Telr

Remote Integration Guide

May 10th 2016

Contents

About this guide

This guide describes the specifications of the advanced remote processing system for Telr. The intended audience is the merchant's technical staff or the merchant's system integrator.

Copyright

While every effort has been made to ensure the accuracy of the information contained in this publication, the information is supplied without representation or warranty of any kind, is subject to change without notice and does not represent a commitment on the part of Telr. We assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from this material or any part thereof, or any supplementary materials subsequently issued by Telr. Telr has made every effort to ensure the accuracy of this material.

Introduction

Merchants who collect and store their shoppers' payment details on their own platform can use the remote system as an effective payment-processing gateway. With this system, the merchant collects both order and payment details and then communicates the relevant payment details on a per order basis with payment gateway for processing. All communications with payment gateway are handled using server-to-server methods.

The benefits of using the remote system for the merchant include being able to retain full control over the payment process and also the payment pages displayed to shoppers.

However, for this system to work successfully, it is important that the merchant ensures their own system operates within a secure environment so that payment details, which they collect and store, are protected. In view of the cost involved in establishing appropriate security measures, this model only applies to merchants with established high transaction volumes.

You will need to request a remote integration for your account; this is not enabled by default.

As part of the setup process you will be supplied with an authentication key, and you will need to provide the IP address details of the servers that will be making the remote requests.

As a minimum you will need to ensure that you have the following:

- A secure (HTTPS) server used for collecting the payment details.
- Full compliance with PCI-DSS which includes (but is not limited to) secure encrypted storage of all card details.
- Ability to implement 3-D Secure authentication for E-Commerce class transactions.

Security

A core issue associated with using the remote interface is security. The collection and storage of payment information, such as card numbers and cardholder names must take place in a secure environment. Even if you do not store the card details, handling the card details (such as allowing customers to enter them within your systems and then sending those to the gateway) will require PCI DSS certification.

Payment Card Industry Data Security Standard (PCI DSS)

PCI DSS is a global Card Scheme initiative that aims to ensure that every entity that handles, stores or processes cardholder data does so in a secure manner. MasterCard and Visa have combined their own security standards for cardholder data creating an aligned program, which is now endorsed by American Express, JCB and Diners. Much of PCI DSS relates to the technology involved in capturing and processing card data and this is particularly relevant to those merchants who process and capture cardholder data on their own systems rather than those who use the Hosted Payment Pages.

For more information, please refer to PCI DSS and to the PCI Security Standards Council at: www.pcisecuritystandards.org. If you want any help to gain compliance this site also lists PCI approved Quality Security Assessors (QSA's) who can provide technical advice.

Request methods

All payment requests must be sent using the HTTP POST method to one of three possible URL's. The request data can either be in the form of standard HTTP URL encoded parameters, or it can be sent as an XML document.

The format of the response is dependent on which URL is used:

URL	Response format
https://secure.telr.com/gateway/remote.html	HTTP parameter format
https://secure.telr.com/gateway/remote.txt	Plain text
https://secure.telr.com/gateway/remote.xml	XML document

Example purchase request using HTTP URL encoded parameters:

```
ivp_store=1&ivp_authkey=x47gy2avf9&ivp_trantype=sale&ivp_tranclass=moto&
ivp_desc=Product%20details&ivp_currency=GBP&ivp_amount=24.95&ivp_test=0&
ivp_cn=4000000000000002&ivp_exm=10&ivp_exy=2012&bill_fname=Card&
bill_sname=Holder&bill_addr1=Address&bill_city=London&bill_country=GB&
bill_email=test@test.com
```

Example refund request using HTTP URL encoded parameters:

```
ivp_store=1&ivp_authkey=x47gy2avf9&ivp_trantype=refund&ivp_tranclass=ecom&
ivp_currency=USD&ivp_amount=34.70&tran_ref=010321000864&ivp_test=0
```

