API Integration Guide

V 1.7

IP Payments Pty Ltd
Level 3, 441 Kent Street
Sydney
NSW 2000
Australia
(ABN 86 095 635 680)

T +61 2 9255 9500 F
+61 2 8248 1276
www.ippayments.co
m

Table of Contents

1 About this document

1.1 Document history

Version	Date Modified	Author	Summary of Changes
V1.0	01/05/2014	Anne Kehoe	Document created based on API specification.
V1.1	10/09/2014	Anne Kehoe	Content update.
V1.2	31/10/2014	Anne Kehoe	Addition of void request.
V1.3	27/05/2015	Anne Kehoe	Addition of NZ bank account details.
V1.4	20/07/2015	Celine Wang	Added additional data to query request
V1.5	12/01/2015	Celine Wang	Updated to Query API
V1.6	27/06/2016	Celine Wang	Updated to Production URL

1.2 Definitions

The following terms and abbreviations are used in this document:

Term	Description
IPP	IP Payments, a premium payments solutions provider uniquely skilled in providing high-quality, efficient and customised solutions to corporate organisations in all industry sectors.
Merchant	For the purposes of this document your company will be referred to as the 'merchant'. A person or company involved in wholesale trade, supplying goods or services to a business or consumer market.
Acquiring Bank	An acquiring bank (or acquirer) is a bank or financial institution that processes credit or debit card payments on behalf of a merchant.
CR	A change request is an issue, defect or new requirement which is raised by a business person and/or representative of a local affiliate which is not described or described in a different way in the latest version of the SDS document (+ amendments).
CC	Creditcard.
Security Code (CVV2/CSC2/CCV)	The Security Code is a 3 or 4 digit code on the back of the cardholder's card. This is used to verify the customer is in possession of the card.
PAN	Primary Account Number (Credit Card Number).
PCI-DSS	Payment Card Industry Data Security Standard. PCI-DSS is an information security standard for organizations that handle cardholder information for the major debit, credit, prepaid, e-purse, ATM, and POS cards.
Access Portal	Access Portal is the platform used by IPP to implement our hosted payment applications – HPP and iHPP.
HPP	The Hosted Payment Page is a standalone payment page which is not integrated into an application.

iHPP	The integrated Hosted Payment Page is integrated into the merchant's website dynamically accepting transaction data prior to the customer entering their card details. Notification of the transaction result is sent back to the merchant in realtime.
iFrame	Inline Frame, a HTML tag used to embed another document within an existing HTML document. Specifically used in this document to describe how to embed the IPP payment page in the merchants website.
DL	Direct Link, this is the value used to specify which iHPP template is to be used. Required where more than one iHPP exists for a particular client account.
HTML	Hypertext Markup Language
POST	A method for sending HTML form data over the Internet. Post data is encoded within the message body.
GET	A method for sending HTML form data over the Internet. Get data is encoded by a browser into the URL.
CSS	Cascading Style Sheets is a style sheet language used for describing the presentation of a web page.
SST	Secure Session Token
URL	Uniform Resource Locator
WSDL	Web Services Description Language
PRM	Payment Relationship Manager, IPP's transaction reporting tool used for user administration, viewing transaction history, refunding and downloading reports among other functionality.
CSV	CSV meaning Comma Separated Values is a report format which can be downloaded from our reporting tool, PRM.
API	Application Programming Interface, a merchant can use IPP's API to gain access to the features and data of our services and applications.
SOAP	*Simple Object Access Protocol, a protocol specification for exchanging structured information in the implementation of web services.*
XML	eXtensible Markup Language, defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
SIPP	Statement of Invoice Presentment and Payment
TCP/IP	Transmission Control Protocol/Internet Protocol
Tokenisation	Storage of the customer's card data against a unique reference called a token in IPP's secure PCI-DSS compliant system for future use in recurring or one-click payments. This removes the need for the merchant to store card data minimising their PCI DSS scope. This is an additional service and must be enabled on your account.
Token	The unique reference that the customers card data is stored against in IPP's secure PCI-DSS compliant system.
MOD10 Check	A simple algorithm used to validate a credit card number.

2 Purpose of this document

The purpose of this document is to describe the requirements and functionality of the API solution that will be implemented for you by IP Payments Pty Ltd ("IPP").

This document also outlines what is involved for you, the merchant, to integrate into the IPP API.

If this document does not meet your needs, please contact us to discuss your requirements and the bespoke solutions we can offer.

3 Introduction

IP Payments is a premium payments solutions provider, uniquely skilled in providing high-quality, efficient, reliable and customised solutions to corporate organisations in all industry sectors. We develop and manage web based billing, payment and reconciliation services for some of the most recognised brand names in the world.

API

IPP offer our merchants the ability to securely and efficiently process online, real-time transactions via an API. The API web service accepts and processes SOAP requests from a remote location over TCP/IP. Transaction results are returned in real-time via the API.

