



Direct API 2 Guide

Contents

Contents.....	2
Introduction.....	4
Audience	4
API Version Control.....	4
Address	4
Security / Authentication	4
Conventions	5
Quick Start Guide	6
Direct Credit Card Payment:	6
Credit Card payment via Voucher:.....	6
Pay using Virtual Card or Voucher:	7
Starting a payment session	8
Merchant Authentication	8
Customer Billing Information.....	9
Get Customer Cards – (Optional).....	11
Payment	12
Payment using a new Card	12
Store a new Card.....	14
Payment using a Stored Card.....	15
Payment using Card via Voucher	16
Payment using Bank Transfer via Voucher	18
Payment using Virtual Card	20
Payment using Voucher	21
Additional Procedures	22
Withdraw To Bank Account	22
Get Cards For Withdraw	23
Withdraw To Card.....	24
Remove Card.....	25
Withdraw To Virtual Credit Card	26
Transaction Status	28
Calculating “md5sig”	30
MD5 Pseudocode.....	30
Code Samples.....	31

Annex I – Error Handling.....	33
Annex II – Currencies and Countries	34
Currency Codes	34
Country Codes.....	34
Annex III – Test data	35

Introduction

The purpose of this document is to provide an in-depth description of Direct Payment Gateway, and the processes by which Merchants can receive money from their customers by processing a secure direct server to server transaction.

All transactions are sent using Secure Sockets Layer (SSL) encryption to protect card information as it is transmitted across the Internet.

Audience

Being technical in nature, this document is intended for Software Developers, Technical Consultants and System Analysts who are responsible for supporting the Merchant's website.

API Version Control

The information provided in this document is accurate and reliable for standard processing as of its publication date. Previous versions of the API specification should be avoided for new implementations.

Address

Production / Test:

Please contact your account manager for URL details.

Security / Authentication

All HTTP requests must be sent over SSL. It is the Merchant's responsibility to ensure a secure channel is used.

To begin testing, you must first be supplied with a gateway account; please contact your account manager to proceed.

Conventions

Terminology conventions – “Merchant” and “Customer”

Gateway system has two levels of customer:

- Merchant is a customer of gateway that uses the system to process requests.
- Customer is a customer of the merchant.

Code Examples

A field name in gateway systems is written in Courier font. A large amount of code or XML is included in a box such as the below.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
</wallet>
```

All fields that are processed through gateway systems are lower case, and there is no space or hyphen between words in order to avoid any confusion when programming. For example, the field for submitting a field including a Session ID through the system is called <sessionid>.

Quick Start Guide

Direct Credit Card Payment:

Pay using a new Credit Card (Method 1):

[Merchant Authentication](#)
[Customer Billing Info](#)
[Payment New Card](#)
[3D-Secure](#)

Pay using a new Credit Card (Method 2):

[Merchant Authentication](#)
[Customer Billing Info](#)
[Store a new Card](#)
[Payment using a Stored Card](#)
[3D-Secure](#)

Pay using a stored Credit Card:

[Merchant Authentication](#)
[Customer Billing Info](#)
[Get Customer Cards](#)
[Payment using a Stored Card](#)
[3D-Secure](#)

Credit Card payment via Voucher:

Pay using a new Credit Card via Voucher:

[Merchant Authentication](#)
[Customer Billing Info](#)
[Store a new Card](#)
[Payment using Card via Voucher](#)
[3D-Secure](#)

Pay using a stored Credit Card via Voucher:

[Merchant Authentication](#)
[Customer Billing Info](#)
[Get Customer Cards](#)
[Payment using Card via Voucher](#)
[3D-Secure](#)

Pay using Virtual Card or Voucher:

Pay using a Virtual Card:

[Merchant Authentication](#)

[Customer Billing Info](#)

[Payment using Virtual Card](#)

Pay using a pre-purchased Voucher:

[Merchant Authentication](#)

[Customer Billing Info](#)

[Payment using Voucher](#)

Starting a payment session

Merchant Authentication

The gateway XML API is session based, and so the merchant must first authenticate themselves to retrieve a Session Token, which must then be used for all subsequent communication relating to that particular transaction. All fields for this action are mandatory.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example Value
version	Defines the version of the interface	3	Y	Current API version is 2.3
action	Define first step of the iteration	255	Y	CreateSession
username	merchant username	200	Y	testname
password	merchant password	200	Y	Testpass123

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <version>2.0</version>
  <action>CreateSession</action>
  <username>testname</username>
  <password>Testpass123</password>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="CreateSession">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <expires>2013-01-25T12:18:52</expires> <!-- UTC Time -->
  </response>
</wallet>
```

Customer Billing Information

After a successful authentication you must use `CustomerBillingInfo` action to provide further details of the customer and the unique customer identifier in use by merchant system (i.e. username from merchant system should be passed to the gateway as `extusername`).