The same refund request sent as an XML document would be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
        <store>1</store>
        <key>x47gy2avf9</key>
        <tran>
                <type>refund</type>
                <class>ecom</class>
                <currency>USD</currency>
                <amount>34.70</amount>
                <ref>010321000864</ref>
                <test>0</test>
        </tran>
</remote>
```

You can send either request format to any of the URL's. For example, you can send the request in as HTTP URL parameters to `remote.xml` in order to get the response back as an XML document.

All data must be sent in UTF-8 encoding. In order to allow processing via the global card network, only certain characters can be used. These are:

Unicode set name	Characters allowed (hex code)
Basic Latin	0009, 000A, 000D, 0020-007E
Latin-1 Supplement	00A0-00FF
Latin Extended-A	0100-017F
Latin Extended-B	0180-024F

Transaction Categories

There are two categories of transactions, an initial transaction (a new sale for a new customer) and a follow-up transaction (such as a refund, or a continuous-authority sale such as could be used for returning customers or subscription payments)

Initial transactions require the full card details. Your systems must be PCI DSS certified in order to be able to use these.

Follow-up transactions do not have any card information, instead a reference is provided to the initial transaction and the gateway will retrieve the required card information from that initial transaction. PCI DSS certification is not required to use the follow-up transactions.

	Transaction Class		
Transaction Type	**E-Commerce**	**Mail/Telephone Order**	**Continuous Authority**
Sale	Initial (Requires PCI DSS)	Initial (Requires PCI DSS)	Follow-up
Auth	Initial (Requires PCI DSS)	Initial (Requires PCI DSS)	*Not Available*
Verify	Initial (Requires PCI DSS)	Initial (Requires PCI DSS)	*Not Available*
Void	Follow-up	Follow-up	Follow-up
Capture	Follow-up	Follow-up	*Not Available*
Release	Follow-up	Follow-up	*Not Available*
Refund	Follow-up	Follow-up	Follow-up

Any integration method can be used to obtain the transaction reference required in a follow-up transaction. For instance you can use the Hosted Payment Pages to process the initial transaction and then use remote for the follow-up. This combination would avoid any PCI DSS certification requirements for your systems as the card data would only be entered on and handled by the gateway.

The different parameters required for either initial or follow-up transactions are listed below.

HTTP URL Parameters

Description	**Parameter**	**Initial Trans**	**Follow-up**
Core transaction data			
Store ID	ivp_store	**Required**	**Required**
Authentication key *(Note 1)*	ivp_authkey	**Required**	**Required**
Transaction type	ivp_trantype	**Required**	**Required**
Transaction class	ivp_tranclass	**Required**	**Required**
Transaction description	ivp_desc	**Required**	*Not used*
Transaction cart ID *(Note 2)*	ivp_cart	**Required**	*Not used*
Transaction currency *(Note 3)*	ivp_currency	**Required**	**Required**
Transaction amount *(Note 4)*	ivp_amount	**Required**	**Required**
Previous transaction reference *(Note 5)*	tran_ref	*Not used*	**Required**
Test mode *(Note 6)*	ivp_test	**Required**	**Required**
Card details			
Card number	ivp_cn	**Required**	*Not used*
Expiry date – month *(Note 7)*	ivp_exm	**Required**	*Not used*
Expiry date – year *(Note 7)*	ivp_exy	**Required**	*Not used*
CVV *(Note 8)*	ivp_cv	Conditional	*Not used*
Card holder details			
Title (Mr, Mrs, etc)	bill_title	Optional	*Not used*
Forenames	bill_fname	**Required**	*Not used*
Surname	bill_sname	**Required**	*Not used*
Street address – line 1	bill_addr1	**Required**	*Not used*
Street address – line 2	bill_addr2	Optional	*Not used*
Street address – line 3	bill_addr3	Optional	*Not used*
City	bill_city	**Required**	*Not used*
Region/State *(Note 9)*	bill_region	Optional	*Not used*
Country *(Note 10)*	bill_country	**Required**	*Not used*
Zip/Area/Postcode *(Note 9)*	bill_zip	Optional	*Not used*
Email address	bill_email	**Required**	*Not used*
Phone number *(Note 9)*	bill_phone1	Optional	*Not used*
IP Address *(Note 11)*	bill_ip	Conditional	*Not used*
Delivery details *(Note 12)*			
Title (Mr, Mrs, etc)	delv_title	Optional	*Not used*
Forenames	delv_fname	Conditional	*Not used*
Surname	delv_sname	Conditional	*Not used*
Street address – line 1	delv_addr1	Conditional	*Not used*
Street address – line 2	delv_addr2	Optional	*Not used*
Street address – line 3	delv_addr3	Optional	*Not used*
City	delv_city	Conditional	*Not used*
Region/State *(Note 9)*	delv_region	Optional	*Not used*
Country *(Note 10)*	delv_country	Conditional	*Not used*
Zip/Area/Postcode *(Note 9)*	delv_zip	Optional	*Not used*
Phone number *(Note 9)*	delv_phone1	Optional	*Not used*
Extra data fields *(Note 13)*			
Extra data elements	xtra_*	Optional	*Not used*

Notes:

1. The Authentication Key will be provided to you as part of the Remote Integration setup process after you request that this integration type is enabled for your account.
2. An example use of the cart ID field would be your own transaction or order reference.
3. Currency must be sent as a 3 character ISO code. A list of currency codes can be found at the end of this document. For voids or refunds this must match the original transaction currency.
4. The transaction amount must be sent in major units, for example 9 dollars 50 cents must be sent as 9.50 not 950. There must be no currency symbol, and no thousands separators. The decimal part must be separated using a dot.

   For voids, the amount must exactly match the amount of the original transaction.

   For refunds, captures and releases the amount must not exceed the amount of the original transaction.
5. The previous transaction reference is required for any void, refund, capture or release transaction. It must contain the reference that was supplied in the response for the original transaction.
6. Test mode of zero indicates a live transaction. If this is set to any other value the transaction will be treated as a test.
7. Card dates must be sent as a numeric values with a two digit month (01-12) and a 4 digit year (e.g. 2010)
8. The requirement for the CVV (card security code) depends on the transaction class:

Continuous Authority	- Not Used
Mo/To	- Optional (** - see below*)
E-Commerce	- Required

9. Some fields (such as region, zip code and phone number) are optional within the gateway itself, but may be mandated by the acquirer you are using. If your acquirer mandates these details, then they must be supplied as part of the request.
10. Country must be sent as a 2 character ISO code. A list of country codes can be found at the end of this document.
11. The requirement for the IP address depends on the transaction class. Where it is sent, it must be sent in the dotted-decimal format, for example 1.2.3.4

Continuous Authority	- Not Used
Mo/To	- Optional (** - see below*)
E-Commerce	- Required

12. Delivery details are optional, but if used then as a minimum the forename, surname, address line 1, city and country must be sent. Acquirer specific requirements for details such as region and zip code also apply here.
13. The optional extra data fields can be used to hold additional data that relates to this transaction. The parameter name must be a between 6 and 20 characters, consist only of lower case a-z, digits 0-9 and the underscore. The name must start 'xtra_'. For example, a customer account number could be sent as 'xtra_acnum'.

    There are two reserved fields that cannot be used – 'xtra_fields' and 'xtra_signature'

** Although these values (CVV and IP Address) are optional within the gateway for Mo/To class transactions, some acquiring banks may mandate one or both of these. Please check with support to confirm if these are required for your account.*

Transaction Types

The `ivp_trantype` and `ivp_tranclass` parameters set the type of transaction that will be processed. These set both the processing action and the processing category. The options are:

`ivp_trantype`	
auth	Seek authorisation from the card issuer for the amount specified. If authorised, the funds will be reserved but will not be debited until such time as a corresponding capture command is made. This is sometimes known as pre-authorisation.
capture	Debit the funds that have been reserved on a card using a previous auth transaction. The amount that to be captured can be any amount up to the full amount of the initial auth. It cannot exceed the initial amount. This is sometimes known as post-authorisation or as pre-authorisation completion.
release	Release funds that had previously been reserved using an auth transaction. It is possible for the amount that is captured to be lower than the initial auth amount. In this case the remaining balance should be released back to the card holder. A release transaction cannot be voided. The full amount of the auth can be released in order to cancel the transaction. A release cannot be done on any transaction type other than an auth.
sale	Immediate purchase request. This has the same effect as would be had by performing an auth transaction followed by a capture transaction for the full amount. No additional capture stage is required.
refund	Credit the amount specified back to the card holder. A refund can be processed against a sale or capture transaction.
void	Cancel either a sale, refund or capture transaction. This can only be done within a few hours of the transaction you are attempting to cancel.
verify	Confirm that the card details given are valid. No funds are reserved or taken from the card.
`ivp_tranclass`	
moto	Process as Mail Order / Telephone Order.
ecom	Process as an Internet based E-Commerce transaction. The use of 3-D Secure is mandatory for this class of transaction.
cont	Process as a continuous authority transaction, for example a recurring subscription.

For E-Commerce transactions it is mandatory that you also implement 3-D Secure as part of the authorisation process. The payment gateway provides remote access to an MPI which will simplify this process for you. See the section on 3-D Secure later in this document.

For items or services that have an immediate delivery, the sale transaction method is often the simplest one to use. If there can be a delay between the transaction and eventual shipping of the goods, then separate auth and capture transactions should be used. You must ensure that you perform the capture stage of the transaction otherwise the funds will not be taken from the card and you will not receive payment. You should process the capture request before actually shipping the goods to ensure that the card used is still valid.

XML Request layout

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
        <store>Store ID</store>
        <key>Authentication Key</key>
        <tran>
                <type>Transaction type</type>
                <class>Transaction class</class>
                <cartid>Transaction cart ID</cartid>
                <description>Transaction description</description>
                <test>Test mode</test>
                <currency>Transaction currency</currency>
                <amount>Transaction amount</amount>
                <ref>Previous transaction reference</ref>
        </tran>
        <card>
                <number>Card number</number>
                <expiry>
                        <month>Expiry date – month</month>
                        <year>Expiry date – year</year>
                </expiry>
                <cvv>CVV</cvv>
        </card>
        <billing>
                <name>
                        <title>Title</title>
                        <first>Forenames</first>
                        <last>Surname</last>
                </name>
                <address>
                        <line1>Street address – line 1</line1>
                        <line2>Street address – line 2</line2>
                        <line3>Street address – line 3</line3>
                        <city>City</city>
                        <region>Region</region>
                        <country>Country</country>
                        <zip>Zip/Area/Postcode</zip>
                </address>
                <email>Email address</email>
                <ip>IP address</ip>
        </billing>
</remote>
```

