You can view further company information relating to **AIB Merchant Services** (the trading name of First Merchant Processing (Ireland) Limited, which operates in Ireland, **First Trust Merchant Services** which operates in Northern Ireland and **AIB GB Merchant Services** which operates in Great Britain at www.aibms.com. The AIB and First Trust names and AIB and Ark logos are registered trademarks that are used by the above mentioned under licence. First Merchant Processing (Ireland) Ltd trading as AIB Merchant Services is regulated by the Central Bank of Ireland.

AuthiCon2.013