Telr | innovate payments

Mobile API Integration Guide

Online Payment Processing for Businesses Worldwide

www.telr.com

www.innovatepayments.com

Contents

About this guide

This guide describes the specifications of the mobile API for Innovate Payments. The intended audience is the merchant's technical staff or the merchant's system integrator.

Copyright

Introduction

The mobile API allows transactions to be processed from within a mobile App. With this system, the App collects both order and payment details and then communicates the relevant payment details on a per order basis with payment gateway for processing.

The benefits of using the mobile API for the merchant include being able to retain full control over the payment process.

However, for this system to work successfully, it is important that the merchant ensures the App uses secure methods for handling of the card details, must destroy any card data as soon as the authorisation process has completed, must not store or log that information anywhere, and must not transmit the information to any other system.

You will need to request the mobile API integration for your account; this is not enabled by default.

As part of the setup process you will be supplied with an authentication key.

If your App will be collecting the card details (rather than using the hosted payment pages) then it will be required to be certified though PA DSS. This includes, but is not limited to, the following:

- Card details MUST be destroyed as soon as the authorisation process has completed.
- Card details MUST NOT be stored or logged in any way.
- Card details MUST NOT be transmitted to any other system.

Security

To protect the authentication key that you must use as part of each transaction request, you should not store this key within the actual App code. Other details, such as your Store ID, can be held within the App code.

We would advise that your App obtains the authentication key by downloading it from your servers as part of either an installation or registration phase. This process must be over a secure encrypted link (for example, HTTPS). Once the App has the authentication key, it should store it in the device in a secure method, such as using the iOS Keychain.

This method also allows you to update the authentication key should this be needed.

Request methods

All payment requests must be sent as an XML request using the HTTP POST method to the following URL:

https://secure.innovatepayments.com/gateway/mobile.xml

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

XML Request layout

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<store>Store ID</store>
	<key>Authentication Key</key>                          (Note 1)
	<device>
		<type>Mobile device type</type>                    (Note 2)
		<id>Mobile device ID</id>                          (Note 3)
		<agent>WebView user agent header</agent>           (Note 4)
		<accept>WebView accept header</accept>             (Note 4)
	</device>
	<app>
		<name>Application name</name>
		<version>Application version</version>
		<user>Application user ID</user>                   (Note 5)
		<id>Application installation ID</id>
	</app>
	<tran>
		<test>Test mode</test>                             (Note 6)
		<type>Transaction type</type>
		<class>Transaction class</class>
		<cartid>Transaction cart ID</cartid>               (Note 7)
		<description>Transaction description</description>
		<currency>Transaction currency</currency>          (Note 8)
		<amount>Transaction amount</amount>                (Note 9)
		<ref>Previous transaction reference</ref>          (Note 10)
	</tran>
	<card>
		<number>Card number</number>
		<expiry>
			<month>Expiry date - month</month>             (Note 11)
			<year>Expiry date - year</year>
		</expiry>
		<cvv>CVV</cvv>                                     (Note 12)
	</card>
	<billing>                                              (Note 13)
		<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>                     (Note 14)
			<zip>Zip/Area/Postcode</zip>
		</address>
		<email>Email address</email>
	</billing>
</mobile>
```


Notes:

1. The Authentication Key will be supplied by Innovate Payments as part of the Mobile API setup process after you request that this integration type is enabled for your account. This should not be stored permanently within the App.
2. Description of the device the App is running on.
3. Unique ID for the mobile device (such as identifierForVendor in iOS)
4. The User-Agent and Accept headers that will be used in any webview based requests. These are required as part of E-Commerce class transactions. If no value is sent for these fields, then the gateway will use the headers that are sent as part of the mobile API request. See the section on 3-D Secure for more details.
5. Your reference for the customer/user that is running the App. This should relate to their account within your systems.
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. An example use of the cart ID field would be your own transaction or order reference.
8. 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 currency of the original transaction.
9. 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.
10. The previous transaction reference is required for any continuous authority transaction. It must contain the reference that was supplied in the response for the original transaction.
11. Card dates must be sent as a numeric values with a two digit month (01-12) and a 4 digit year (e.g. 2010)
12. 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

13. Customer Forename, Surname, Address Line 1, City, Country and Email Address are the minimum required details for a transaction to be processed.
14. Country must be sent as a 2 character ISO code. A list of country codes can be found at the end of this document.

** Although the CVV is optional within the Innovate Payments gateway for Mo/To class transactions, some acquiring banks may mandate this. Please check with Innovate Payments support to confirm if this is required for your account.*

Transaction types

The transaction type and class parameters set how the transaction that will be processed. These set both the processing action and the processing category. Some transaction types (such as void and refund) are not applicable for use via the Mobile API, you should use either the Merchant Administration System or the Remote API for these. The options are:

`Type`	
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	*Not available within the Mobile API*
release	*Not available within the Mobile API*
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	*Not available within the Mobile API*
void	*Not available within the Mobile API*
verify	Confirm that the card details given are valid. No funds are reserved or taken from the card.
paypage	Use the hosted payment page to capture and process the card details. See the Payment Page section in this guide for more details.
`Class`	
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 you may be required to perform 3D Secure authentication for the card holder. If this is required, then you will have to direct the customer to a website operated by their card issuer. Until this process is completed, the transaction cannot be authorised.

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.

Authorisation response

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<auth>
		<status>Authorisation status</status>
		<code>Authorisation code</code>
		<message>Authorisation message</message>
		<tranref>Transaction reference</tranref>
		<cvv>Result of the CVV check</cvv>
		<avs>Result of the AVS check</avs>
		<cardcode>Card type code</cardcode>
		<cardlast4>Last 4 digits of the card used</cardlast4>
	</auth>
</mobile>
```