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example Value
<code>action</code>	define second step of the iteration	255	Y	<code>CustomerBillingInfo</code>
<code>sessionid</code>	session token	50	Y	<code>XXXX0001018171651FDSD</code>
<code>extusername</code>	Customers unique ID given by merchant	500	Y	<code>daredevil1979</code>
<code>title</code>	Customer's title. Accepted values: Mr, Mrs or Ms	50	Y	<code>Mr</code>
<code>firstname</code>	Customer's first name	200	Y	<code>Matthew</code>
<code>lastname</code>	Customer's last name	200	Y	<code>Murdock</code>
<code>dateofbirth</code>	Date of birth of the customer. The format is yyyy-MM-dd.	10		<code>1990-01-25</code>
<code>address</code>	Customer's address (e.g. street)	500	Y	<code>12 Green Street</code>
<code>address2</code>	Customer's address (e.g. town)	500		<code>Blue Town</code>
<code>telephone</code>	Customer's phone number. Only numeric values are accepted	20	Y	<code>0777777777</code>
<code>postcode</code>	Customer's postal code/ZIP Code. Only alphanumeric values are accepted (no punctuation marks etc.)	50	Y	<code>SW2 0GH</code>
<code>city</code>	Customer's city	200	Y	<code>London</code>
<code>state</code>	Customer's state or region.	200		<code>England</code>
<code>country</code>	Customer's country in the ISO 3166-1 alpha format	3	Y	<code>GBR</code>
<code>userip</code>	IP address of the customer. This field is required to benefit from certain features of the Fraud Control system	15	Y	<code>185.3.73.26</code>
<code>email</code>	Customer's email address	200	Y	<code>matt@murdock.com</code>

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>CustomerBillingInfo</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <extusername>daredevil1979</extusername>
  <title>Mr</title>
  <firstname>Matthew</firstname>
  <lastname>Murdoch</lastname>
  <dateofbirth>1990-01-25</dateofbirth>
  <address>12 Green Street</address>
  <address2></address2>
  <telephone>0777777777</telephone>
  <postcode>SW2 0GH</postcode>
  <city>London</city>
  <state>England</state>
  <country>GBR</country>
  <userip>185.3.73.26</userip>
  <email>matt@murdoch.com</email>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="CustomerBillingInfo">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <extusername>daredevil1979</extusername>
    <title>Mr</title>
    <firstname>Matthew</firstname>
    <lastname>Murdoch</lastname>
    <email>matt@murdoch.com</email>
  </response>
</wallet>
```

Get Customer Cards – (Optional)

Action `GetCustomerCards` retrieves the list of credit/debit card stored under customer's `extusername` in the gateway system.

This action should be used in workflows where it is required to process payment using a previously stored card for direct payment (see page 15) or for card payment via Voucher (see page 16).

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example Value
<code>action</code>	define third step of the iteration	255	Y	<code>GetCustomerCards</code>
<code>sessionid</code>	session token	50	Y	<code>XXXX0001018171651FDSD</code>
<code>currency</code>	ISO 4217 Currency Code	3	Y	<code>GBP</code>

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>GetCustomerCards</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <currency>GBP</currency>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="GetCustomerCards">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <customercards>
      <card>
        <customercardid>1345</customercardid>
        <cardtype>VISA</cardtype>
        <cardnumber>442345*****5432</cardnumber>
        <expirymonth>02</expirymonth>
        <expiryear>2013</expiryear>
        <cardnickname>Test Card 01</cardnickname>
      </card>
      <card>
        <customercardid>1346</customercardid>
        <cardtype>VISA</cardtype>
        <cardnumber>442345*****5433</cardnumber>
        <expirymonth>04</expirymonth>
        <expiryear>2014</expiryear>
        <cardnickname>Test Card 02</cardnickname>
      </card>
    </customercards>
  </response>
</wallet>
```

Payment

After successfully setting up a payment session, you must use one of the following available procedures to process a customer payment.

Payment using a new Card

Action `PaymentNewCard` allows merchants to process a customer credit/debit card payment, and optionally store the credit/debit card under customer's extusername in the system for future use.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	PaymentNewCard
sessionid	session token	50	Y	XXXX0001018171651FDSD
cardtype	VISA MASTERCARD MAESTRO	50	Y	VISA
cardnumber	Card Number	50	Y	4434567890123456
nameoncard	Cardholders name as displayed on the card	500	Y	John Smith
startmonth	MM	2	Y	02
startyear	YYYY	4	Y	2011
expirymonth	MM	2	Y	02
expiryyear	YYYY	4	Y	2014
issuenumber	Number	50		2
cardnickname	Customer's memorable nick name	200		Test VISA 01
savecard	0=false 1=true	1	Y	0
merchanttxid	Merchant Transaction ID	255	Y	TX0001
securitycode	3-digit CVV code on the rear of the customer's card	5	Y	321
amount	10.00 1000.00 10000.00 - decimal	50	Y	43
currency	ISO 4217 Currency Code	3	Y	GBP
threedstatusurl	URL on the merchants system to return the payment response in a server-side POST after 3DS authentication	500	O	
threedreturnurl	URL on the merchants website to return the customer after 3DS authentication	500	Y	