3.1 PCI-DSS Compliance

IPP adhere to the highest standard of PCI-DSS compliance - Level 1. It can be a common misconception that if you are processing online payments with a secure payment gateway you are automatically PCI-DSS compliant. This is not the case, PCI-DSS applies to all organisations which store, process or transmit cardholder information. How you manage your payment process will define the level of PCI-DSS that you must adhere to.

Please note, using IPP's API to capture and transmit credit card data will bring your website, development process and internal systems in to scope for PCI-DSS.

IPP offer solutions that will allow you to capture and transmit credit card data without adding additional PCIDSS scope to your business. Please contact IPP for further information on these products and services.

4 Process Overview

The following process outlines how you can use IPP's API to carry out transaction requests to debit a customer for a single payment.

4.1 System access

IP Payments supports a test and live web service, each of which can be found at the following locations:

- The test web service URL is located at: https://demo.ippayments.com.au/interface/api/dts.asmx
- The live web service URL is located at: https://www.ippayments.com/interface/api/dts.asmx

At these locations, you will also find sample SOAP requests and responses, as well as the WSDL.

The Secure Remote API web service employs the following business logic for processing online transactions:

- A remote application connects and authenticates with IPP's server.

- The transaction data is passed to the API in a SOAP request with a number of parameters required for processing.
- IPP carry out the action required for the API request called and transactional data is logged to the IPP database.
- The API responds with result data to the remote application.

The above process occurs for each API call.

4.2 SOAP and XML Construction

IPP use the SOAP protocol to exchange structured XML messages for transaction processing. Payment request and response XML transactions are submitted as a parameter in the SOAP message.

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values. The <trnXML> field will hold the transaction XML that will advise IPP of what action to take.

```
POST /interface/api/dts.asmx HTTP/1.1
Host: www.ippayments.com.au
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: IPP web service URL

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SOAP METHOD xmlns="http://www.ippayments.com.au/interface/api/dts">
      <trnXML>string</trnXML>
    </SubmitSinglePayment>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length



<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SubmitSinglePaymentResponse xmlns="http://www.ippayments.com.au/interface/api/dts">
      <SubmitSinglePaymentResult>string</SubmitSinglePaymentResult>
    </SubmitSinglePaymentResponse>
  </soap:Body>
</soap:Envelope>
```

The transaction messages discussed in the rest of this document will define the XML which must be submitted in the SOAP message. Please note that XML tags within each API parameter are case sensitive.

Please see an example of the transaction XML below. You must ensure you are passing the transaction XML as a parameter in the SOAP message. If your application is not handling this correctly, you can achieve this through the use of the CDATA tag as shown in the below example, the additional CDATA tags are highlighted in bold red.

Please note that this is valid for all API functions throughout this document.

```
<![CDATA[

    <Transaction>
        <CustNumber>12345678</CustNumber>
        <CustRef>123456</CustRef>
        <Amount>5500</Amount>
        <TrnType>1</TrnType>
        <CreditCard Registered="False">
            <CardNumber>4005550000000001</CardNumber>
            <ExpM>05</ExpM>
            <ExpY>2013</ExpY>
            <CVN>123</CVN>
            <CardHolderName>John Smith</CardHolderName>
        </CreditCard>
        <Security>
            <UserName>username</UserName>
            <Password>password</Password>
        </Security>
        <TrnSource>192.168.0.1</TrnSource>
    </Transaction>

]]>
```

4.3 Security

Transactions are processed via an industry standard secure https connection. This means the merchant can be assured when the transaction is in transit:

1. Sensitive data such as credit card numbers will remain confidential because all information is 128bit encrypted; and
2. The server the merchant is connecting to will be authenticated as belonging to IP Payments through PKI certificates issued by a root Certificate Authority.

In addition to the above security, each merchant transaction request received by IP Payments is authenticated via a pre-allocated User Id and Password, both 128bit encrypted.

4.4 Transaction Types

The following list shows the available transaction types when processing a single payment.

Transaction Types

TRNTYPE	NAME	DESCRIPTION
1	Credit Card – Purchase (Default)	Process a real time purchase transaction when transaction is submitted.
2	Credit Card – Auth	Process a real time pre-auth transaction when transaction is submitted.
5	Refund	Process a refund transaction when refund request is submitted.
7	Direct Entry – Debit	Process a direct debit transaction when transaction is submitted.
8	Direct Entry – Credit	Process a direct credit transaction when transaction is submitted.

The following transaction types can be submitted to IP Payments for reporting purposes only. No action will be taken to process a payment on these transaction types however the data will be reported on in IPP's reporting tool, PRM. If you would like further information on these transaction types, please contact us.

- Cash
- EFT
- EFT – POS
- Cheque
- Money Order
- BPay
- Post Bill Pay

5 Payment Submission

5.1 Submit Single Payment

The following process outlines how you can use IPP's API to carry out a transaction to debit a customer for a single payment.