Field	Description
status	Authorisation status. A indicates an authorised transaction. H also indicates an authorised transaction, but where the transaction has been placed on hold. Any other value indicates that the request could not be processed.
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.
message	The authorisation or processing error message.
tranref	The payment gateway transaction reference allocated to this request.
cvv	Result of the CVV check: Y = CVV matched OK N = CVV not matched X = CVV not checked E = Error, unable to check CVV
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
cardcode	Code to indicate the card type used in the transaction. See the code list at the end of the document for a list of card codes.
cardlast4	The last 4 digits of the card number used in the transaction. This is supplied for all payment types (including the Hosted Payment Page method) except for PayPal.

The AVS check is currently only available for cards issued in the United Kingdom, United States of America or Canada, and only if your acquirer supports this service.

For transactions which are not authorised, it is recommended to display a generic message explaining that the transaction could not be processed rather than any specific error text.

Request verification

A request verification URL can be configured as part of this API. If this is set, then when the API receives a transaction request, the details (not including the card information) will be set to the verification URL in order to confirm that the request is valid.

The transaction details will be sent as an XML request, directly from the payment gateway to the merchants systems. The verification request will not be sent via the mobile device.

The verification request will contain the transaction details received by the mobile API.

Verification message from gateway to merchant systems:

```
<?xml version="1.0" encoding="UTF-8"?>
<verify>
	<store>Store ID</store>
	<key>Verification Key</key>
	<device>
		<type>Mobile device type</type>
		<id>Mobile device ID</id>
	</device>
	<app>
		<name>Application name</name>
		<version>Application version</version>
		<user>Application user ID</user>
		<id>Application installation ID</id>
	</app>
	<tran>
		<test>Test mode</test>
		<type>Transaction type</type>
		<class>Transaction class</class>
		<cartid>Transaction cart ID</cartid>
		<description>Transaction description</description>
		<currency>Transaction currency</currency>
		<amount>Transaction amount</amount>
		<ref>Previous transaction reference</ref>
	</tran>
</verify>
```

This provides the opportunity to either block the request, or allow it to continue for authorisation.

The verification response should consist simply of a Yes or No status

```
<?xml version="1.0" encoding="UTF-8"?>
<verify>
	<allow>Yes or No</allow>
</verify>
```

You should ensure that the verify process completes as quickly as possible. The transaction cannot be processed until this has been done. If the request to the verification URL cannot be completed (such as an error occurring whilst processing the request), the gateway will assume that verification has failed and will block the transaction request.