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>PaymentNewCard</action>
  <sessionid>XXXX0001018171651FDSD </sessionid>
  <cardtype>VISA</cardtype>
  <cardnumber>4111111111111111</cardnumber>
  <nameoncard>John Smith</nameoncard>
  <startmonth>02</startmonth>
  <startyear>2011</startyear>
  <expirymonth>02</expirymonth>
  <expiryyear>2014</expiryyear>
  <issuenum>2</issuenum>
  <cardnickname>Test VISA 01</cardnickname>
  <savecard>1</savecard>
  <merchanttxid>TX0001</merchanttxid>
  <securitycode>321</securitycode>
  <amount>43</amount>
  <currency>GBP</currency>
  <threedstatusurl>https://asecureurl.com/listener</threedstatusurl>
  <threedreturnurl>https://asecureurl.com/handler?x=1&y=2</threedreturnurl>
</wallet>
```

Server Response (Visa 3DS / MasterCard SecureCode Redirection)

If required by the gateway it may be necessary to perform 3D Secure with a transaction. In such cases the merchant will be required to handle the Payer Authentication redirection for a customer to complete 3D Secure authentication with their bank.

When required this will be indicated by inclusion of a `threedsecure` tag within the XML response. The merchant is required to perform a HTTP Redirect of the customer to the specified `redirecturl` which must include the specified GET querystring (note that the ampersand must be decoded to form a valid URL).

It is our recommendation that this redirect is not made within an iFrame as the ACS provider pages on which customer authentication takes place are not hosted by the gateway and so we cannot easily determine the dimensions to display these pages optimally in such circumstances.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentNewCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <threedsecure>
      <redirecturl>https://example.com/3dsecure.ashx?
sessionid=XXXX0001018171651FDSD&txtype=PaymentNewCard</redirecturl>
    </threedsecure>
  </response>
</wallet>
```

Following completion of Payer Authentication the customer will be returned to the merchant `threedreturnurl` and the Payment Authorisation response will be delivered to the merchant system; within HTTP POST parameter `depositresult` on `threedreturnurl` or instead as a server-side XML post within the response stream to the `threedstatusurl` should it be included as a field within the original payment request.

Server Response (Payment Authorisation)

The ultimate result of the Credit/Debit Card payment request will be returned following 3D Secure authentication, or directly in cases where 3D Secure is not required, in the following format.

Note: `md5sig` will only be returned in cases where payment was processed via 3D Secure (where the `threedsecure` tag is a value of 1). This is for merchants to ensure the authenticity

of the depositresult receive, and it is generated by the gateway. For details on how to calculate this hash please see page 27 of this manual

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentNewCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</ txtimestamp >
    <amount>134.76</amount>
    <currency>USD</currency>
    <threedsecure>1</threedsecure> <!--0=false | 1=true-->
    <pares>Y</pares>
    <error>0</error> <!-- 0=false -->
    <errormessage></errormessage>
    <md5sig>54DD1C8A82657E3DAAC60BA6E6BA2E10</md5sig>
  </response>
</wallet>
```

Store a new Card

Action StoreNewCard can be used to store a credit/debit card under customer's extusername in the system for future use – be that within the same payment session or a subsequent one. This is ideal for merchants with asynchronous payment workflows.

This action must be used by merchants wishing to process card payments via Voucher (see page 16).

The option to savecard in this action should be set to true to store a card within the system for use in a future payment session.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	StoreNewCard
sessionid	session token	50	Y	XXXX0001018171651FDSD
cardtype	VISA MASTERCARD MAESTRO	50	Y	VISA
cardnumber	Card Number	50	Y	4434567890123456
nameoncard	Cardholders name as displayed on the card	500	Y	John Smith
startmonth	MM	2	Y	02
startyear	YYYY	4	Y	2011
expirymonth	MM	2	Y	02
expiryyear	YYYY	4	Y	2014
issuenumber	Number	50		2
cardnickname	Customer's memorable nick name	200		Test VISA 01
savecard	0=false 1=true	1	Y	0

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>StoreNewCard</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <cardtype>VISA</cardtype>
```

```

<cardnumber>4111111111111111</cardnumber>
<nameoncard>John Smith</nameoncard>
<startmonth>02</startmonth>
<startyear>2012</startyear>
<expirymonth>01</expirymonth>
<expiryyear>2014</expiryyear>
<issuenum></issuenum>
<cardnickname>John VISA</cardnickname>
<savecard>0</savecard>
</wallet>

