Payments Scheduler Integration Guide

V 1.6

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	15/05/2015	Celine Wang	Update to SOAP and XML Construction.
V1.2	15/06/2015	Anne Kehoe	Added CustomerStorageNumber. Added Processing location for Scheduled Payments section.
V1.3	12/12/2015	Celine Wang	Updated to content.
V1.4	15/02/2016	Celine Wang	Updated to Process Overview
V1.5	27/06/2016	Celine Wang	Updated to Production URL
V1.6	21/07/2016	Praneeth Mariada	Update 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.

IPP's payment scheduler provides the ability to set up recurring payments for your customers without the hassle of managing the schedule in your system.

3.1 Integration Methods

3.1.1 API (Web Service)

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. Please see the section API Overview for further information on how to connect to this service.

3.1.1.1 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 such as iHPP and Tokenisation that will allow you to capture and transmit credit card data without adding additional PCI-DSS scope to your business. Please contact IPP for further information on these products and services if you are concerned about your PCIDSS compliance.

4 Process Overview

IPP provide the ability to create payments schedule for your customers enabling you to process recurring (variable or fixed amount) payments in a secure, integrated and efficient manner.

You can set up a payment schedule via PRM or using our API as detailed below. A payment schedule can be set up on a registered customer whose details are stored in the IP Payments system or you can submit the payment details in the schedule request. For further information on customer registration, please see the customer registration integration guide.

Payment Scheduler is extremely flexible in a situation where a payment is scheduled for a date that does not occur in a given month, for example, if a payment is scheduled for the 31st of each month but there is no 31st of the month, we will process the scheduled payment on the 30th of April, June, September, November and the 28th of February (or the 29th if it exists).

Please note the Payment scheduler is compatible with IPP's customer registration service but not tokenisation. For further information, please contact IPP.

4.1 Transaction Types

The following list shows the available transaction types for scheduled payments.

Transaction Types		
TRNTYPE	**NAME**	**DESCRIPTION**
1	Credit Card – Purchase (Default)	Process a purchase transaction when this transaction type is submitted.

7	Direct Entry – Debit	Process a direct debit transaction when transaction is submitted.

4.2 Payment Scheduler via PRM

IPP allow you to set up payment schedules for customers through our online management tool PRM. Please see the PRM user guide for more information on how to enable this functionality.

4.3 Payment Scheduler via API

A payments schedule can be set up for registered customers where you have previously stored the customer's payment details either via PRM or the customer registration API. You can also submit the payment details in the schedule set up request if you have not previously registered the customer. Please see details below on how to set up Payment Schedules using the API.

4.3.1 Processing location for Scheduled Payments

Please note the payment schedule must be set up on the account with which you wish to process the payments on. When submitting the API requests detailed below, you must use a username which is set up on the account where you wish the future payments to be processed.

4.3.2 Submit Payment Schedule

4.3.2.1 SubmitPaymentSchedule Request

SOAP Method: SubmitPaymentSchedule

The following list provides an overview of available transaction elements to submit a payment schedule.

SubmitPaymentSchedule – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**
Schedule/CustomerStorageNumber	16	Alpha Numeric	123456789	O	The account within your account structure where the customers payment details are stored. If this value us not populated, the payment details will be accessed from the account tied to the username field.
Schedule/CustNumber	64	Alpha Numeric	12345678	M	Merchant assigned unique customer number can be used for token based processing.
Schedule/Amount	10	Numeric	5500	M	Amount entered as an integer eg. $55.00 = 5500.

Schedule/TrnType	2	Numeric	1	O	The transaction type. Please see section Transaction Types for valid values.
Schedule/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'.
Schedule/CreditCard/CardNumber	16	Numeric	4005550000000001	O	Customer credit card number.
Schedule/CreditCard/ExpM	2	Numeric	05	O	Expiry month of customer credit card. The expiry date field is normally not required for recurring transactions. AMEX and Diners do require the expiry date to be provided (must be a valid expiry date that was supplied by card issuers).
Schedule/CreditCard/ExpY	4	Numeric	2013	O	Expiry year of customer credit card.
Schedule/CreditCard/CardHolderName	32	Alpha Numeric	John Smith	O	Name as it appears on credit card.