5.1.1 Submit Single Payment Request

SOAP Method: SubmitSinglePayment

The list below provides an overview of the available transaction elements that should be submitted in the XML request, all of which are stored by IP Payments.

SubmitSinglePayment – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**

Transaction/AccountNumber	16	Alpha Numeric	123456789	O	This value dictates which account the transaction will be processed through. If this value us not populated, the transaction will be processed to the account tied to the username field.
Transaction/CustRef	64	Alpha Numeric	Invoice1234	M	A mandatory reference for the transaction sent by you for reporting purposes.
Transaction/CustNumber	64	Alpha Numeric	12345678	O	Merchant assigned unique customer number can be used for token based processing.
Transaction/Amount	10	Numeric	5500	M	Amount entered as an integer eg. $55.00 = 5500.
Transaction/TrnType	2	Numeric	1	O	The transaction type. Please see section Transaction Types for valid values.
Transaction/CreditCard	N/A	N/A	N/A	M	This is an XML node only. Please note, either the CreditCard node or the DirectEntry node will be submitted in a transaction depending on the customer's payment method. This node has one attribute, 'Registered'. This attribute is used for recurring payments where you have previously registered the customer's details. This is set to 'True' if a previously registered credit card is to be used or or 'False' if processing a payment for an unregistered customer / payment method.
					For single submission payments this attribute should be set to 'False'.
Transaction/CreditCard/CardNumber	16	Numeric	4005550000000001	O	Customer credit card number.
Transaction/CreditCard/ExpM	2	Numeric	05	O	Expiry month of customer credit card.
Transaction/CreditCard/ExpY	4	Numeric	2013	O	Expiry year of customer credit card.
Transaction/CreditCard/CardHolderName	32	Alpha Numeric	John Smith	O	Name as it appears on credit card.

Transaction/CreditCard/CVN	4	Numeric	123	O	CVN (Security Code) of the customer credit card.
Transaction/DirectEntry	N/A	N/A	N/A	M	This is an XML node only. Please note, either the CreditCard node or the DirectEntry node will be submitted in a transaction depending on the customer's payment method. This node has one attribute, 'Registered'. This attribute is used for recurring payments where you have previously registered the customer's details. This is set to 'True' if a previously registered payment method is to be used or 'False' if processing a payment for an unregistered customer / payment method. For single submission payments this attribute should be set to 'False'.
Transaction/DirectEntry/AccNo	16	Numeric	098124564	O	AU Bank account number. For NZ bank details: The NZ Bank Account and Suffix should be included in this field with no spaces.
Transaction/DirectEntry/AccRouting	16	Numeric	013259	O	Account routing number (BSB in Australia). For NZ bank details: The NZ Bank and Branch codes should be included in this field with no spaces.
Transaction/DirectEntry/AccTitle	32	Alpha Numeric	John Smith	O	Title of bank account for registration.
Transaction/Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name.
Transaction/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password.
Transaction/UserDefined	N/A	N/A	N/A	O	This is an XML node only. User defined elements can be included in the transaction request and used for reporting purposes at a later stage.

Transaction/UserDefined/Reference1	32	Text	YourReference1	O	User defined elements can be included in the transaction request and used for reporting purposes at a later stage.
Transaction/UserDefined/Reference2	64	Text	YourReference2	O	User defined elements can be included in the transaction request and used for reporting purposes at a later stage.
Transaction/UserDefined/Reference3	128	Text	YourReference3	O	User defined elements can be included in the transaction request and used for reporting purposes at a later stage.
Transaction/UserDefined/Reference4	1024	Text	YourReference4	O	User defined elements can be included in the transaction request and used for reporting purposes at a later stage.
Transaction/UserDefined/Reference5	1024	Text	YourReference5	O	User defined elements can be included in the transaction request and used for reporting purposes at a later stage.
Transaction/TrnSource	16	*Alpha Numeric*	192.168.0.1	O	Internet Protocol Address of the payment source. If this is left blank, the system will use the IP of the SOAP client server.

5.1.1.1 Example XML for single creditcard purchase

```
<Transaction>
    <CustNumber>12345678</CustNumber>
    <CustRef>123456</CustRef>
    <Amount>5500</Amount>
    <TrnType>1</TrnType>
    <CreditCard Registered="False">
        <CardNumber>4005550000000001</CardNumber>
        <ExpM>05</ExpM>
        <ExpY>2013</ExpY>
        <CVN>123</CVN>
        <CardHolderName>John Smith</CardHolderName>
    </CreditCard>
    <Security>
        <UserName>username</UserName>
        <Password>password</Password>
    </Security>
    <UserDefined>
        <Reference1>Ref1234</Reference1>
    </UserDefined>
    <TrnSource>192.168.0.1</TrnSource>
</Transaction>
```