```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```

<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="StoreNewCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <customercardid>3456</customercardid>
    <error>0</error> <!-- 0=false -->
    <errormessage></errormessage>
  </response>
</wallet>

```

Payment using a Stored Card

Action `PaymentStoredCard` allows merchants to process a customer credit/debit card payment using a card stored under customer's `extusername` in the system.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
<code>action</code>	define third step of the iteration	255	Y	<code>PaymentStoredCard</code>
<code>sessionid</code>	session token	50	Y	<code>XXXX0001018171651FDSD</code>
<code>customercardid</code>	Captured from "customercards" or "registercards" action	50	Y	5234
<code>merchanttxid</code>	Merchant Transaction ID	255	Y	<code>TX0001</code>
<code>securitycode</code>	3-digit CVV code on the rear of the customer's card	5	Y	321
<code>amount</code>	10.00 1000.00 10000.00-decimal	50	Y	34.50
<code>currency</code>	ISO 4217 Currency Code	3	Y	GBP
<code>threedstatusurl</code>	URL on the merchants system to return the payment response in a server-side POST after 3DS authentication	500	O	
<code>threedreturnurl</code>	URL on the merchants website to return the customer after 3DS authentication	500	Y	

Request XML

```

<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>PaymentStoredCard</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <customercardid>5234</customercardid>
  <merchanttxid>TX0001</merchanttxid>
  <securitycode>321</securitycode>

```

```
<amount>34.50</amount>
<currency>GBP</currency>
<threedstatusurl>https://asecureurl.com/listener</threedstatusurl>
<threedreturnurl>https://asecureurl.com/handler?x=1&y=2</threedreturnurl>
</wallet>
```

Server Response

If required by the gateway it may be necessary to perform 3D Secure with a transaction. In such cases the merchant will be required to make the Payer Authentication redirection for a customer to complete 3D Secure authentication with their bank.

Details of this system response and what is required of a merchant is detailed on page 14 of this manual. Please note that XML returned as part of this workflow will always reference the PaymentStoredCard response type.

The ultimate result of the Credit/Debit Card payment request will be returned following 3D Secure authentication, or directly in cases where 3D Secure is not required, in the following format.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentStoredCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</txtimestamp>
    <amount>134.76</amount>
    <currency>USD</currency>
    <threedsecure>0</threedsecure> <!-- 0=false -->
    <pares></pares>
    <error>0</error> <!-- 0=false -->
    <errormessage></errormessage>
  </response>
</wallet>
```

Payment using Card via Voucher

Action `PaymentCardVoucher` allows merchants to process a customer payment by using a credit/debit card stored under customer's `extusername` in the system to buy and instantly redeem prepaid voucher.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
<code>action</code>	define third step of the iteration	255	Y	PaymentCardVoucher
<code>sessionid</code>	session token	50	Y	XXXX0001018171651FDSD
<code>customercardid</code>	Captured from "customercards" or "registercards" action	50	Y	5234
<code>merchanttxid</code>	Merchant Transaction ID	255	Y	TX0001
<code>securitycode</code>	3-digit CVV code on the rear of the customer's card	5	Y	321
<code>amount</code>	10.00 1000.00 10000.00-decimal	50	Y	34.50
<code>currency</code>	ISO 4217 Currency Code	3	Y	GBP
<code>threedstatusurl</code>	URL on the merchants system to return the payment response in a server-side POST after 3DS authentication	500		

threedreturnurl	URL on the merchants website to return the customer after 3DS authentication	500	Y	
-----------------	--	-----	---	--

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>PaymentCardVoucher</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <customercardid>5234</customercardid>
  <merchanttxid>TX0001</merchanttxid>
  <securitycode></securitycode>
  <amount>34.50</amount>
  <currency>GBP</currency>
  <threedstatusurl>https://asecureurl.com/listener</threedstatusurl>
  <threedreturnurl>https://asecureurl.com/handler?x=1&y=2</threedsreturnurl>
</wallet>
```

Server Response

If required by the gateway it may be necessary to perform 3D Secure with a transaction. In such cases the merchant will be required to make the Payer Authentication redirection for a customer to complete 3D Secure authentication with their bank.

Details of this system response and what is required of a merchant is detailed on page 14 of this manual. Please note that XML returned as part of this workflow will always reference the **PaymentCardVoucher** response type.

The ultimate result of the Credit/Debit Card payment request will be returned following 3D Secure authentication, or directly in cases where 3D Secure is not required, in the following format.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentCardVoucher">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid> TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</ txtimestamp >
    <amount>134.76</amount>
    <currency>USD</currency>
    <threedsecure>0</threedsecure> <!--0=false | 1=true-->
    </pares>
    <vouchercode>123412341234qwer</vouchercode>
    <voucherpin>12345</voucherpin>
    <error>0</error> <!-- 0=false -->
    <errormessage></errormessage>
  </response>
</wallet>
```

Payment using Bank Transfer via Voucher

Action `PaymentBankVoucher` allows merchants to process a payment via the gateway system and instantly redeem prepaid voucher.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	PaymentBankVoucher
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
amount	10.00 10.25 1000 1000.25 - decimal	50	Y	500
currency	ISO 4217 Currency Code	3	Y	TRY
voucheramount	10.00 10.25 1000 1000.25 - decimal	50	Y	161
vouchercurrency	ISO 4217 Currency Code	50	Y	EUR
paymentstatusurl	URL on the merchants system to return the payment response in a server-side POST after bank transfer	500	Y	https://asecureurl.com/listener
paymentreturnurl	URL on the merchants website to return the customer after bank transfer	500	Y	https://asecureurl.com/handler?x=1&y=2

