Application Programming Interface (API)

Online Payment Processing for Businesses Worldwide

Contents

About this guide

This guide describes the Application Programming Interface that is available as part of the Innovate Payments Gateway. This guide is intended for use by developers who have knowledge of web based services and of XML.

Copyright

Introduction

The Innovate Payments API makes it easy to integrate your applications with the merchant administration system. Important data, such as transaction details, can be retrieved as required. The API responds with XML formatted data.

API configuration

Access to the API is controlled using an API Key. This key control what services can be used, and from what location they can be accessed.

To add a new key you need to use the Merchant Administration System, select the 'API' menu, and click on the 'Add new entry' option.

You can now configure which services will be available using this key, and can limit where this key can be used from by setting one or more approved IP addresses.

Whilst a key can be configured to allow access from any location, we strongly advise that you set IP address limits. The IP details provided would, for example, be the IP details of your servers that require access to the API. You can supply multiple IP addresses by separating each one with a comma. As requests are made over the internet, the IP details given must be the publically routable IP details and not internal/private IP space.

Add new API entry

Access details				
Merchant ID:	1234			
API Key:	01eba1beM7Jp4sRgJ3qGXM$J			
IP Limit:	91.75.72.164, 91.75.72.165			
Services				
Transactions				
☑ Recent transactions	☐ Transaction details	☐ Linked transactions	☐ Search by cart ID	☐ Search by email
☐ Search by card number				
Reports				
☐ Recent reports				
Repeat billing agreements				
☐ Recent agreements	☐ Agreement details	☐ Agreement history	☐ Cancel agreement	

[Update]

Authentication

Each request made to the API must include authentication details. The API uses the HTTP Basic Access Authentication method.

Basic access authentication

Basic access authentication is a method for a HTTP user agent to provide a user name and password when making a request.

HTTP Basic authentication implementation is one of the easiest methods to use as it doesn't require cookies, session handling, or a login sequence. HTTP Basic authentication simply uses static headers.

The user agent sends the authentication credentials by using the Authorization header. The header is constructed as follows:

- Username and password are combined into a string "username:password"
- The resulting string is then encoded using Base64
- The authorization method and a space i.e. "Basic " is then put before the encoded string.

For example, if the user agent uses 'Aladdin' as the username and 'open sesame' as the password then the header is formed as follows:

```
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

API username and password

When accessing the Innovate Payments API, the username part of the authentication string is your merchant ID, and the password is the API key.

Requests

Requests should be made using the GET method unless otherwise stated for the given service, and must include the Authorization header.

If the Authorization header is not present, or is not valid, then a '403 Forbidden' response will be generated.

Responses

Response data is provided in XML format. All data is encoded using the UTF-8 character set.

Requesting an unknown service will return a '404 Not Found' response. If the service requested is a known service, but the supplied Authentication details do not have access to that service then the response will be '403 Forbidden'

Common data elements

Many of the responses contain common data elements; these are described in the following pages.

Transaction type

This contains details of the transaction type. It includes both a text based name for the type, and a numeric type code.

Data:

```
<type>
  <name>Transaction type name</name>
  <code>Transaction type code</code>
</type>
```

The transaction types are as follows:

Name	Code	Details
Sale	1	A standard online transaction. If authorised the relevant amount will be debited from the customers' card.
Void	2	A method used to cancel a sale transaction. This can only be done within a short time after the sale transaction was processed. The void amount is always the same as the original sale amount.
Refund	3	Used to credit an amount back to the customer. The refund amount may be the same as or less than the original Sale amount.
Refund Reversal	4	This can be used to cancel a refund, and can only be done within a short time after the refund was processed. Not all acquirers support this transaction type.
Auth	5	A pre-authorisation transaction. This is similar to a sale, but the amount is not immediately debited from the customers' card. Instead the amount is reserved on that card, and can then be debited at a later date using a capture transaction. Until the capture transaction is processed, no funds will be debited.
Release	6	Used to release any funds that have previously been reserved by an auth transaction. No funds are debited from the card, and the capture option can no longer be used.
Capture	7	Used to debit funds that have been previously reserved using an auth transaction. This completes the transaction processing.
Capture Reversal	8	This can be used to cancel a capture transaction, and can only be done within a short time after the capture was processed. Not all acquirers support this transaction type.

Transaction class

This contains details of the transaction class. It includes both a text based name and a numeric code.

Data:

```
<class>
  <name>Transaction class name</name>
  <code>Transaction class code</code>
</class>
```

The transaction classes are as follows:

Name	Code
Mail Order/Telephone Order	1
E-Commerce	2
Continuous Authority	4

Authorisation results

Details of the transaction status. If the transaction was authorised, then the status will be 'A'. If it has been authorised but placed on hold, the status will be 'H'. Other status codes indicate that the transaction could not be processed. A list of transaction status codes can be found at the end of this document.

Data:

```
<auth>
  <status>Authorisation status</status>
  <code>Authorisation code</code>
  <message>Authorisation message</message>
</auth>
```

Customer information

This contains the name and address details of the customer.

Data:

```
<billing>
  <title>Customer title</title>
  <fname>Customer forename(s)</fname>
  <sname>Customer surname</sname>
  <addr1>Address line 1</addr1>
  <addr2>Address line 1</addr2>
  <addr3>Address line 1</addr3>
  <city>City</city>
  <region>Region</region>
  <country>Country – full name</country>
  <country_iso>Country – 2 character ISO code</country_iso>
  <zip>Post/Area code</zip>
  <tel>Telephone number</tel>
  <email>Email address</email>
</billing>
```

Transaction information

Individual transaction details.

Data:

```
<transaction>
  <id>Transaction reference</id>
  <prev_id>Transaction reference of the previous transaction</prev_id>
  <init_id>Transaction reference of the initial transaction</init_id>
  <type>CDE: Transaction type</type>
  <class>CDE: Transaction class</class>
  <auth>CDE: Authorisation results</auth>
  <amount>Transaction amount</amount>
  <currency>Transaction currency</currency>
  <description>Transaction description</description>
  <cartid>Cart ID</cartid>
  <test>Test mode. 0=Live, 1=Test</test>
  <date>Transaction time and date, GMT</date>
  <store>
    <id>Store ID</id>
    <name>Store name</name>
  </store>
  <billing>CDE: Customer information</billing>
  <card>
    <number>Card number. Only the first and last 4 digits are shown</number>
    <type>Card type, if known</type>
    <country_iso>Card issue country, if known</country_iso>
  </card>
  <notecount>Number of notes attached to this transaction</notecount>
  <notes>
    <note>        -- repeated for each note attached to this transaction
      <text>Note text</text>
      <ref>Additional transaction related to this note, if any</ref>
    </note>
  </notes>
</transaction>
```

For transactions which are a follow-up transaction, such as a void, the <prev_id> contains the transaction reference of the transaction being voided. For an initial transaction, such as a sale or auth, if it is set then it will be the same as the main transaction ID.

In a multi-transaction sequence, such as a refund being performed on a capture transaction, the <init_id> will contain the reference to the initial auth transaction.

Repeat billing agreement

Details of an individual repeat billing agreement

Data:

```
<agreement>
  <id>Agreement ID </id>
  <store>
    <id>Store ID</id>
    <name>Store name</name>
  </store>
  <created>Time and date the agreement was created, GMT</created>
  <status>Agreement status, numeric version</status>
  <statustxt>Agreement status, text version</statustxt>
  <currency>Agreement currency</currency>
  <initial>Initial transaction amount. 0 if no initial transaction.</initial>
  <recurring>
    <period>Repeat billing period – weekly or monthly</period>
    <interval>Number of repeat billing periods between each transaction</interval>
    <day>Which day of the week or day of the month each transaction is on</day>
    <count>Maximum number of repeat transaction. 0 for no limit</count>
    <amount>Recurring transaction amount.</amount>
  </recurring>
  <final>Final transaction amount for agreements with repeat limit.</final>
  <description>Agreement description</description>
  <transaction>
    <description>Description to use for each transaction.</description>
    <cartid>Cart ID to use for each transaction</cartid>
    <test>Test mode. 0=Live, 1=Test</test>
  </transaction>
  <custid>Customer ID</custid>
  <billing>CDE: Customer information</billing> (Optional, depends on service used)
</agreement>
```

The agreement status codes are as follows:

Name	Code	Notes
Pending	0	Agreement is being setup and has not yet started
Running	1	Agreement is running normally
Completed	2	Agreement has completed (where a limit is set on the number of payments)
Failed	3	Agreement stopped due to multiple payment errors.
Overdue	4	Payment overdue. Only for agreements where payment is by invoice.
Cancelled	5	Agreement has been cancelled.

Services

The following pages detail each of the services available. For each service, the required URL will be shown, and an example of the data provided by that service. Unless specified otherwise, the request method for each service must be GET

Where the response data includes any of the common data elements, refer to the relevant section above for the details of that element. These will be marked in the response examples by the prefix CDE followed by the element title.

Recent transaction list

https://secure.innovatepayments.com/tools/api/xml/transaction

This will provide a list of the most recent transactions processed within the last 48 hours. A maximum of 30 transactions will be included.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
  <trancount>Count of transactions in the results</trancount>
  <transaction>CDE: Transaction information</transaction>
</transactions>
```

The `<transaction>...</transaction>` section will be repeated for each transaction in the list.