To include the delivery details, add a <delivery> .... </delivery> section as follows

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
        <store>Store ID</store>
        <key>Authentication Key</key>
        <tran>
                ....
        </tran>
        <card>
                ....
        </card>
        <billing>
                ....
        </billing>
        <delivery>
                <name>
                        <title>Title</title>
                        <first>Forenames</first>
                        <last>Surname</last>
                </name>
                <address>
                        <line1>Street address – line 1</line1>
                        <line2>Street address – line 2</line2>
                        <line3>Street address – line 3</line3>
                        <city>City</city>
                        <region>Region</region>
                        <country>Country</country>
                        <zip>Zip/Area/Postcode</zip>
                </address>
        </delivery>
</remote>
```

To include the extra data fields, add a <extra> .... </extra> field as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
        <store>Store ID</store>
        <password>Store password</password>
        <tran>
                ....
        </tran>
        <card>
                ....
        </card>
        <billing>
                ....
        </billing>
        <extra>
                <val1>data for xtra_val1</val1>
                <acnum>data for xtra_acnum</acnum>
                ....
        </extra>
</remote>
```

To convert the ‘xtra_name’ format to xml, simply remove the ‘xtra_’ part and include the remainder inside the <extra>…</extra> section.

You can include both delivery and extra data sections if required.

Authorisation response

The actual format of the response will depend on the URL used in the request. All formats contain the same details; you simply select whichever format is the easiest for you to process and integrate with your existing systems.

Field	**Description**
auth_status	Authorisation status. A indicates an authorised transaction. Any other value indicates that the request could not be processed.
auth_code	If the transaction was authorised, this contains the authorisation code from the card issuer. Otherwise it contains a code indicating why the transaction could not be processed.
auth_message	The authorisation or processing error message.
auth_tranref	The payment gateway transaction reference allocated to this request.

auth_cvv	Result of the CVV check:
	Y = CVV matched OK
	N = CVV not matched
	X = CVV not checked
	E = Error, unable to check CVV

auth_avs	Result of the AVS check:
	Y = AVS matched OK
	P = Partial match (for example, post-code only)
	N = AVS not matched
	X = AVS not checked
	E = Error, unable to check AVS

The AVS check is currently only available for cards issued in the United Kingdom, United States of America or Canada.

Response examples

Authorised – via remote.html

Response is on a single line:

```
auth_status=A&auth_code=tst123&auth_message=Authorised&
auth_tranref=010321000871&auth_cvv=Y&auth_avs=X
```

Authorised – via remote.txt

Response is on multiple lines:

```
auth_status=A
auth_code=tst123
auth_message=Authorised
auth_tranref=010321000871
auth_cvv=Y
auth_avs=X
```

Authorised – via remote.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
        <auth>
                <status>A</status>
                <code>tst123</code>
                <message>Authorised</message>
                <tranref>010321000871</tranref>
                <cvv>Y</cvv>
                <avs>X</avs>
        </auth>
</remote>
```

Not authorised - via remote.html