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>PaymentBankVoucher</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <amount>500</amount>
  <currency>TRY</currency>
  <voucheramount>161</voucheramount>
  <vouchercurrency>EUR</vouchercurrency>
  <paymentstatusurl>https://asecureurl.com/listener</paymentstatusurl>
  <paymentreturnurl>https://asecureurl.com/handler?x=1&y=2</paymentreturnurl>
</wallet>
```

Server Response

Server response will contain secure payment URL. The merchant will be required to handle the redirection for a customer to complete bank transfer with bank.

The merchant is required to perform a HTTP Redirect of the customer to the specified `redirecturl` which must include the specified GET querystring (note that the ampersand must be decoded to form a valid URL).

It is our recommendation that this redirect is not made within an `iFrame` as the bank transfer provider pages on which customer authentication takes place are not hosted by the gateway and so we cannot insure that `iFrames` are supported by bank transfer provider.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentBankVoucher">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <securepayment>
<redirecturl>https://securedpaymenturl.com/banktransfer.ashx?sessionid=XXXX0001018171651FDSD
</redirecturl>
    </securepayment>
  </response>
</wallet>
```

Following completion of bank transfer the customer will be returned to the merchant paymentreturnurl and transaction partial information will be posted.

HTTP POST parameters are:

status – transaction status. Possible values are WAITING and ERROR

merchanttxid – merchant unique transaction identifier

sessionid – system session token

```
status : WAITING
merchanttxid: TX0001
sessionid: XXXX0001018171651FDSD
```

After 3 to 15 minutes, status will be posted as a server-side XML post to the paymentstatusurl.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentBankVoucher">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</txtimestamp>
    <amount>500</amount>
    <currency>TRY</currency>
    <voucheramount>161</voucheramount>
    <vouchercurrency>EUR</vouchercurrency>
    <vouchercode>123412341234qwer</vouchercode>
    <voucherpin>12345</voucherpin>
    <error>0</error>
    <errormessage></errormessage>
    <md5sig>54DD1C8A82657E3DAAC60BA6E6BA2E10</md5sig>
  </response>
</wallet>
```

Note: md5sig is for merchants to ensure the authenticity of the depositresult receive, and it is generated by the gateway. For details on how to calculate this hash please see page 27 of this manual

Payment using Virtual Card

Action `PaymentVirtualCard` allows merchants to process a payment via the customer's issued Virtual Card.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	PaymentVirtualCard
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
virtualcardcode	16 digit numeric code	16	Y	1234123412341234
virtualcardpin	2 digits of Merchant's choice and the rest masked with "x", e.g. 5xx6x, xxx43, x54xx...	5	Y	1xx5x
amount	10.00 1000.00 10000.00-decimal	50	Y	34.50
currency	ISO 4217 Currency Codes	3	Y	GBP

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>PaymentVirtualCard</action>
  <sessionid>XXXX0001018171651FDSD </sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <virtualcardcode>1234123412341234</virtualcardcode>
  <virtualcardpin>1xx5x</virtualcardpin>
  <amount>34.50</amount>
  <currency>GBP</currency>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentVirtualCard">
    <sessionid>XXXX0001018171651FDSD </sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</txtimestamp>
    <amount>34.50</amount>
    <currency>GBP</currency>
    <error>0</error> <!-- 0=false -->
    <errorMessage></errorMessage>
  </response>
</wallet>
```

Payment using Voucher

Action `PaymentVoucher` allows merchants to process a payment for customers using a prepaid voucher

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, leave empty.

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	PaymentVoucher
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
vouchercode	16 digit alpha numeric code	16	Y	123412341234QWER
voucherpin	2 digits by Merchant's choice and the rest masked with "x", e.g. 5xx6x, xxx43, x54xx...	5	N**	1xx5x
amount	10.00 1000.00 10000.00-decimal	50	N**	34.50
currency	ISO 4217 Currency Codes	3	N**	GBP

****Note:** Fields are not required only in case when voucher has no values (PIN, amount, currency) printed or voucher receptor doesn't require.

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>PaymentVoucher</action>
  <sessionid>XXXX0001018171651FDSD </sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <vouchercode>123412341234QWER</vouchercode>
  <voucherpin>1xx5x</voucherpin>
  <amount>34.50</amount>
  <currency>GBP</currency>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentVoucher">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</txtimestamp>
    <amount>34.50</amount>
    <currency>GBP</currency>
    <error>0</error> <!-- 0=false -->
    <errorMessage></errorMessage>
  </response>
</wallet>
```

Additional Procedures

The following three procedures are optional.

Withdraw To Bank Account

Action `WithdrawToBankAccount` allows customer to withdraw money to their bank account via the gateway system.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	WithdrawToBankAccount
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
amount	10.00 10.25 1000 1000.25 - decimal	50	Y	20
currency	ISO 4217 Currency Code	3	Y	GBP
iban	International Bank Account Number	200	Y	GB29NWBK60161331926819
withdrawstatusurl	URL on the merchants system to return the payment response in a server-side POST after withdraw	500	Y	https://asecureurl.com/handler?x=1&y=2
swiftcode	Bank Swift Code	200	Y	TEBUTRIS
branchcode	Bank branch code	200	Y	1231 OR Empty string

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>WithdrawToBankAccount</action>
  <sessionid>XXXX0001018171651FDSD </sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <amount>20</amount>
  <currency>GBP</currency>
  <iban>GB29NWBK60161331926819</iban>
  <withdrawstatusurl>https://asecureurl.com/handler?x=1&y=2</withdrawstatusurl>
  <swiftcode>TEBUTRIS</swiftcode>
  <branchcode></branchcode>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag with the final response which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="WithdrawToBankAccount">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <status>WAITING</status>
  </response>
</wallet>
```