5.1.1.2 Example XML of single Direct Debit purchase

```
<Transaction>
        <CustNumber>12345678</CustNumber>
        <CustRef>123456</CustRef>
        <Amount>5500</Amount>
        <TrnType>7</TrnType>
        <DirectEntry Registered="False">
                <AccNo>098124564</AccNo>
                <AccRouting>013259</AccRouting>
                <AccTitle>John Smith</AccTitle>
        </DirectEntry>
        <Security>
              <UserName>username</UserName>
              <Password>password</Password>
           </Security>
        <UserDefined>
              <Reference1>Ref1234</Reference1>
        </UserDefined>
        <TrnSource>192.168.0.1</TrnSource>
</Transaction>
```

5.1.1.3 Example XML of pre-auth transaction

A pre-auth transaction reserves the funds on customer's card without debiting the money from the customer's card. A follow up capture request **must** be sent with this transaction type to settle the transaction, debit the customers card and receive the funds.

```
<Transaction>
        <CustNumber>12345678</CustNumber>
        <CustRef>123456</CustRef>
        <Amount>5500</Amount>
        <TrnType>2</TrnType>
        <CreditCard Registered="False">
                <CardNumber>4005550000000001</CardNumber>
                <ExpM>05</ExpM>
                <ExpY>2013</ExpY>
                <CVN>123</CVN>
                <CardHolderName>John Smith</CardHolderName>
        </CreditCard>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
        <TrnSource>192.168.0.1</TrnSource>
</Transaction>
```

This can be useful for example:

- Where the merchant requires the ability to check if items are in stock prior to debiting the customers card.
- Or where the merchant wants to reserve funds on a customer's card but carry out fraud checks prior to accepting the order and debiting the card.

After the pre-auth has been completed and if you want to proceed with the transaction, then you must send a capture request to notify IP Payments to initiate the settlement process. Please see section Payment Capture for details on how to capture the funds reserved by a pre-auth.

5.1.2 Submit Single Payment Response

The following list provides an overview of available response elements.

SubmitSinglePayment Response – XML Element Description			
XML ELEMENT	**FORMAT**	**SAMPLE**	**DESCRIPTION**
Response/ResponseCode	Numeric	0	0 = Approved 1 = Not Approved
Response/Timestamp	Alpha Numeric	23-Sep-2011 15:33:25	Timestamp is the date and time that IP Payments has received the transaction request. The format is dd-MMM-yyyy hh:mm:ss
Response/Receipt	Alpha Numeric	10001197	Receipt is a unique reference for this transaction issued by IP Payments. Please note the receipt number will be required for any follow up transactions such as capture or refund requests.
Response/SettlementDate	Alpha Numeric	23-Sep-2011	SettlementDate is the date on which approved funds are settled by the acquiring bank/financial institution. The format is dd-MMM-yyyy
Response/DeclinedCode	Numeric	5	DeclinedCode value is present only when ResponseCode is 1. Please see the declined code list in Appendix 2 for all possible values.
Response/DeclinedMessage	Alpha Numeric	Do Not Honour	DeclinedMessage value is present only when ResponseCode is 1. This contains a textual description of the declined code.

5.1.2.1 Example XML of single payment response

```
<Response>
          <ResponseCode>0</ResponseCode>
         <Timestamp>02-Jul-2014 11:07:08</Timestamp>
         <Receipt>10001197</Receipt>
          <SettlementDate>02-Jul-2014</SettlementDate>
          <DeclinedCode></DeclinedCode>
          <DeclinedMessage></DeclinedMessage>
</Response>
```

5.2 Payment Capture

A pre-auth transaction reserves the funds on customer's card without debiting the money from the customer's card. A follow up capture request **must** be sent with this transaction type to settle the transaction, debit the customers card and receive the funds. Please see section Example XML of pre-auth transaction for full details on how to submit a pre-auth.

5.2.1 Submit Single Capture Request

SOAP Method: SubmitSingleCapture

The list below provides an overview of the available transaction elements that should be submitted in the XML request. The receipt number provided in the original payment response will be required to process this transaction.

SubmitSingleCapture – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**
Capture/Receipt	8	Numeric	12345678	M	Receipt number issued by IP Payments from originally processed authorisation, provided in the authorisation response.
Capture/Amount	10	Numeric	5500	M	Amount entered as an integer eg. $55.00 = 5500
Capture/Security /UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name
Capture /Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password

5.2.1.1 Example XML of Single Capture Request

Please note this is a follow up transaction which will capture the funds reserved by the pre-auth transaction.

Please see example request XML to capture a payment below.

```
<Capture>
        <Receipt>123456</Receipt>
        <Amount>5500</Amount>
        <Security>
                <UserName>api.user.client</UserName>
                <Password>pw123sec</Password>
        </Security>
</Capture>
```

5.2.2 Submit Single Capture Response

5.2.2.1 Example XML of Single Capture Response

```
<Response>
          <ResponseCode>0</ResponseCode>
         <Timestamp>02-Jul-2014 11:07:08</Timestamp>
         <Receipt>88918350</Receipt>
          <SettlementDate>02-Jul-2014</SettlementDate>
          <DeclinedCode></DeclinedCode>
          <DeclinedMessage></DeclinedMessage>
</Response>
```

5.3 Void Transaction

IPP provide the ability to process a void transaction request to stop the funds being taking from a customer's card. The void transaction must be sent prior to the settlement time of the transaction. The settlement time can vary depending on the acquiring bank you are processing through, please confirm with IPP the settlement time for your acquiring bank.
Once the transaction has been settled, a refund transaction must be sent to fund the money back to the customer's card.