Schedule/DirectEntry					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'.
Schedule/ DirectEntry /AccNo	16	Numeric	098124564	O	Bank account number.
Schedule/ DirectEntry /AccRouting	16	Numeric	013259	O	Account routing number (BSB in Australia).
Schedule/ DirectEntry /AccTitle	32	Alpha Numeric	John Smith	O	Title of bank account for registration.
Schedule/Schedule/Frequ ency	1	Alpha Numeric	S	M	The frequency of the schedule. One of: S - Single Payment W - Weekly F – Fortnightly M – Monthly Q – Quarterly A – Annually

Schedule/Schedule/StartDate	10	Alpha Numeric	2011-09-23	M	The start date of the schedule. The first payment will be processed on this date. For single payments this is the date of the single payment. Format YYYY-MM-DD Please note the Start Date must be in the future, otherwise it will get rejected.
Schedule/Schedule/EndDate	10	Alpha Numeric	2011-09-23	O	If present, no payments will be made on or after this date. If this field is left blank the payment will continue until cancelled. Not applicable for Single payments.
Schedule/Schedule/EndPaymentCount	6	Numeric	12	O	If this parameter is present, no payments will be made once the number of payments submitted (inclusive of declined payments) reaches this value. Not applicable for Single payments.
Schedule/Schedule/NewAmountDate	10	Alpha Numeric	2012-01-23	O	If the recurring amount is to change in the future, then any recurring payments processed on or after this date will use the NewAmount. If this value is present then NewAmount is mandatory. If Schedule/NewAmount is present, Schedule/NewAmountDate needs to be populated.
					Format YYYY-MM-DD

Schedule/Schedule/NewAmount	10	Numeric	5800	O	If the recurring amount is to change in the future, then any recurring payments processed on or after the NewAmountDate will use this NewAmount. If this value is present then NewAmountDate is mandatory. Amount entered as an integer eg. $55.00 = 5500
Schedule/Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name. Note the username must be set up on the account where you wish the scheduled payments to be processed through.
Schedule/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password

4.3.2.1.1 Example XML for Credit Card Payment Schedule

Credit card payment schedule for single payment:

```
<Schedule>
    <CustomerStorageNumber>VAULT1</CustomerStorageNumber>
    <CustNumber>123456</CustNumber>
    <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>
    <Schedule>
        <Frequency>S</Frequency>
        <StartDate>2012-03-01</StartDate>
    </Schedule>
    <Security>
        <UserName>username</UserName>
        <Password>password</Password>
    </Security>
</Schedule>
```

Example XML for Credit Card Payment Schedule – Fortnightly with 24 payments

```
<Schedule>
        <CustNumber>123456</CustNumber>
        <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>
        <Schedule>
                <Frequency>F</Frequency>
                <StartDate>2012-03-01</StartDate>
                <EndPaymentCount>24</EndPaymentCount>
        </Schedule>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</Schedule>
```

Example XML for Credit Card Payment Schedule – Monthly, ending after a specific date

```
<Schedule>
        <CustNumber>123456</CustNumber>
        <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>
        <Schedule>
                <Frequency>M</Frequency>
                <StartDate>2012-03-01</StartDate>
                <EndPaymentDate>2013-12-01</EndPaymentDate>
        </Schedule>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</Schedule>
```

4.3.2.1.2 Example XML for Registered Direct Debit Payment

Example also shows a monthly schedule with 12 payments and changing amount after 6 months

```
<Schedule>
        <CustNumber>123456</CustNumber>
        <Amount>5500</Amount>
        <TrnType>7</TrnType>
        <DirectEntry Registered="True">
        </DirectEntry>
        <Schedule>
                <Frequency>M</Frequency>
                <StartDate>2011-12-01</StartDate >
                <EndPaymentCount>12</EndPaymentCount>
                <NewAmountDate>2012-05-01</NewAmountDate>
                <NewAmount>6500</NewAmount>
        </Schedule>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</Schedule>
```

4.3.2.2 SubmitPaymentSchedule XML Response

The following list provides an overview of available response elements.