XML posted to merchant status URL

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="WithdrawToBankAccount">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</txtimestamp>
    <amount>20</amount>
    <currency>GBP</currency>
    <error>0</error>
    <errormessage></errormessage>
    <md5sig>54DD1C8A82657E3DAAC60BA6E6BA2E10</md5sig>
  </response>
</wallet>
```

Get Cards For Withdraw

Action GetCardsForWithdraw gets the list of customers cards that have a minimum of one successful deposit processed.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	GetCardsForWithdraw
sessionid	session token	50	Y	XXXX0001018171651FDSD

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>GetCardsForWithdraw</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag with the final response which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="GetCardsForWithdraw">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <customercards>
      <card>
        <customercardid>1345</customercardid>
        <cardtype>VISA</cardtype>
        <cardnumber>442345*****5432</cardnumber>
        <expirymonth>02</expirymonth>
        <expiryear>2017</expiryear>
        <nameoncard>Test Card 01</nameoncard>
      </card>
      <card>
        <customercardid>1346</customercardid>
        <cardtype>VISA</cardtype>
        <cardnumber>442345*****5433</cardnumber>
        <expirymonth>04</expirymonth>
        <expiryear>2017</expiryear>
        <nameoncard>Test Card 02</nameoncard>
      </card>
    </customercards>
  </response>
</wallet>
```

Withdraw To Card

Action `WithdrawToCard` allows Customers to withdraw money to their cards via the gateway system.

To do a withdraw to card first we have to call the action `GetCardsForWithdraw` to get the list of customers cards that have a minimum of one successful deposit processed.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	WithdrawToCard
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
customercardid	Captured from "customercards" or "registercards" action	50	Y	5234
securitycode	3-digit CVV code on the rear of the customer's card	5	Y	321
amount	10.00 10.25 1000 1000.25 - decimal	50	Y	20
currency	ISO 4217 Currency Code	3	Y	GBP

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>WithdrawToCard</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <customercardid>5234</customercardid>
  <securitycode>321</securitycode>
  <amount>20</amount>
  <currency>GBP</currency>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag with the final response which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="WithdrawToCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>3075</wallettxid>
    <txtimestamp>2016-11-23T15:56:37</txtimestamp>
    <amount>20</amount>
    <currency>GBP</currency>
    <error>0</error>
    <errorMessage></errorMessage>
  </response>
</wallet>
```

Remove Card

Action `RemoveCard` enables merchants to remove a credit/debit card previously stored under customer's extusername in the the gateway system.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	RemoveCard
sessionid	session token	50	Y	XXXX0001018171651FDSD
customercardid	Captured from "customercards" or "registercards" action	50	Y	2

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>RemoveCard</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <customercardid>2</customercardid>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="RemoveCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <error>0</error> <!-- 0=false -->
    <errorMessage></errorMessage>
  </response>
</wallet>
```

Withdraw To Virtual Credit Card

Action `Withdraw To Virtual Credit Card` allows Customers to withdraw via virtual credit card via the gateway system.

To do a withdraw first we have to call the action `WithdrawToVCC` to get the payout redirect URL.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	WithdrawToVCC
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
amount	10.00 10.25 1000 1000.25 - decimal	50	Y	20
currency	ISO 4217 Currency Code	3	Y	GBP
withdrawstatusurl	URL on the merchants system to return the withdraw response in a server-side POST after withdraw	500	Y	https://merchant.com?handler.ashx

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>WithdrawToVCC</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <amount>20</amount>
  <currency>GBP</currency>
  <withdrawstatusurl>https://merchant.com?handler.ashx</withdrawstatusurl>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="WithdrawToVCC">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <payouturl>
      <redirecturl>https://dummyurl.com?hand.ashx?sessionid=001W</redirecturl>
    </payouturl>
  </response>
</wallet>
```

The next step is to post to payout redirect URL. The server response can be with virtual card display url or with xml virtual card details information. It depends how the merchant account is setup.

Server response example 1 (card details in the response)

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="WithdrawToVCC">
    <sessionid>{0}</sessionid>
    <merchanttxid>{1}</merchanttxid>
    <status>WAITING</status>
    <VCC> -- VCC can be empty
      <CardNumber>4111111111111111</CardNumber>
      <ExpiryMonth>06</ExpiryMonth>
      <ExpiryYear>22</ExpiryYear>
      <CVV>123</CVV>
    </VCC>
  </response>
</wallet>
```