Transaction details

https://secure.innovatepayments.com/tools/api/xml/transaction/*{tranref}*

This will provide the details for a given transaction reference. The transaction reference must be included within the URL, where it shows {tranref} above.

For example, the details for transaction reference 011287290362 would be obtained from https://secure.innovatepayments.com/tools/api/xml/transaction/011287290362

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction>CDE: Transaction information</transaction>
```

Linked transactions

https://secure.innovatepayments.com/tools/api/xml/transaction/*{tranref}*/linked

This will provide the details for all transaction linked to the given reference. The transaction reference must be included within the URL, where it shows {tranref} above.

For example, the details for transactions linked to reference 011287290362 would be obtained from https://secure.innovatepayments.com/tools/api/xml/transaction/011287290362/linked

This allows you to find the complete list of events for any given transaction including the initial auth or sale, and any subsequent capture, refund or void transactions. The initial transaction reference for the specified transaction is obtained, and then all transactions which have the same initial transaction reference are returned. A maximum of 30 transactions will be included.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
  <trancount>Count of transactions in the results</trancount>
  <transaction>CDE: Transaction information</transaction>
</transactions>
```

The `<transaction>...</transaction>` section will be repeated for each transaction in the list.

Transactions search – Cart ID

https://secure.innovatepayments.com/tools/api/xml/transaction/*{ref}*/cart

This will provide the details for all transaction that have the given reference as their cart ID. The reference must be included within the URL, where it shows {ref} above.

For example, the details for transactions with cart ID ABC123 would be obtained from https://secure.innovatepayments.com/tools/api/xml/transaction/ABC123/cart

This will return all matching transactions, including the initial auth or sale, and any subsequent capture, refund or void transactions. This is most effective if the cart ID used in the initial transaction request is a unique ID for the transaction. A maximum of 30 transactions will be included.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
  <trancount>Count of transactions in the results</trancount>
  <transaction>CDE: Transaction information</transaction>
</transactions>
```

The `<transaction>...</transaction>` section will be repeated for each transaction in the list.

Transactions search – Email

https://secure.innovatepayments.com/tools/api/xml/transaction/*{addr}*/email

This will provide the details for all transaction that have the given address as their billing email address. The address must be included within the URL, where it shows {addr} above.

For example, the details for transactions with email user@domain.com would be obtained from https://secure.innovatepayments.com/tools/api/xml/transaction/user@domain.com/email

This will return all matching transactions, including the initial auth or sale, and any subsequent capture, refund or void transactions. A maximum of 30 transactions will be included.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
  <trancount>Count of transactions in the results</trancount>
  <transaction>CDE: Transaction information</transaction>
</transactions>
```

The `<transaction>...</transaction>` section will be repeated for each transaction in the list.

Transactions search – Card number

https://secure.innovatepayments.com/tools/api/xml/transaction/*{num}*/card

This will provide the details for all transaction that where made with the given card number. The number must be included within the URL, where it shows {num} above.

The card number can be provided in two ways, as either an 8 digit value (containing the first 4 and last 4 digits of the card number) or as a 12 digit transaction reference.

Examples:

Details for transactions with card starting 4000 and ending 0002:
　　　https://secure.innovatepayments.com/tools/api/xml/transaction/40000002/card

Details for transactions made use the same card as transaction 011287290362:
　　　https://secure.innovatepayments.com/tools/api/xml/transaction/011287290362/card

This will return all matching transactions, including the initial auth or sale, and any subsequent capture, refund or void transactions. A maximum of 30 transactions will be included.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<transactions>
  <trancount>Count of transactions in the results</trancount>
  <transaction>CDE: Transaction information</transaction>
</transactions>
```

The `<transaction>...</transaction>` section will be repeated for each transaction in the list.

Recent reports

https://secure.innovatepayments.com/tools/api/xml/report

This will provide a list of the most recent reports that have been generated, such as the Daily and Monthly transaction reports. A maximum of 50 reports will be included.

Data provided:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<reports>
  <repcount>Count of reports in the results</repcount>
  <report>
    <id>Report ID</id>
    <store>Store ID</store>
    <date>Time and date report was generated, GMT</date>
    <file1>Filename of the first report file</file1>
    <file2>Filename of the second report file, if any</file2>
    <description>Description of the report</description>
    <comments>Additional information about the report</comments>
  </report>
</reports>
```

The `<report>...</report>` section will be repeated for each report in the list.