Update billing details

The verification response can also contain the billing details for the customer. This allows the initial mobile API request to omit the billing section. In this case, the verification response should be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<verify>
	<allow>Yes or No</allow>
	<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>
	</billing>
</verify>
```

If the <billing> section is provided, then any billing details that were sent as part of the initial request will be replaced with the details here. These details must include all of the required information.

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, the 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.

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 transaction level tokenization by default. In this way, only the initial transaction requires the card details. In all other transactions the card details do not form part of the transaction request. Instead, the request must include the payment gateway transaction reference from the initial transaction. All required details, such as card number and expiry date, will be retrieved from the transaction record held by gateway.

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. 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.

Using 3-D Secure with the mobile API integration

When processing an e-commerce class transaction, if 3-D Secure authentication is required, then the response to the transaction request will contain details of the URL that the customer must be directed to in order to complete the transaction. This is known as a webview response rather than an authorisation response.

If no 3-D Secure authentication is required, then the response will be a standard authorisation response, and there are no additional steps to be done.

The display of the 3-D Secure page is controlled by the card issuer, the styles or layouts used cannot be changed.

During the initial check to see if any 3-D Secure authentication is needed, details of the browser that will be used must be provided. This can be passed in as part of the transaction request in the <agent> and <accept> parts of the <device> section.

These must represent the User-Agent and Accept headers that would be used by any browser based process (webview) – these are often different to the values that are sent using XML requests.

The header values are passed to the 3-D Secure system at the time then authentication requirements are checked. If authentication is required, then the user must be taken to the web page required by their card issuer. The authentication can be rejected in the browser headers presented at that stage are not the same as the headers used when initially checking for 3-D Secure.

If the headers that will be used by the browser are the same as those that are used when sending the mobile API requests, these fields can be left blank.

WebView response

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<webview>
		<start>URL the customer must be directed to</start>
		<close>URL that marks the end of the process</close>
		<abort>URL that marks an error during the process</abort>
		<code>Transaction code used in completing the request</code>
	</webview>
</mobile>
```

When a webview response is received, the App will need to direct the customer to the URL given as the start address. The App should monitor the progress of the webview, and once it reaches the URL provided as the close address, it should close the web display and continue with the transaction process.

To complete the transaction, the App must now make a second request to the gateway. This request includes will trigger the final authorisation stage of the transaction, and return the authorisation response. The request must be sent to:

https://secure.innovatepayments.com/gateway/mobile_complete.xml

XML Request Layout for transaction completion:

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<store>Store ID</store>
	<key>Authentication Key</key>
	<complete>Transaction code obtained in the WebView response</complete>
</mobile>
```

If the App detects an error during the webview process, or the webview reaches the URL given as the abort address, then it should close the web display and send an abort request to the gateway at the following URL:

https://secure.innovatepayments.com/gateway/mobile_complete.xml

XML Request Layout for transaction abort:

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<store>Store ID</store>
	<key>Authentication Key</key>
	<abort>Transaction code obtained in the WebView response</abort>
</mobile>
```


Using the Hosted Payment Pages

You can generate a transaction request which will use the Hosted Payment Pages to capture the customer's card details, rather than having them entered directly into your App.

This removes any PA DSS certification requirements for your application, as it is no longer handling any card data. Transactions processed in this way can still be used as the base for any future continuous-authority transactions if required. The Hosted Payment Page method will automatically deal with any 3-D Secure pages without any further action required from the App.

To generate a Hosted Payment Page request, set the transaction type field to 'PayPage'. You must not include any card details (card number, expiry or cvv). Where possible, you should provide as much information about the customer as you can (name and address) as this will simplify the payment process for the customer. If these details are not supplied, then the Hosted Payment Page will prompt for them. The customers email address must be supplied.

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<store>Store ID</store>
	<key>Authentication Key</key>
	<device>
		... See XML Response Layout for details ...
	</device>
	<app>
		... See XML Response Layout for details ...
	</app>
	<tran>
		<test>Test mode</test>
		<type>PAYPAGE</type>
		<cartid>Transaction cart ID</cartid>		(Note 1)
		<description>Transaction description</description>
		<currency>Transaction currency</currency>
		<amount>Transaction amount</amount>
	</tran>
	<billing>
		... See XML Response Layout for details ...
	</billing>
</mobile>
```

If the request can be processed, a webview response will be generated. This will direct the customer to the Hosted Payment Pages. The webview process should be monitored as described in the WebView response section above, including sending the final transaction completion request through the mobile API.

Notes:

1. The Hosted Payment Pages require a unique identifier for each transaction. You must use a unique value for the cart ID for each transaction request.
2. The customers email address is a mandatory field as part of the request, but other customer billing fields (name and address) can be left blank if they are not known. However, it is recommended to provide these if possible, as that simplifies the payment process for the customer.
3. No server-to-server callback is made for transactions processed via the Hosted Payment Pages in this way, as the transaction is not server generated.

PayPal transactions

PayPal transactions can be processed through the Mobile API by sending an E-Commerce class Sale request, with the card number set to 'PayPal'. No other card data fields are needed, and the billing fields are also not required (the billing details will be supplied by PayPal when the transaction is completed)

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile>
	<store>Store ID</store>
	<key>Authentication Key</key>
	<device>
		... See XML Response Layout for details ...
	</device>
	<app>
		... See XML Response Layout for details ...
	</app>
	<tran>
		<test>Test mode</test>
		<type>SALE</type>
		<class>ECOM</class>
		<cartid>Transaction cart ID</cartid>		(Note 1)
		<description>Transaction description</description>
		<currency>Transaction currency</currency>	(Note 2)
		<amount>Transaction amount</amount>
	</tran>
	<card>
		<number>PayPal</number>
	</card>
</mobile>
```

If the PayPal request can be processed, a webview response will be generated. This will direct the customer to the PayPal Express Checkout pages. The webview process should be monitored as described in the WebView response section above, including sending the final transaction completion request through the mobile API.