Server response example 2 (display url)

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="WithdrawToVCC">
    <sessionid>{0}</sessionid>
    <merchanttxid>{1}</merchanttxid>
    <status>WAITING</status>
    <redirecturl></redirecturl> <!--redirecturl can be empty-->
  </response>
</wallet>
```

After this the final status of the withdraw will be sent to the provided withdraw status url.

Refund a Credit/Debit Card payment

Use the `RefundCreditCard` call to process a refund back to a customer's credit/debit card.

You must be approved for this feature of the API to be activated against your merchant profile; please contact your account manager to request access to this action.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	RefundCreditCard
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
wallettxid	Gateway Transaction ID	255	Y	45117
amount	10.00 1000.00 10000.00-decimal number	50	Y	34.50
currency	ISO 4217 Currency Codes	3	Y	GBP

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>RefundCreditCard</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <wallettxid>45117</wallettxid>
  <amount>34.50</amount>
```

```
<currency>GBP</currency>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="RefundCreditCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid> <!--new refund tx reference -->
    <txtimestamp>2013-01-25T12:18:52</ txtimestamp >
    <amount>34.50</amount>
    <currency>GBP</currency>
    <error>0</error> <!-- 0=false -->
    <errormessage></errormessage>
  </response>
</wallet>
```

Transaction Status

The action `TransactionStatus` returns status of a transaction which is identified using `merchanttxid` and `wallettxid`.

Note: "Y" denotes mandatory fields. If you do not have values for mandatory fields, provide "-" (dash).

Important: If the Gateway Transaction ID (`wallettxid`) is not available, provide "0" value.

Request Parameters	Description	Max. Length	Required	Example
action	define third step of the iteration	255	Y	TransactionStatus
sessionid	session token	50	Y	XXXX0001018171651FDSD
merchanttxid	Merchant Transaction ID	255	Y	TX0001
wallettxid	Gateway Transaction ID ("0" if not available)	255	Y	45117

Request XML

```
<?xml version="1.0" encoding="utf-8"?>
<wallet>
  <action>status</action>
  <sessionid>XXXX0001018171651FDSD</sessionid>
  <merchanttxid>TX0001</merchanttxid>
  <wallettxid>45117</wallettxid>
</wallet>
```

Server Response

The resulting XML will contain a `response` tag for the requested action which will be populated with child elements corresponding to the result of the operation.

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="TransactionStatus">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
```

```
<txtimestamp>2013-01-25T12:18:52</txtimestamp>  
<amount>34.50</amount>  
<currency>GBP</currency>  
<transactiontype>REFUND</transactiontype>  
<error>0</error>  
<errormessage></errormessage>  
</response>  
</wallet>
```

Calculating “md5sig”

The md5sig is constructed by performing an MD5 calculation on a string built up by concatenating the other fields returned to the merchant’s `threedstausurl` and `threedreturnurl`. This element will be returned in `depositresult` XML response when 3D secure is used during payment with one of the following API actions:

- PaymentNewCard
- PaymentStoredcard
- PaymentCardVoucher

Specifically the MD5 hash is a concatenation of the following fields:

- **merchanttxid**
- the uppercase MD5 value of the ASCII equivalent of the secretword supplied to you by your Intergration Manager
- **amount**
- **currency**
- **session**

The purpose of the md5sig field is to ensure the integrity of the data posted back to the Merchants’ server.

Please always compare the md5sig field's value posted by the gateway system with the one you calculated yourself. In order to calculate it yourself, you need to take the values of the fields listed above exactly as they were posted back to you, concatenate them and perform a MD5 calculation on this string.

MD5 Pseudocode

```
UPPER(md5(merchanttxid(UPPER(md5(secretword)))amountcurrencysessionid))
```

Code Samples

PHP Example

```
<?php

function post_to_gateway($api_host, $xml_data) {

    $ch = curl_init();

    curl_setopt($ch,CURLOPT_URL,$api_host);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: text/xml'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
    curl_setopt($ch,CURLOPT_POST, 1);
    curl_setopt($ch,CURLOPT_POSTFIELDS,$xml_data);

    $result = curl_exec($ch);
    curl_close($ch);

    return $result;

}

$api_host = 'https://example.com/wallet.ashx';

$xml_string =
'<?xml version="1.0"?>
<wallet>
    <version>2.0</version>
    <action>CreateSession</action>
    <username>testuser330</username>
    <password>testpass</password>
</wallet>';

$xml = new SimpleXMLElement($xml_string);
$xml_data = $xml->asXML();

$gateway_request = post_to_gateway ($api_host,$xml_data);

$ gateway_response = simplexml_load_string($gateway_request);

$sessionid = $ gateway_response->response->sessionid;