5.3.1 Submit Single Void Request

SOAP Method: SubmitSingleVOID

The list below provides an overview of the available transaction elements that should be submitted in the XML request.

SubmitSingleVOID – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**
Void/Receipt	8	Numeric	12345678	M	Receipt from originally processed purchase/capture transaction.
Void /Amount	10	Numeric	5500	M	Amount entered as an integer eg. $55.00 = 5500.
Void /Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name.
Void/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password.

5.3.1.1 Example XML of Single Void Request

Please see example XML to void a payment below.

```
<Void>
        <Receipt>123456</Receipt>
        <Amount>5500</Amount>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</Void>
```

5.3.2 Submit Single Void Response

5.3.2.1 Example XML of Single Void Response

```
<Response>
        <ResponseCode>0</ResponseCode>
        <SettlementDate>02-Jul-2014</SettlementDate>
        <DeclinedCode></DeclinedCode>
        <DeclinedMessage></DeclinedMessage>
</Response>
```

5.4 Refund Transaction

IPP offer the ability to process a refund to the customer's card which was used in the original payment. You can refund up to 100% of the originally processed amount. The receipt number provided in the original payment response will be required to process this transaction.

IPP also offer the ability to process an orphan refund i.e. a refund transaction which is not associated with an original transaction. For further information on this request, please contact IPP.

5.4.1 Submit Single Refund Request

SOAP Method: SubmitSingleRefund

The list below provides an overview of the available transaction elements that should be submitted in the XML request.

SubmitSingleRefund – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**

Refund/Receipt	8	Numeric	12345678	M	Receipt from originally processed purchase/capture transaction.
Refund/Amount	10	Numeric	5500	M	Amount entered as an integer eg. $55.00 = 5500. You can process up to 100% of the originally processed amount.
Refund/Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name.
Refund/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password.

5.4.1.1 Example XML of Single Refund Request

Please see example XML to refund a payment below.

```
<Refund>
        <Receipt>123456</Receipt>
        <Amount>5500</Amount>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</Refund>
```

5.4.2 Submit Single Refund Response

5.4.2.1 Example XML of Single Refund Response

```
<Response>
        <ResponseCode>0</ResponseCode>
        <Timestamp>02-Jul-2014 11:07:08</Timestamp>
        <Receipt>88918350</Receipt>
        <SettlementDate>02-Jul-2014</SettlementDate>
        <DeclinedCode></DeclinedCode>
        <DeclinedMessage></DeclinedMessage>
</Response>
```

5.5 Query Transaction

IPP allow merchants to query the status of a transaction at any time, this API call is useful if the initial transaction request times out and the merchant is unaware of the transaction status. Multiple matches can be returned.

5.5.1 Submit Single Query Request

SOAP Method: QueryTransaction

The following list provides an overview of available query transaction elements. The query transaction will return all transaction results that match the data submitted in the transaction elements. Submit as many elements as possible to ensure a unique response is returned.

QueryTransaction – XML Element Description					
ELEMENT NAME	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**
QueryTransaction/Criteria/AccountNumber	16	Alpha Numeric	5487ABC	O	This value dictates which account the transaction will be queried. If this value us not populated, the transaction will be processed to the account tied to the username field.
QueryTransaction/Criteria/TrnStartTimestamp	19	Alpha Numeric	2015NOV-01 00:00:01	M	The start date/time for the transaction search. Format yyyy-MMM-dd HH:MM:SS
QueryTransaction/Criteria/TrnEndTimestamp	19	Alpha Numeric	2015MAR-01 23:59:59	M	The end date/time for the transaction search. Format yyyy-MMM-dd HH:MM:SS
QueryTransaction/Criteria/CustRef	64	Alpha Numeric	Invoice1234	O	Merchant assigned unique reference value.
QueryTransaction/Criteria/CustNumber	64	Numeric	12345678	O	Merchant assigned unique customer number.
QueryTransaction/Criteria/Amount	10	Numeric	5500	O	Amount entered as an integer eg. $55.00 = 5500.
QueryTransaction/Criteria/Receipt	10	Numeric	12345678	O	Search for a specific transaction in the time frame via a previously provided receipt number.
QueryTransaction/Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name.
QueryTransaction/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password.
QueryTransaction/Criteria/TrnType	2	Numeric	1	O	The transaction type. Please see section Transaction Types for valid values.
QueryTransaction/Criteria/CardNumberPrefix	4	Numeric	4005	O	First four digits of credit card number.