Please see our 'Payment Pages: PayPal Integration' guide for details on how to configure your PayPal account, how to add your PayPal details to the gateway, and how to configure any exchange rates. Thought the mobile API does not make use of the hosted payment pages for this, the steps required to configure your PayPal account and add the details to the gateway are the same.

Notes:

1. PayPal require a unique identifier for each transaction. You must use a unique value for the cart ID for each transaction request.
2. The transaction currency should match your PayPal account currency, though you can use other currencies if you have configured exchange rate details for them within the payment gateway.

Device Identification

The <device> and <app> parts of the XML request are used to help identify the device, application and user. Other than the Application name and Application Version fields, these must not be hard-coded to pre-set values within the App - they must be determined from the actual device the App is installed on.

For each transaction request a reference is created using these idenfication values. For any Continuous-Authority requests (recurring payments), the reference must match the reference generated for the initial transaction. If it does not match, then the request will be declined.

WebView browser headers

Transaction requests where the App captured the card details and where that request is using the E-Commerce transaction class, must provide the User Agent and Accept headers that will be used within any subsequent WebView request.

These headers are transmitted to the 3-D Secure system as part of the card enrolment check. Should 3-D Secure verification be required, the issuers systems may compare any headers received from within the WebView at that stage with the headers used during the enrolment check. If these do not match, some card issuers may reject the authentication request, which would cause the transaction to decline.

The methods used when sending XML requests to the gateway often use different headers to those within a WebView session, as such you should ensure that you obtain the correct headers before sending the request. In some systems this may only be possible by creating a WebView element, intercepting the headers from the http request it was going to send out, and closing the WebView before it actually makes any communication. This can be done by attaching event handlers to the appropriate stage of the WebView. This check should be done either at application startup, or just prior to making a transaction request that will require these headers. It should not be done at application installation time, as the browser details used within the WebView may be updated after the application is installed.

Location Information

Where possible, you should also provide the current location of the user as provided by the location services with the device. These can be supplied using the following fields:

```
<mobile>
          ...
       <location>
             <lat>Current latitude</lat>
             <long>Current longitude</long>
       </location>
          ...
</mobile>
```

This data is not used as part of the transaction processing, but a record of the data is kept with the transaction. This can be of use should a customer dispute a transaction, providing additional details of where the device was when the transaction was processed.

Authorisation response codes

Status	Code	Message
A	*Set by issuer*	Authorised
H	*Set by issuer*	Authorised, but placed on hold. Manual inspection required
P	00	Pending *Transaction is still being processed.*
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

Status	Code	Message
D	47	3DSecure authentication rejected
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
E	58	Error connecting to PayPal
E	59	Request aborted
E	60	Verification failed
D	61	Not authorised
D	62	Not authorised
D	63	Address verification (AVS) mismatch
D	64	Card security code (CVV) and address (AVS) mismatch
D	65	Card is not enabled for e-commerce
D	66	Card cancelled
D	67	Transaction not permitted by issuer
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 mobile API without first requesting that it is added to your account. **You must first request that the mobile API 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.

Card Type Codes

These codes are part of the authorisation response.

VI	Visa
VC	Visa Credit
VD	Visa Debit
VE	Visa Electron
VP	Visa Purchasing
VB	Visa Corporate
MS	MasterCard
MC	MasterCard Credit
MD	MasterCard Debit
MA	Maestro UK
MU	Maestro
AX	American Express
AM	American Express
AP	American Express Purchasing
DN	Diners
JC	JCB
DS	Discover
PP	PayPal
XX	Unknown

Supported Currency Codes

AED	United Arab Emirates Dirham
BHD	Bahraini Dinar
CAD	Canadian Dollar
EUR	Euro
IDR	Indonesian Rupiah
GBP	Pound Sterling
JOD	Jordanian Dinar
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.

Document history

Release	Changes
1.03	Added Hosted Payment Page option for capturing card details. Additional information regarding device identification, the requirements for the browser User-Agent and Accept headers, and optional location information.
1.02	Added PayPal transaction method. Ability to set customer billing details within the request verification response.
1.01	Added abort URL to webview response. For E-Commerce class transactions, the browser User-Agent and Accept headers must be provided as these may be required by the 3-D Secure authentication system.
1.00	Initial release