// With the gateway sessionid a merchant should proceed to make a
// CustomerBillingInfo request before then continuing to
// complete the required payment request

?>
```

C# example

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Web;
using System.IO;

String Response = "";
String Request=
@"<wallet><version>2.0</version><action>CreateSession</action><username>testuser330</username><password>testpass</password></wallet>";

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(new
Uri("https://example.com/wallet.ashx"));

request.Method = "POST";
request.ContentType = "text/xml;charset=utf-8";
request.UserAgent = "wallet/2.0";

byte[] bytes = Encoding.UTF8.GetBytes(Request);
request.ContentLength = bytes.Length;

ServicePointManager.ServerCertificateValidationCallback += (sender, certificate, chain,
sslPolicyErrors) => true;

        using (Stream putStream = request.GetRequestStream())
        {
            putStream.Write(bytes, 0, bytes.Length);
        }

        using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
        using (StreamReader reader = new StreamReader(response.GetResponseStream()))
        {
            Response = reader.ReadToEnd();
        }

```

Annex I – Error Handling

If an error occurs in communication with the API, the resulting XML will contain a response tag for the requested action and an error tag therein with `error_code` and `error_msg` children that contain the error message:

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="CreateSession">
    <error>
      <error_code>5201</error_code>
      <error_msg>CANNOT_LOGIN</error_msg>
    </error>
  </response>
</wallet>
```

The list of API return codes is listed below.

error_code	Associated API Action	Recommended Action
5201	CreateSession	Please check your gateway account credentials
5202	CustomerBillingInfo	Please ensure your XML request includes all required fields
5203	GetCustomerCards	Please ensure your XML request includes all required fields
5204	StoreNewCard	Please ensure your XML request includes all required fields
5205	RemoveCard	Please ensure your XML request includes all required fields
5206	PaymentNewCard	Please ensure your XML request includes all required fields
5207	PaymentStoredCard	Please ensure your XML request includes all required fields
5208	PaymentCardVoucher	Please ensure your XML request includes all required fields
5209	PaymentVirtualCard	Please ensure your XML request includes all required fields
5210	PaymentVoucher	Please ensure your XML request includes all required fields
5211	RefundCreditCard	Please ensure your XML request includes all required fields
5212	TransactionStatus	Please ensure your XML request includes all required fields
5100	Payment{0}	The payment action is not enabled for your account, please contact support

Should an error occur during customer payment this detail will be included within the transaction response:

```
<?xml version="1.0" encoding="utf-8"?>
<wallet version="2.0">
  <response type="PaymentNewCard">
    <sessionid>XXXX0001018171651FDSD</sessionid>
    <merchanttxid>TX0001</merchanttxid>
    <wallettxid>45117</wallettxid>
    <txtimestamp>2013-01-25T12:18:52</ txtimestamp >
    <amount>134.76</amount>
    <currency>USD</currency>
    <threedsecure>1</threedsecure> <!--0=false | 1=true-->
    <pares>Y</pares>
    <error>1</error> <!-- 1=true -->
    <errormessage>Not sufficient funds</errormessage>
    <md5sig>54DD1C8A82657E3DAAC60BA6E6BA2E10</md5sig>
  </response>
</wallet>
```

For the full list of issuer response codes that can be returned within the `errormessage` field please contact support.

Annex II – Currencies and Countries

Currency Codes

Please contact your account manager to confirm your currency requirements for these to be configured against your gateway account.

Currency should be passed in the ISO 4217 format as per:

https://en.wikipedia.org/wiki/ISO_4217

Currencies currently supported include, but are not limited to:

Great British Pounds	GBP	Euro	EUR
US Dollar	USD	Norwegian Krone	NOK
Canadian Dollar	CAD		

Country Codes

Gateway does not accept customers from the following countries: Afghanistan, Cuba, Myanmar, Nigeria, North Korea, Sudan, Syria, Somalia, and Yemen.

Country should be passed in the ISO 3166-1 alpha-3 format as per:

https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3

Annex III – Test data

Actions `PaymentNewCard`, `PaymentStoredCard` and `PaymentCardVoucher` on the merchant test platform can be completed using the following test details:

To test response for successful transaction	
Field	Value
Card number	4111111111111111
Expiration date	12/2015
CVV/CV2	123

To test response for declined transaction	
Field	Value
Card number	4000000000000002
Expiration date	12/2015
CVV/CV2	123

Actions `PaymentVirtualCard` and `PaymentVoucher` on the merchant test platform can be completed using the following test details:

To test response for successful transaction	
Field	Value
Voucher code	3333 4444 5555 ABCD
Pin number	Any
Currency	Any

To test response for failed transaction	
Field	Value
Voucher code	3333 4444 5555 DCBA
Pin number	Any
Currency	Any

Actions `WithdrawToCard` on the merchant test platform can be completed using the following test details:

To test response for successful withdraw to card action	
Field	Value
Card number	4111111111111111
CVV/CV2	Any

To test response for failed withdraw to card action	
Field	Value
Card number	4000000000000002
CVV/CV2	Any