```
auth_status=E&auth_code=02&auth_message=Invalid%20card%20number&
auth_tranref=000321000874&auth_cvv=X&auth_avs=X
```

Tokenization

Tokenization is the process of replacing some piece of sensitive data with a value that is not considered sensitive in the context of the environment that consumes the token and the original sensitive data.

Transaction level

The payments gateway uses a transaction level tokenization by default. In this way, only the initial transaction requires the card details. In all other transactions (such as a Refund, Void, or further authorisations when using the Continuous Authority class) the card details do not form part of the transaction request. Instead, the request must include the payment gateway transaction reference for the initial transaction. All required details, such as card number and expiry date, will be retrieved from the transaction record held by gateway.

As the card details are not required for any follow-up transactions, merchants can simplify their PCI DSS compliance by not storing any of the card details. The transaction reference cannot be used to obtain a copy of the card details.

If additional sale transactions using the Mo/To or E-Commerce class transactions are required, then the full card details will need to be sent as part of the request.

Recurring transactions

Using the Continuous Authority transaction class, you can process additional authorisations against a card. This could be used, for example, in a subscription based system where the customer is required to pay a set amount every month.

The first transaction must be processed as either an E-Commerce transaction or as a MoTo (Mail Order / Telephone Order) transaction. If this transaction is authorised, then additional transactions can be processed as Continuous Authority.

These additional transactions requests are processed by setting a transaction type of ‘sale’, with a transaction class of ‘cont’. Although this is a sale request, it is treated as a follow-up transaction in the same way as would be done for a refund or void. Card and billing details must not be sent as part of the request, it must instead use the ‘tran_ref’ field to set the transaction ID of the initial authorisation. The card and billing details will be retrieved from that initial authorisation. Other transaction details, such as cart ID and description, must be sent as per a normal Sale request.

The initial transaction must be an authorised sale, using either the E-Commerce or MoTo transaction class. If the transaction referenced is not a sale, was not authorised, or is not of the required class then the authorisation request will be rejected.

Should a customer require changing of their card details, maybe through needing to use a different card or simply down to the existing card expiring, then a new E-Commerce or MoTo transaction will be required to make an initial authorisation based on the new card details. All recurring transactions after that point should now refer to this new transaction and not the original transaction.

3-D Secure

The 3-D Secure protocol was developed by Visa to improve the security of Internet payments. It is designed to allow authentication of cardholders by their issuers at participating merchants. The objective is to benefit all participants by providing issuers the ability to fully authenticate cardholders during an online purchase, reducing the likelihood of fraudulent usage of Visa cards and improving overall transaction performance. It has since been licensed by other card schemes such as MasterCard, JCB and American Express. Each card scheme has its own brand name for 3-D Secure:

Scheme	3-D Secure brand name
Visa	Verified by Visa
MasterCard	SecureCode
JCB	J/Secure
American Express	SafeKey

The card holder verification takes place on a server called an Access Control Server (ACS) which is operated by the card issuer. The merchant or payment gateway is not involved in capturing or processing any of the authentication details.

The advantage for merchants is the reduction of "unauthorised transaction" chargebacks. The main advantage for cardholders is that there is a decreased risk of other people being able to use their payment cards fraudulently on the Internet.

In most current implementations of 3-D Secure, the issuing bank prompts the buyer for a password that is known only to the bank and the buyer. Since the merchant does not know this password and is not responsible for capturing it, it can be used by the issuing bank as evidence that the purchaser is indeed their cardholder. This decreases risk in two ways:

1. Copying card details, either by writing down the numbers on the card itself or by way of modified terminals or ATMs, does not result in the ability to purchase over the Internet because of the additional password, which is not stored on or written on the card.
2. Since the merchant does not capture the password, there is a reduced risk from security incidents at online merchants - there is no way for anyone to get the associated password.

Merchant integration with the 3-D Secure system is handled via a Merchant Plug-In (MPI). Normally this is an additional component that has to be installed within the merchants processing systems, and can be an expensive element to purchase and maintain.

NOTE: 3-D Secure is mandatory for E-Commerce class transactions.

Using 3-D Secure with the remote integration

To avoid the requirement for merchant systems to have their own MPI, The Remote Integration supports the option for merchants to use the gateway MPI without needing to install any additional components. This method also simplifies the work that would otherwise be needed to integrate your systems with an MPI.

Basic details of the transaction need to be sent to the remote MPI interface, this includes the card number, expiry date, transaction currency and amount. This information will be used to check if card holder verification is required. The response from the MPI interface will include a session ID, and may also include details of the ACS (Access Control Server) that needs to be used along with the data that needs to be sent to that ACS.

If there are no ACS details within the response, then you can immediately proceed with the transaction processing – however you must include the MPI session ID within the transaction request. Under the 3-D Secure system the transaction can be covered by the additional liability shift offered by 3-D Secure even if additional cardholder authentication is not required. If you do not include the MPI session ID within the transaction request, then your transaction cannot be covered by this possible liability shift.

If ACS details are returned, then you have to direct the consumer to the URL provided. This URL is operated by the card issuer, not by payment gateway. At this point the consumer is required to provide whatever additional identification the card issuer requires, and once that has been completed an authentication response will be returned to your system. Your system must include that authentication response, and the MPI Session ID, within the transaction processing request.

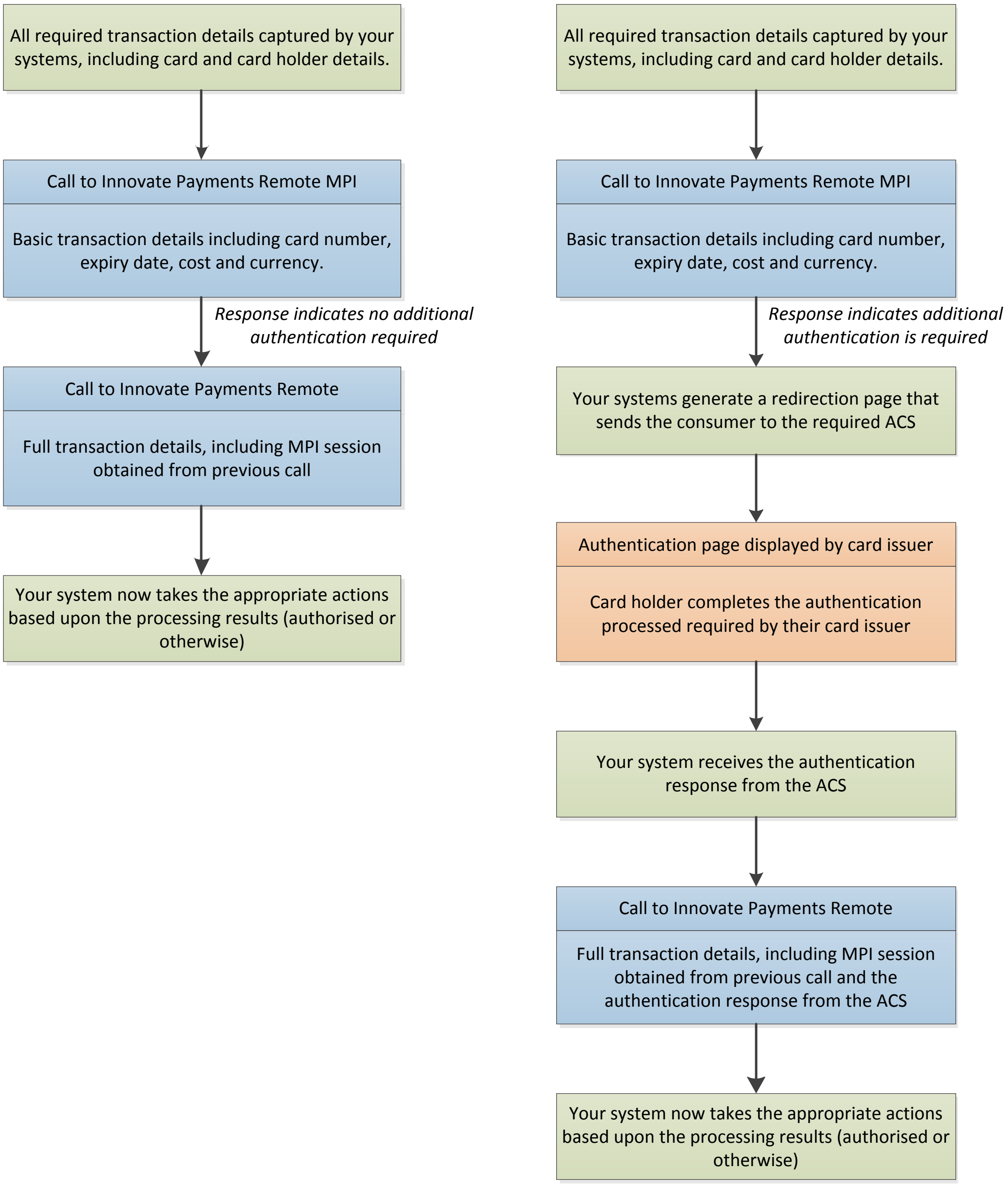
The two possible process flows are as shown on the following page.

MPI processing is only applicable to E-Commerce class transactions, and only for Sale transactions. It cannot be used for used for other transaction classes or methods.

NOTE: 3-D Secure is mandatory for E-Commerce class transactions.

Possible process flows

There are two possible process flows depending on if additional authentication is required or not.


All required transaction details captured by your systems, including card and card holder details.
Call to Innovate Payments Remote MPI
Basic transaction details including card number, expiry date, cost and currency.
Response indicates no additional authentication required
Call to Innovate Payments Remote
Full transaction details, including MPI session obtained from previous call
Your system now takes the appropriate actions based upon the processing results (authorised or otherwise)
All required transaction details captured by your systems, including card and card holder details.
Call to Innovate Payments Remote MPI
Basic transaction details including card number, expiry date, cost and currency.
Response indicates additional authentication is required
Your systems generate a redirection page that sends the consumer to the required ACS
Authentication page displayed by card issuer
Card holder completes the authentication processed required by their card issuer
Your system receives the authentication response from the ACS
Call to Innovate Payments Remote
Full transaction details, including MPI session obtained from previous call and the authentication response from the ACS
Your system now takes the appropriate actions based upon the processing results (authorised or otherwise)

Request methods

All payment requests must be sent using the HTTP POST method to one of three possible URL's. The request data can either be in the form of standard HTTP URL encoded parameters, or it can be sent as an XML document.

The format of the response is dependent on which URL is used:

URL	Response format
https://secure.telr.com/gateway/remote_mpi.html	HTTP parameter format
https://secure.telr.com/gateway/remote_mpi.txt	Plain text
https://secure.telr.com/gateway/remote_mpi.xml	XML document

Example MPI request using HTTP URL encoded parameters:

```
ivp_store=1&ivp_authkey=x47gy2avf9&ivp_trantype=sale&ivp_tranclass=ecom&
ivp_desc=Product%20details&ivp_currency=GBP&ivp_amount=24.95&ivp_test=0&
ivp_cn=4000000000000002&ivp_exm=10&ivp_exy=2012
```

The same request sent as an XML document would be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
      <store>1</store>
      <key>x47gy2avf9</key>
      <tran>
            <type>sale</type>
            <class>ecom</class>
            <description>Product details</description>
            <currency>GBP</currency>
            <amount>24.95</amount>
            <test>0</test>
      </tran>
      <card>
            <number>4000000000000002</number>
            <expiry>
                  <month>10</month>
                  <year>2012</year>
            </expiry>
      </card>
</remote>
```

You can send either request format to any of the URL's. For example, you can send the request in as HTTP URL parameters to `remote_mpi.xml` in order to get the response back as an XML document.

All data must be sent in UTF-8 encoding. In order to allow processing via the global card network, only certain characters can be used. These are:

Unicode set name	Characters allowed (hex code)
Basic Latin	0009, 000A, 000D, 0020-007E
Latin-1 Supplement	00A0-00FF
Latin Extended-A	0100-017F
Latin Extended-B	0180-024F

HTTP URL Parameters

Description	**Parameter**
Core transaction data	
Store ID	ivp_store
Authentication key	ivp_authkey
Transaction type	ivp_trantype - Must be 'sale'
Transaction class	ivp_tranclass - Must be 'ecom'
Transaction cart ID	ivp_cart
Transaction description	ivp_desc
Transaction currency *(Note 1)*	ivp_currency
Transaction amount *(Note 2)*	ivp_amount
Test mode *(Note 3)*	ivp_test
Card details	
Card number	ivp_cn
Expiry date – month *(Note 4)*	ivp_exm
Expiry date – year *(Note 4)*	ivp_exy
Browser details	
Browser user agent *(Note 5)*	brsr_useragent
Browser accept header *(Note 6)*	brsr_accepthdr

Notes:

1. Currency must be sent as a 3 character ISO code. A list of currency codes can be found at the end of this document.
2. The transaction amount must be sent in major units, for example 9 dollars 50 cents must be sent as 9.50 not 950. There must be no currency symbol, and no thousands separators. The decimal part must be separated using a dot.
3. Test mode of zero indicates a live transaction. If this is set to any other value the transaction will be treated as a test.
4. Card dates must be sent as a numeric values with a two digit month (01-12) and a 4 digit year (e.g. 2016)
5. The browser user agent data is the content of the 'User-Agent' header provided to your server by the consumers browser.
6. The browser accept data is the content of the 'Accept' header provided to your server by the consumer's browser.

XML Request layout

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
	<store>Store ID</store>
	<key>Authentication key</key>
	<tran>
		<type>Transaction type</type>
		<class>Transaction class</class>
		<cartid>Transaction cart ID</cartid>
		<description>Transaction description</description>
		<test>Test mode</test>
		<currency>Transaction currency</currency>
		<amount>Transaction amount</amount>
	</tran>
	<card>
		<number>Card number</number>
		<expiry>
			<month>Expiry date - month</month>
			<year>Expiry date - year</year>
		</expiry>
	</card>
	<browser>
		<agent>Browser User-Agent header</agent>
		<accept>Browser Accept header</accept>
	</browser>
</remote>
```

MPI response

The actual format of the response will depend on the URL used in the request. All formats contain the same details; you simply select whichever format is the easiest for you to process and integrate with your existing systems.

Field	Description
mpi_session	Session reference for this transaction. Must be included in the remote transaction processing request that is sent to authorise the transaction.
mpi_acsurl	The URL of the Access Control Server to be used for the additional authentication stage, if any
mpi_pareq	Data to be sent to the ACS

If the remote MPI request could not be processed, then no mpi_session will be supplied and instead there will be an error message. The response may also include other fields (such as mpi_xid, mpi_scheme and mpi_level) which are not used as part of this integration and can be ignored. Only the 3 fields shown in the table above are relevant to this integration method.

Response XML

```
	<?xml version="1.0" encoding="UTF-8"?>
	<remote>
		<mpi>
			<session>MPI Session ID</session>
			<acsurl>Access Control Server URL</acsurl>
			<pareq>Data to be sent to the ACS</pareq>
		</mpi>
	</remote>
```

ACS Redirect

If the response indicates that additional authorisation is required, then your systems must direct the consumer to the required page. This must be done using the HTTP POST method to the URL given. The data provided in the mpi_pareq field must be sent to the ACS as a field called 'PaReq' – the data must not be altered in any way otherwise the transaction will fail.

The mpi_pareq data can be large; your systems must be able to accept any size of data in this field. They must also be able to accept an ACS URL of up to 2048 bytes.

In addition to the PaReq field, you must also supply a field known as 'TermUrl'. This contains the URL within your systems that you want the ACS to return the consumer to. The ACS will make a HTTP POST to this URL, which will contain the result of the authentication.

A further optional field, called 'MD', can also be supplied. The content of this field will be sent back as part of the data returned to your systems along with the authentication result. You can use this field as a method of tracking the payment session.

As HTTP POST must be used for ACS redirect, you will need to generate a page with a HTML form containing all of the required fields. This can be automatically submitted using JavaScript, but you must also provide an alternate submission method in case the consumers browser does not support JavaScript or does not have it enabled.

An example of this would be:

```
<form name="acsform" action="[acsurl obtained from the MPI request]" method="post">
<input type="hidden" name="PaReq" value="[The pareq data from the MPI request]">
<input type="hidden" name="MD" value="[Optional transaction reference]">
<input type="hidden" name="TermUrl" value="[return URL on your site]">
<noscript><input type="Submit"></noscript>
</form>
<script>
function autosub() {
  document.forms['acsform'].submit();
}
document.onload=autosub;
</script>
```

After the authentication process has been completed, the ACS will make a HTTP POST back to the URL you gave as the 'TermUrl' field. This will contain two fields, the first is the 'MD' field which will contain the same data as you provided in the request. The second is 'PaRes' which contains the result of the authentication.

The ACS page can be displayed as a full page by the browser, or it can be included as an inline frame within another page. The inline frame dimensions must be a minimum of 390 pixels wide by 400 pixels high. The page must not be displayed as a pop-up. See the section on 'Merchant Requirments' later in this document for more information.

Completing the transaction

Once the ACS has sent the authentication response back to your systems, you can then complete the transaction processing using the usual remote request. You must include the MPI session ID and the authentication response that was received. This response must not be altered in any way otherwise the transaction will fail. The response data can be large; your systems must be able to accept any size of data in this field.

The additional HTTP URL parameters for the remote transaction processing are:

Description	Parameter
MPI Session ID	mpi_session
Authentication response	mpi_pares

If there was no additional authorisation stage required, you must still include the mpi_session value in the transaction processing request.

The details of the transaction (such as card number, amount, currency, cart ID and description) must be identical in the transaction processing request and the MPI request. If any details are changed, the transaction will not be able to be processed.

To include the fields within a XML request for transaction processing, they should be sent as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
        <store>Store ID</store>
        <key>Authentication key</key>
        <tran>
                ....
        </tran>
        <card>
                ....
        </card>
        <billing>
                ....
        </billing>
        <mpi>
                <session>MPI Session ID</session>
                <pares>Authentication response</pares>
        </mpi>
</remote>
```

Implementation best practices

3-D Secure has now been in use for several years, and valuable lessons can be learned from merchant and cardholder experiences during this time. The following points will assist you in implementing 3-D Secure in a way which is compliant with scheme requirements, and most importantly easy for the cardholder to use and therefore complete transactions.

Use of Inline Page instead of Pop-Up

Originally many merchants implemented a pop-up window for 3-D Secure, i.e. the authentication process appeared in a separate window to the payment system.

Pop-up suppression software has gained market adoption through stand-alone applications as well options built in to modern browsers. These will usually prevent the window from being displayed. Cardholders have also learned to disregard pop-up windows due to the high frequency of pop-up windows being used for unsolicited advertisements, and may just close them automatically without looking at them.

It is also important to note that as of 2004, Visa and MasterCard have prohibited the use of pop-up pages for 3-D Secure.

Pre-Authentication Message

You should include a brief message to cardholders on the checkout page informing them that they may be required to provide additional authentication details. The message must not say that this will happen, as in some cases the process may continue without an authentication screen. This message should be placed on the page where it is likely the cardholder will notice and read it – for example next to the button that has to be clicked to make the payment.

A recommended pre-authentication message is as follows:

> The next screen you see may be payment card verification through your card issuer.

3-D Secure logos

You should display the appropriate 3-D Secure logos (such as Verified by Visa or MasterCard SecureCode) on the payment page as an additional method of indicating to the cardholder that the authentication steps may be required. If also serves to reassure cardholders if the authentication page is displayed as they have already seen the relevant logos within your payment page.

Inline options – Framed or Full page

With full page display, the entire browser display is replaced with the 3-D Secure authentication page, before returning to your systems where you can then display your own pages again. This has the benefit of being a simpler implementation with fewer possibilities for mistakes, however it can disrupt the payment process as the browser navigation bar will show a different domain to the one the cardholder has been using so far, and the page layout will be different which can cause confusion.

With the framed approach you have more control over the page display, and the browser navigation bar continues to show the address of your site. This can be a more difficult method to implement.

Points to keep in mind when using the framed approach:

- Provide enough space so that the 3-D Secure frame can fit. The minimum size required is 390x400 pixels.
- Avoid having a 'crowded' screen which could cause confusion
- Ensure that the 3-D Secure frame is not pushed out of the viewable area on low resolution screens. The 3-D Secure frame must be viewable without having to scroll.
- There should not be other links or exit points that may distract the cardholder from completing the 3-D Secure process (such as search options, navigation menus etc)
- All page elements must be HTTPS. Never mix HTTP and HTTPS.

Within the framed approach you should also display a message that advises the cardholder not to click back or reload as that may cause problems with the authentication process.

Activity indicators

You should display something to the cardholder that lets them know authorisation is in progress. It may take several seconds for the initial cardholder enrolment checks to be undertaken, and you don't want the cardholder to think that the process has stopped and abort the purchase.

The activity indicators should include some visual movement to reassure the cardholder that the process is continuing.

Activity indicators should be displayed during the initial MPI check to find out if additional authentication is required, and again during the actual authorisation process.

No promotional messages.

Visa mandate that merchants must not display promotional messages to cardholders during the payment process, for example messages advertising special offers.

Example transaction request

The following shows an example of sending a continuous-authority sale transaction using PHP. This would be a follow-up transaction to an existing authorised e-commerce or moto class transaction.

You must replace the example values given below with actual valid details for your store, such as your actual store ID and authentication key, and the actual product/cost details.

```
$params = array(
        'ivp_store'           =>      'Your Store ID',
        'ivp_authkey'         =>      'Your Authentication Key',
        'ivp_trantype'        =>      'sale',
        'ivp_tranclass'       =>      'cont',
        'ivp_desc'            =>      'Product Description',
        'ivp_cart'            =>      'Your Cart ID',
        'ivp_currency'        =>      'AED',
        'ivp_amount'          =>      '100.00',
        'tran_ref'            =>      '12 digit reference of intial ecom/moto transaction',
        'ivp_test'            =>      '1'
);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://secure.telr.com/gateway/remote.html");
curl_setopt($ch, CURLOPT_POST, count($params));
curl_setopt($ch, CURLOPT_POSTFIELDS,$params);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Expect:'));
$results = curl_exec($ch);
curl_close($ch);
```

To keep the sample code provide short and easy to read, no error detection or error handling is included. You must check that the curl request has worked as expected (status result of 200 OK), and that you have valid response data.