QueryTransaction/Criteria/CardNumberSuffix	3	Numeric	001	O	Last three digits of credit card number.
QueryTransaction/Criteria/ExpM	2	Numeric	05	O	Expiry month of customer credit card.
QueryTransaction/Criteria/ExpY	4	Numeric	2013	O	Expiry year of customer credit card.
QueryTransaction/Criteria/CardHolderName	64	Alpha Numeric	John Smith	O	Name as it appears on credit card.
QueryTransaction/Criteria/AccNo	16	Numeric	098124564	O	Bank account number.
QueryTransaction/Criteria/AccRouting	16	Numeric	013-259	O	Account routing number (BSB in Australia).
AdditionalData *Additional fields only required if you wish to receive the elements back.*					
QueryTransaction/AdditionalData/Core/Amount	6	Alpha	Amount	O	Amount passed in the query request.
QueryTransaction/AdditionalData/Core/CustRef	7	Alpha	CustRef	O	Additional transactional reference passed in the query request.
QueryTransaction/AdditionalData/Core/CustNumber	10	Alpha	CustNumber	O	Merchant assigned unique customer number passed in the query request.
QueryTransaction/AdditionalData/Core/TrnTypeID	9	Alpha	TrnTypeID	O	Merchant assigned unique customer number passed in the query request.
QueryTransaction/AdditionalData/Core/TrnStatusID	11	Alpha	TrnStatusID	O	Merchant assigned unique customer number passed in the query request.

5.5.1.1 Example XML of Single Query Request

Please see example XML to query a payment below.

```
<QueryTransaction>
    <Criteria>
        <AccountNumber>12345678</AccountNumber>
        <TrnStartTimestamp>2015-02-23 00:00:00</TrnStartTimestamp>
        <TrnEndTimestamp>2015-02-24 23:59:59</TrnEndTimestamp>
        <CustRef>SS123456</CustRef>
        <CustNumber>890</CustNumber>
```

```
                <Amount>2900</Amount>
        </Criteria>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</QueryTransaction>
```

5.5.1.2 Example XML of Single Query Request with AdditionalData

Please see example XML to query a payment below.

```
<QueryTransaction>
        <Criteria>
                <AccountNumber>12345678</AccountNumber>
                <TrnStartTimestamp>2015-02-23 00:00:00</TrnStartTimestamp>
                <TrnEndTimestamp>2015-02-24 23:59:59</TrnEndTimestamp>
                <CustRef>SS123456</CustRef>
                <CustNumber>890</rCustNumber>
                <Amount>2900</Amount>
                </Criteria>
        <AdditionalData>
                <Core>Amount</ Core >
                < Core>CustRef</ Core >
                < Core> CustNumber </ Core >
                <Core>TrnTypeID</Core>
                <Core>TrnStatusID</Core>
        </AdditionalData>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</QueryTransaction>
```

5.5.2 Submit Single Query Response

The following list provides an overview of available query transaction response elements.

QueryTransaction Response – XML Element Description			
XML ELEMENT	**FORMAT**	**SAMPLE**	**DESCRIPTION**

QueryResponse	XML Root Node Only.	N/A	This node has three attributes: 'MatchingCount' - the number of active payment schedules found. This count will determine how many 'QueryResponse/Response' elements there are – one for each returned transaction. 'MatchingReturned' – This is the number of matches actually returned – with a maximum of 1000. 'Error' – will contain one of the following values:
			0 - Transactions retrieved successfully 1 - Invalid username/password 2 - Invalid Account Number 3 - Invalid Cust Number 99 - Exception encountered Please note that issues with dates will return an exception '99'
QueryResponse/Response/ResponseCode	Numeric	0	0 = Approved 1 = Not Approved
QueryResponse/Response/Timestamp	Alpha Numeric	23-Sep-2011 15:33:25	Timestamp is the date and time that IP Payments has received the transaction request. The format is dd-MMM-yyyyhh:mm:ss
QueryResponse/Response/Receipt	Alpha Numeric	10001197	Receipt is a unique reference for this transaction issued by IP Payments.
QueryResponse/Response/SettlementDate	Alpha Numeric	23-Sep-2011	SettlementDate is the date on which approved funds are settled by the acquiring bank/financial institution. The format is ddMMM-yyyy
QueryResponse/Response/DeclinedCode	Numeric	5	DeclinedCode value is present only when ResponseCode is 1. Please see the declined code list in Appendix 2 for all possible values.
QueryResponse/Response/DeclinedMessage	Alpha Numeric	Do Not Honour	DeclinedMessage value is present only when ResponseCode is 1. This contains a textual description of the declined code.
AdditionalData *Additional fields only returned if you have*			
QueryResponse/Response/Amount	Numeric	5500	Amount passed in the Single Query Request.
QueryResponse/Response/CustRef	Alpha Numeric	Invoice1234	Additional transactional reference passed the Single Query Request.
QueryResponse/Response/CustNumber	Numeric	12345678	Unique customer number passed in the Single Query Request.