SubmitPaymentSchedule Response – XML Element Description			
XML ELEMENT	**FORMAT**	**SAMPLE**	**DESCRIPTION**
Response/ResponseCode	Numeric	0	0 = Accepted 1 = Not Accepted
Response/ScheduleIdentifier	Alpha Numeric	00005689	This uniquely identifies the schedule within IPP. Will only be present if ResponseCode is 0
Response/DeclinedCode	Numeric	2	DeclinedCode value is present only when ResponseCode is 1. Possible values are: 1 - Invalid username/password 2 - Invalid schedule data 99 - Exception encountered
Response/DeclinedMessage	Alpha Numeric	Invalid schedule data	DeclinedMessage value is present only when ResponseCode is 1. This contains a textual description of the declined code.

4.3.2.2.1 Example SubmitPaymentSchedule XML Response

```
<Response>
        <ResponseCode>0</ResponseCode>
        <ScheduleIdentifier>00005689</ScheduleIdentifier>
        <DeclinedCode></DeclinedCode>
        <DeclinedMessage></DeclinedMessage>
</Response>
```

4.3.2.2.2 Example SubmitPaymentSchedule XML Response – invalid data

```
<Response>
        <ResponseCode>1</ResponseCode>
        <ScheduleIdentifier></ScheduleIdentifier>
        <DeclinedCode>2</DeclinedCode>
        <DeclinedMessage>Invalid Schedule data</DeclinedMessage>
</Response>
```

4.3.3 Query PaymentSchedules

4.3.3.1 QueryPaymentSchedules Request

The following list provides an overview of available elements to query payment schedules.

QueryPaymentSchedules – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**
QuerySchedules/AccountNumber	16	Alpha Numeric	123456789	O	Account/Merchant identifier.
QuerySchedules/CustNumber	32	Alpha Numeric	12345678	M	Biller assigned unique customer number, can be used for token based processing
QuerySchedules/Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name
QuerySchedules/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password

Example XML for QueryPaymentSchedules Request

This Method takes one parameter, which is an XML document:

```
<QuerySchedules>
        <CustNumber>123456</CustNumber>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</QuerySchedules>
```

4.3.3.2 QueryPaymentSchedules XML Response

The following list provides an overview of available response elements.

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

QueryResponse	N/A	N/A	XML Root Node Only. This node has two attributes: 'MatchingCount' - the number of active payment schedules found. This count will determine how many 'QueryResponse/Schedule' elements there are – one for each active schedule. 'Error' – will contain one of the following values: 0 - Schedules retrieved successfully 1 - Invalid username/password 2 - Invalid account number 3 - Invalid customer number 99 - Exception encountered
QueryResponse/Schedule/Identifier	Alpha Numeric	00004568	This uniquely identifies the schedule within IPP.
QueryResponse/Schedule/DateAdded	Alpha Numeric	2011-10-23	
QueryResponse/Schedule/Frequency	Alpha Numeric	F	The frequency of the schedule. One of: S - Single Payment W - Weekly F – Fortnightly M – Monthly Q – Quarterly A - Annually
QueryResponse/Schedule/StartDate	Alpha Numeric	2012-01-12	The SubmitPaymentSchedule of the schedule Format yyyy-MM-dd
QueryResponse/Schedule/EndDate	Alpha Numeric	2012-12-12	End date of schedule. Format yyyy-MM-dd Not applicable for Single payments. Can be blank.
QueryResponse/Schedule/EndPaymentCount	Numeric	12	The maximum number of payments that will be processed for this schedule.Not applicable for Single payments. Can be blank.
QueryResponse/Schedule/NewAmountDate	Alpha Numeric	2012-06-12	If the recurring amount is to change in the future, then any recurring payments processed on or after this date will use the NewAmount. If this value is present then NewAmount will be present. Format yyyy-MM-dd
QueryResponse//Schedule/NewAmount	Numeric	6500	If the recurring amount is to change in the future, then any recurring payments processed on or after the NewAmountDate will use this NewAmount. If this value is present then NewAmountDate will be present. Amount entered as an integer eg. $55.00 = 5500
QueryResponse/Schedule/CurrentPaymentCount	Numeric	2	The number of payments processed in this schedule.
QueryResponse/Schedule/NextPaymentDate	Alpha Numeric	2011-10-28	The next payment date in this schedule. Format yyyy-MM-dd