Report download

The remote API can also be used to obtain a list of available reports, and to download a report. These are the same reports as are available through the download pages within the merchant administration system.

Request methods

All payment requests must be sent using the HTTP POST method to the report URL. The report list is only available as an XML document. The request data can either be in the form of standard HTTP URL encoded parameters, or it can be sent as an XML document.

URL	Response format
https://secure.telr.com/gateway/reports.xml	XML document

HTTP URL Parameters

Description	Parameter
Authentication	
Store ID	ivp_store
Authentication key	ivp_authkey
Report details	
File reference	ivp_fileref

XML Request layout

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
       <store>Store ID</store>
       <key>Authentication key</key>
       <fileref>File reference</fileref>
</remote>
```

Usage options

If no file reference is provided, then the API will generate an XML document containing a list of the available reports.

```
<?xml version="1.0" encoding="UTF-8"?>
<remote>
       <reports>
              <report>
              <id>Report ID</id>
              <date>Report date</date>
              <type id="Report type ID">Report type name</type>
              <file>
                     <ref>Report file reference</ref>
                     <name>Report file name</name>
              </file>
              </report>
       </reports>
</remote>
```

The <report>...</report> section may be repeated multiple times, depending on the number of available reports. A maximum of the 50 most recent reports will be listed. Within each report section, there will be one or more file sections – some report types generate multiple files, for example a daily transaction report may generate both a summary file and a details file.

To download a specific file, set the file reference in the request to the value that is held in the <ref> part of the <file> section.

Response content types

When downloading the list of available reports, the content type will be:

```
Content-Type: text/xml;charset=UTF-8
```

When downloading a specific report, the response content type will reflect the type of file being downloaded. The response headers will also contain the filename for the report.