QueryResponse/Response/TrnTypeID	Numeric	48	TrnTypeID associated to the found transaction.
QueryResponse/Response/TrnStatusID	Numeric	1	TrnStatusID associated to the found transaction.

5.5.2.1 Example XML of Single Query Response

Please see example Query XML Response below.

```
<QueryResponseMatchingCount="2" MatchingReturned=”2” Error="0">
    <Response>
        <ResponseCode>0</ResponseCode>
        <Timestamp>2015-02-24 21:18:47</Timestamp>
        <Receipt>10001197</Receipt>
        <SettlementDate>2015-02-23</SettlementDate>
        <DeclinedCode></DeclinedCode>
        <DeclinedMessage></DeclinedMessage>
    </Response>
    <Response>
        <ResponseCode>0</ResponseCode>
        <Timestamp>2015-02-24 21:22:01</Timestamp>
        <Receipt>10001199</Receipt>
        <SettlementDate>2015-02-23</SettlementDate>
        <DeclinedCode></DeclinedCode>
        <DeclinedMessage></DeclinedMessage>
    </Response>
</QueryResponse>
```

5.5.2.2 Example XML of Single Query Response with AdditionalData

```
<QueryResponseMatchingCount="2" MatchingReturned=”2” Error="0">
        <Response>
                <ResponseCode>0</ResponseCode>
                <Timestamp>2015-02-24 21:18:47</Timestamp>
                <Receipt>10001197</Receipt>
                <SettlementDate>2015-02-23</SettlementDate>
                <DeclinedCode></DeclinedCode>
                <DeclinedMessage></DeclinedMessage>
                <Amount>2900</Amount>
                <CustNumber>12345678</CustNumber>
                <CustRef>123456</CustRef>
                <TrnTypeID>48</ TrnTypeID >
                <TrnStatusID>1</ TrnStatusID >

        </Response>
        <Response>
                <ResponseCode>0</ResponseCode>
                <Timestamp>2015-02-24 21:22:01</Timestamp>
                <Receipt>10001199</Receipt>
                <SettlementDate>2015-02-23</SettlementDate>
                <DeclinedCode></DeclinedCode>
                <DeclinedMessage></DeclinedMessage>
                <Amount>2900</Amount>
                <CustNumber>12345678</CustNumber>
                <CustRef>123456</CustRef>
                <TrnTypeID>48</ TrnTypeID >
                <TrnStatusID>1</ TrnStatusID >
        </Response>
</QueryResponse>
```

6 Account Hierarchy

IP Payments will set up your account based on brands or divisions in your company, each brand / division will receive an Account Number.

- Different divisions can also be set up for reporting purposes so specific user groups can only access certain division information.
- Each account will be setup with your Acquiring Merchant ID and supporting credentials which will dictate the bank account in which your funds will be settled by your acquirer.
- If you process in multiple currencies, a division will be set up for each currency.

The account number specified in the <AccountNumber> field of your transaction will be used to route the transaction to the correct location in your account.

6.1 Surcharging

IPP provide you with the ability to surcharge your customers based on the card type they enter at the time of payment. This could be used for example if you wish to pass on the cost of payment processing to the customer. The card types supported for surcharging are:

- Visa
- MasterCard
- Amex
- Diners

By default no surcharge will be applied. If you wish to surcharge your customers then you will need to advise IPP prior to account setup, IPP will then set up the surcharge on your account. The following surcharge types can be implemented.

- A fixed amount surcharge by card type.
- A percentage of the payment amount surcharge by card type.

7 Appendix

7.1 Reference Documents

See other guides below which may be useful in implementation of your solution. You can request these guides from IPP.

Document Name	Description
iHPP integration guide	This document details the integration required for IPP's integrated Hosted Payment Page (iHPP), which provides the ability to accept payments and tokenise card data through an IPP hosted page.
Tokenisation integration Guide	This document outlines the integration methods for tokenisation for subsequent token and recurring payments.
PRM User Guide	This document provides a guide to the functionality available in IP Payments reporting tool, Payment Relationship Manager (PRM).