Recent repeat billing agreements list

https://secure.innovatepayments.com/tools/api/xml/agreement

This will provide a list of the most recent repeat billing agreements. A maximum of 30 agreements will be included.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<agreements>
  <agrcount>Count of agreements in the results</agrcount>
  <agreement>CDE: Repeat billing agreement</agreement> (No billing details)
</agreements>
```

The `<agreement>...</agreement>` section will be repeated for each agreement in the list.

Repeat billing agreement details

https://secure.innovatepayments.com/tools/api/xml/agreement/*{agreement_id}*

This will provide the details for a given agreement ID. The agreement ID must be included within the URL, where it shows {agreement_id} above.

For example, the details for agreement 60100 would be obtained from
https://secure.innovatepayments.com/tools/api/xml/agreement/60100

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<agreement>CDE: Repeat billing agreement</agreement> (Includes billing details)
```

Repeat billing agreement history

https://secure.innovatepayments.com/tools/api/xml/agreement/*{agreement_id}*/history

This will provide the event history for a given agreement ID. The agreement ID must be included within the URL, where it shows {agreement_id} above.

For example, the history for agreement 60100 would be obtained from
https://secure.innovatepayments.com/tools/api/xml/agreement/60100/history

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<agreement>
  <eventcount>Count of events in the results</eventcount>
  <event>
    <type>Event type</type>
    <typetxt>Event type, text description</typetxt>
    <date>Time and date of event, GMT</date>
    <note>Event note text</note> (*)
    <paytype>Payment type</paytype> (*)
    <paytypetxt>Payment type, text description</paytypetxt> (*)
    <paycount>Payment number for regular payments</paycount> (*)
    <amount>Transaction amount</amount> (*)
    <tranref>Transaction reference</tranref> (*)
    <invoice>Invoice reference</invoice> (*)
  </event>
</agreement>
```

The `<event>...</event>` section will be repeated for each event in the list.

Fields marked with `(*)` are optional, and will only be present if relevant to that event.

Event type	Description
1	Agreement created
2	Agreement completed
3	Agreement failed (to many declined transactions)
4	Invoice late for payment
5	Agreement cancelled
6	Invoice sent
7	Payment authorised
8	Payment declined
9	Card verified
10	Credit authorised
11	Credit declined
12	Text note added

Payment type	Description
1	Initial payment
2	Regular payment
3	Final payment
4	Manual adjustment (e.g. additional payment or a refund)

Cancel repeat billing agreement

https://secure.innovatepayments.com/tools/api/xml/agreement/*{agreement_id}*

A request sent using the DELETE method will cancel the agreement. The agreement ID must be included within the URL, where it shows {agreement_id} above.

For example, the to cancel agreement 60100 send a DELETE method request to https://secure.innovatepayments.com/tools/api/xml/agreement/60100

This option will provide the same format response as for the agreement details request, showing the updated agreement status.

Data provided:

```
<?xml version="1.0" encoding="UTF-8"?>
<agreement>CDE: Repeat billing agreement</agreement> (Includes billing details)
```

Request method types

URL	GET	PUT	POST	DELETE
transaction	✔	✘	✘	✘
transaction/{tranref}	✔	✘	✘	✘
transaction/{tranref}/linked	✔	✘	✘	✘
transaction/{ref}/cart	✔	✘	✘	✘
transaction/{addr}/email	✔	✘	✘	✘
transaction/{num}/card	✔	✘	✘	✘
report	✔	✘	✘	✘
agreement	✔	✘	✘	✘
agreement/{agreement_id}	✔	✘	✘	✔
agreement/{agreement_id}/history	✔	✘	✘	✘

Transaction status 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
E	58	Error connecting to PayPal
D	80	Not authorised *Card Filter module. Message text can be changed.*
D	90	Not authorised
D	91	Not authorised
D	92	Not authorised
E	98	Internal system error
E	99	Unknown error

Document history

Release	Changes
1.03	Added details of repeat billing services
1.02	Added details of the transaction search services (cart ID, email, card number)
1.01	Added details of the Linked transactions service
1.00	Initial release.