```
Content-Disposition: attachment; filename=DailySummary_20120620.zip
Content-Type: application/zip
```

Authorisation Response codes

Status	Code	Message
A	*Set by issuer*	Authorised
H	*Set by issuer*	Authorised, but placed on hold. Manual inspection required
E	01	Invalid request
E	02	Transaction cost or currency not supplied
E	03	Cart ID not set
E	04	Invalid store ID
E	05	Transaction cost or currency not valid
E	06	Invalid transaction mode
E	07	Card expiry not supplied
E	10	Card number not supplied
E	11	Invalid card number
E	12	Card expired
E	14	Card type mismatch
E	15	Invalid card security code
E	16	Card security code not supplied
E	17	Name not valid/not supplied
E	18	Address not valid/not supplied
E	19	Country not valid/not supplied
E	20	IP address not valid/not supplied
E	21	Card/Currency combination not supported
E	22	Invalid transaction reference
E	23	Amount differs from original
E	24	Currency differs from original
E	25	Original transaction not authorized
E	26	Original transaction already voided
E	27	Original transaction not a sale
E	28	Original transaction not a refund
E	29	Amount greater than available balance
E	30	Card details differ from original
D	31	Not authorized
D	32	Original transaction cannot be voided
C	33	Transaction cancelled
D	34	No response
E	35	Unable to refund
E	36	Previous transaction is on hold
D	37	Blocked by acquirer
E	38	Invalid expiry date
E	39	Invalid transaction class
E	40	Invalid transaction type
D	41	Insufficient funds
D	42	Card security code mismatch
E	43	Email not valid/not supplied
E	44	Phone number not valid/not supplied
E	45	Transaction mode differs from original
D	46	3DSecure authentication not available for this card
D	47	3DSecure authentication rejected

Status	Code	Message
E	48	Description not set
D	49	Sold out
E	50	Card is for ATM use only
D	51	Transaction part 1 not authorised
D	52	Transaction part 2 not authorised
X	53	Authorisation expired
E	54	Transaction part not specified
E	55	Unable to access transaction part
E	56	Duplicate transaction
D	57	Continuous authority not available on referenced transaction
D	80	Not authorised *Card Filter module. Message text can be changed.*
D	90	Not authorised
D	91	Not authorised
D	92	Not authorised
D	93	Card limit exceeded
D	94	Not authorised
E	98	Internal system error
E	99	Unknown error

Response code 01 (Invalid Request)

This indicates that at least one aspect of the request is not valid. It most cases a transaction exception will be generated and these can be viewed within the merchant admin system.

There are many possible causes of the 'Invalid Request' response. In order to help maintain system security the exact cause is not contained with the response. For example the reason could be that the request has been received from an un-authorised IP address, or that the password used is not correct. This information is not divulged as part of the response data, you will need to view the exception report to see the full details.

A common cause of this error is attempting to use the remote integration without first requesting that it is added to your account. **You must first request that the remote integration is added to your account before you start to use it – this is not enabled by default**. As part of the activation process you will be supplied with the authentication key, and you will need to provide the IP address details of the servers which will be making the remote requests. Please ensure that you have read the Introduction, Security and PCI DSS sections at the start of this guide.

Please also keep in mind that the use of 3-D Secure is mandatory for E-Commerce class transactions.

Supported Currency Codes

AED	United Arab Emirates Dirham
BHD	Bahraini Dinar
CAD	Canadian Dollar
EUR	Euro
IDR	Indonesian Rupiah
GBP	Pound Sterling
JPY	Japanese Yen
KHR	Cambodian Riel
KWD	Kuwaiti Dinar
MYR	Malaysian Ringgit
OMR	Omani Rial
PHP	Philippine Peso
QAR	Qatari Rial
SAR	Saudi Riyal
SGD	Singapore Dollar
THB	Thai Baht
USD	US Dollar
VND	Vietnamese Dong

ISO Country Codes

AF	Afghanistan
AL	Albania
DZ	Algeria
AS	American Samoa
AD	Andorra
AO	Angola
AI	Anguilla
AG	Antigua and Barbuda
AR	Argentina
AM	Armenia
AW	Aruba
AU	Australia
AT	Austria
AZ	Azerbaijan
BS	Bahamas
BH	Bahrain
BD	Bangladesh
BB	Barbados
BY	Belarus
BE	Belgium
BZ	Belize
BJ	Benin
BM	Bermuda
BT	Bhutan
BO	Bolivia
BA	Bosnia and Herzegovina
BW	Botswana
BR	Brazil
IO	British Indian Ocean Territory
VG	British Virgin Islands
BN	Brunei Darussalam
BG	Bulgaria
BF	Burkina Faso
BI	Burundi
KH	Cambodia
CM	Cameroon
CA	Canada
CV	Cape Verde
KY	Cayman Islands
CF	Central African Rep
TD	Chad
CL	Chile
CN	China
CX	Christmas Island
CC	Cocos (Keeling) Islands
CO	Colombia
KM	Comoros

CD	Congo, Democratic Rep of
CG	Congo, Republic of
CK	Cook Islands
CR	Costa Rica
CI	Cote d'Ivoire
HR	Croatia
CU	Cuba
CY	Cyprus
CZ	Czech Rep
DK	Denmark
DJ	Djibouti
DM	Dominica
DO	Dominican Rep
EC	Ecuador
EG	Egypt
SV	El Salvador
GQ	Equatorial Guinea
ER	Eritrea
EE	Estonia
ET	Ethiopia
FK	Falkland Islands
FO	Faroe Islands
FJ	Fiji
FI	Finland
FR	France
GF	French Guyana
PF	French Polynesia
GA	Gabon
GM	Gambia
GE	Georgia
DE	Germany
GH	Ghana
GI	Gibraltar
GR	Greece
GL	Greenland
GD	Grenada
GP	Guadeloupe
GU	Guam
GT	Guatemala
GN	Guinea
GW	Guinea-Bissau
GY	Guyana
HT	Haiti
HN	Honduras
HK	Hong Kong
HU	Hungary
IS	Iceland
IN	India
ID	Indonesia
IR	Iran

IQ	Iraq
IE	Ireland
IT	Italy
JM	Jamaica
JP	Japan
JO	Jordan
KZ	Kazakhstan
KE	Kenya
KI	Kiribati
KP	Korea, North
KR	Korea, South
KW	Kuwait
KG	Kyrgyzstan
LA	Laos
LV	Latvia
LB	Lebanon
LS	Lesotho
LR	Liberia
LY	Libya
LI	Liechtenstein
LT	Lithuania
LU	Luxembourg
MO	Macau
MK	Macedonia
MG	Madagascar
MW	Malawi
MY	Malaysia
MV	Maldives
ML	Mali
MT	Malta
MH	Marshall Islands
MQ	Martinique
MR	Mauritania
MU	Mauritius
YT	Mayotte
MX	Mexico
FM	Micronesia
MD	Moldova, Rep of
MC	Monaco
MN	Mongolia
ME	Montenegro
MS	Montserrat
MA	Morocco
MZ	Mozambique
MM	Myanmar
NA	Namibia
NR	Nauru
NP	Nepal
NL	Netherlands
AN	Netherlands Antilles

NC	New Caledonia
NZ	New Zealand
NI	Nicaragua
NE	Niger
NG	Nigeria
NU	Niue
NF	Norfolk Island
MP	Northern Mariana Islands
NO	Norway
OM	Oman
PK	Pakistan
PW	Palau
PS	Palestinian Territory, Occupied
PA	Panama
PG	Papua New Guinea
PY	Paraguay
PE	Peru
PH	Philippines
PN	Pitcairn Islands
PL	Poland
PT	Portugal
PR	Puerto Rico
QA	Qatar
RE	Reunion
RO	Romania
RU	Russian Federation
RW	Rwanda
WS	Samoa
SM	San Marino
ST	Sao Tome and Principe
SA	Saudi Arabia
SN	Senegal
RS	Serbia
SC	Seychelles
SL	Sierra Leone
SG	Singapore
SK	Slovakia
SI	Slovenia
SB	Solomon Islands
SO	Somalia
ZA	South Africa
ES	Spain
LK	Sri Lanka
SH	St Helena
KN	St Kitts and Nevis
LC	St Lucia
PM	St Pierre and Miquelon
VC	St Vincent and Grenadines
SD	Sudan
SR	Suriname

SZ	Swaziland
SE	Sweden
CH	Switzerland
SY	Syria
TJ	Tajikistan
TW	Taiwan, Rep of China
TZ	Tanzania
TH	Thailand
TL	Timor-Leste
TG	Togo
TK	Tokelau
TO	Tonga
TT	Trinidad and Tobago
TN	Tunisia
TR	Turkey
TM	Turkmenistan
TC	Turks and Caicos Islands
TV	Tuvalu
UG	Uganda
UA	Ukraine
AE	United Arab Emirates
GB	United Kingdom
VI	United States Virgin Islands
US	United States of America
UY	Uruguay
UZ	Uzbekistan
VU	Vanuatu
VA	Vatican City
VE	Venezuela
VN	Viet Nam
WF	Wallis and Futuna Islands
EH	Western Sahara
YE	Yemen
ZM	Zambia
ZW	Zimbabwe

Test Cards

These card numbers can be used when testing your integration to the payment gateway. These cards will not work for live transactions.

Card number	Type	CVV	MPI
4000 0000 0000 0002	Visa	123	No
4111 1111 1111 1111	Visa	123	Yes
4444 3333 2222 1111	Visa	123	Yes
4444 4244 4444 4440	Visa	123	Yes
4444 4444 4444 4448	Visa	123	Yes
4012 8888 8888 1881	Visa	123	Yes
5105 1051 0510 5100	Mastercard	123	No
5454 5454 5454 5454	Mastercard	123	Yes
5555 5555 5555 4444	Mastercard	123	Yes
5555 5555 5555 5557	Mastercard	123	Yes
5581 5822 2222 2229	Mastercard	123	Yes
5641 8209 0009 7002	Maestro UK	123	Yes
6767 0957 4000 0005	Solo	123	No
3434 343434 34343	American Express	1234	No
3566 0020 2014 0006	JCB	123	No

When these card numbers are used with a transaction class that requires the card security data (such as e-commerce transactions) then you should use the value 123 for all cards except for American Express which requires 1234.

For e-commerce transactions, cards which show 'Yes' in the MPI column will use a simulated 3D Secure authentication page, allowing you to test the transaction flow when Verified by Visa or MasterCard SecureCode is used.

Simulating decline/error responses

When it test mode, and when using the above test cards, you can simulate any of the transaction response codes by using specific values for the card security code (CVV). By taking the response code you want to simulate, pad that code with a leading '0' in order to make it a 3 digit code and use that for the CVV.

For example, to simulate the Insufficient Funds response (status 'D', code '41') use 041 as the CVV.

You can also simulate an on-hold transaction in the same way. On hold is where the transaction has been authorised, but the anti-fraud system has flagged the transaction for inspection. Whilst the transaction is on-hold, no funds will be debited from the customers' card. You would need to use the Merchant Administration System to either accept or reject the transaction. To simulate the on-hold response within the test system, use a CVV value of '999' with one of the above test cards.