7.2 Bank Response Codes

Code	Response Text	Code	Response Text
APPROVED			
00	Approved	08	Honour with ID
11	Approved VIP (not used)	16	Approved, Update Track 3 (not used)
77	Approved (ANZ only)		
DECLINED			
01	Refer to Card Issuer	41	Lost Card—Pick Up
02	Refer to Issuer's Special Conditions	42	No Universal Amount
03	Invalid Merchant	43	Stolen Card—Pick Up
04	Pick Up Card	44	No Investment Account
05	Do Not Honour	51	Insufficient Funds
06	Error	52	No Cheque Account
07	Pick Up Card, Special Conditions	53	No Savings Account
09	Request in Progress	54	Expired Card
10	Partial Amount Approved	55	Incorrect PIN
12	Invalid Transaction	56	No Card Record
13	Invalid Amount	57	Trans. not Permitted to Cardholder
14	Invalid Card Number	58	Transaction not Permitted to Terminal
15	No Such Issuer	59	Suspected Fraud
17	Customer Cancellation	60	Card Acceptor Contact Acquirer
18	Customer Dispute	61	Exceeds Withdrawal Amount Limits
19	Re-enter Transaction	62	Restricted Card
20	Invalid Response	63	Security Violation
21	No Action Taken	64	Original Amount Incorrect
22	Suspected Malfunction	65	Exceeds Withdrawal Frequency Limit
23	Unacceptable Transaction Fee	66	Card Acceptor Call Acquirer Security
24	File Update not Supported by Receiver	67	Hard Capture—Pick Up Card at ATM
25	Unable to Locate Record on File	68	Response Received Too Late
26	Duplicate File Update Record	75	Allowable PIN Tries Exceeded
27	File Update Field Edit Error	86	ATM Malfunction
28	File Update File Locked Out	87	No Envelope Inserted
29	File Update not Successful	88	Unable to Dispense
30	Format Error	89	Administration Error
31	Bank not Supported by Switch	90	Cut-off in Progress
32	Completed Partially	91	Issuer or Switch is Inoperative
33	Expired Card—Pick Up	92	Financial Institution not Found
34	Suspected Fraud—Pick Up	93	Trans Cannot be Completed
35	Contact Acquirer—Pick Up	94	Duplicate Transmission
36	Restricted Card—Pick Up	95	Reconcile Error
37	Call Acquirer Security—Pick Up	96	System Malfunction
38	Allowable PIN Tries Exceeded	97	Reconciliation Totals Reset

39	No CREDIT Account	98	MAC Error
40	Requested Function not Supported	99	Reserved for National Use

7.3 IP Payments Declined Codes

The following table presents the potential declined codes that may be presented in a declined transaction result:

Code	Response Text
100	System Exception
101	Invalid company identifier
102	Invalid account identifier
103	Invalid API username or password
104	Invalid transaction type identifier
105	Invalid channel identifier
106	Invalid currency identifier
107	Invalid transaction amount
108	No customer identifier supplied
109	No customer reference supplied
110	Invalid credit card number
111	Invalid credit card expiry date
112	Invalid source account number
113	Invalid source account routing number
114	Invalid escrow account number
115	Invalid escrow account routing number
116	Invalid destination account number
117	Invalid destination account routing number
118	Invalid customer identifier
119	Customer status not active
120	Account status not active
121	Account does not have any risk profile rules assigned
122	Registered customer details not found
123	Duplicate in document list
124	CVN required but not supplied
130	<PreviousTrnReceipt> has been supplied, but it requires a credit card capture, refund or cancel transaction type
150	Account not set up to accept supplied currency transactions
151	Account not set up correctly to accept supplied currency transactions
152	Account not set up to accept credit card transactions for the supplied credit card type
153	Account not set up to accept credit card transactions for the supplied amount
154	Merchant account details not set up correctly
155	Interface details not set up correctly
156	Auth transaction not found

157	Auth transaction was declined
158	Capture amount exceeds original auth plus any previous capture total
159	Purchase or Capture transaction was not found
160	Purchase or Capture transaction was declined
161	Refund amount exceeds original purchase or capture plus any previous refund total
162	Risk profile rules failed
170	Account not set up to accept direct entry transactions
171	Account not set up correctly to accept direct entry transactions
180	Exception encountered when retrieving the receipt number
181	Exception encountered when receiving transaction data from client
182	Exception encountered when creating transaction XML log
183	Exception parsing transaction XML
184	Exception validating transaction XML
190	Exception encountered when finding transaction identifier
191	Exception encountered when finding credit card interface to use
192	Exception encountered when submitting transaction to interface
193	Exception encountered when finding direct entry details
200	Interface error
201	Interface Error with successful automatic reversal
300	DE Dishonour - 01 Invalid BSB Number
301	DE Dishonour - 02 Payment Stopped
302	DE Dishonour - 03 Account Closed
303	DE Dishonour - 05 Invalid Account Number
304	DE Dishonour - 06 Refer to Customer
305	DE Dishonour - 07 Challenge Authority to Process
306	DE Dishonour - 09 Technically Invalid
307	DE Dishonour - 04 Account Holder Deceased
308	DE Manually Refunded
400	CC Chargeback - Documentation not Supplied
401	CC Chargeback - Documentation supplied was not legible
402	CC Chargeback - Signature supplied does not match signature on file
403	CC Chargeback - Transaction duplicated
404	CC Chargeback - Goods not delivered
500	Batch Record Exception
600	CC Manually Refunded
700	Invalid Disbursement XML
701	Disbursement XML amount total does not match the transaction amount
702	Account number must be supplied for each disbursement
703	Account number supplied for disbursement does not exist or account is inactive
704	Account number supplied for disbursement appears two or more times
997	Remote Interface Exception
998	Transaction Payment Cancelled
999	Timeout when waiting for a response