Example QueryPaymentSchedules XML Response with two returned schedules

```
<QueryResponseMatchingCount="2" Error="0">
        <Schedule>
                <Identifier>00004568</Identifier>
                <DateAdded>2011-11-15</DateAdded>
                <Frequency>M</Frequency>
                <StartDate>2012-01-12</StartDate>
                <EndDate>2012-12-12</EndDate>
                <EndPaymentCount></EndPaymentCount>
                <NewAmountDate></NewAmountDate>
                <NewAmount></NewAmount>
                <CurrentPaymentCount>0<CurrentPaymentCount>
                <NextPaymentDate>2012-01-12<NextPaymentDate>
        </Schedule>

        <Schedule>
                <Identifier>00004579</Identifier>
                <DateAdded>2011-11-15</DateAdded>
                <Frequency>S</Frequency>
                <StartDate>2012-01-12</StartDate>
                <EndDate></EndDate>
                <EndPaymentCount></EndPaymentCount>
                <NewAmountDate></NewAmountDate>
                <NewAmount></NewAmount>
                <CurrentPaymentCount>0<CurrentPaymentCount>
                <NextPaymentDate>2012-01-12<NextPaymentDate>
        </Schedule>
</QueryResponse>
```

Example QueryPaymentSchedules XML Response with error

```
<QueryResponseMatchingCount="0" Error="99"></QueryResponse>
```

4.3.4 DeletePaymentSchedule

The following list provides an overview of available transaction elements to delete a payment schedule.

DeletePaymentSchedule – XML Element Description					
XML ELEMENT	**MAX SIZE**	**FORMAT**	**SAMPLE**	**MANDATORY / OPTIONAL**	**DESCRIPTION**
DeleteSchedule/ AccountNumber	16	Alpha Numeric	123456789	O	Merchant identifier.
DeleteSchedule/ CustNumber	32	Alpha Numeric	12345678	M	Merchant assigned unique customer number, can be used for token based processing

DeleteSchedule/Schedule/Identifier	10	Alpha Numeric	00004587	M	The unique identifier of the schedule to delete.
Schedule/Security/UserName	32	Alpha Numeric	Username	M	Pre-allocated merchant API user name
Schedule/Security/Password	16	Alpha Numeric	Password	M	Pre-allocated merchant API password

4.3.4.1 Delete Payment Schedule Request

This Method takes one parameter, which is an XML document:

```
<DeleteSchedule>
        <AccountNumber>123456789</AccountNumber>
        <CustNumber>123456</CustNumber>
        <Schedule>
                <Identifier>00004587</Identifier> </Schedule>
        <Security>
                <UserName>username</UserName>
                <Password>password</Password>
        </Security>
</DeleteSchedule>
```

4.3.4.2 DeletePaymentSchedule Response

The following list provides an overview of available response elements.

DeletePaymentSchedule Response – XML Element Description			
XML ELEMENT	**FORMAT**	**SAMPLE**	**DESCRIPTION**
Response/ResponseCode	Numeric	0	0 = Deleted 1 = Not Deleted
Response/DeclinedCode	Numeric	2	DeclinedCode value is present only when ResponseCode is 1. Possible values are: 1 - Invalid username/password 2 - Invalid account number 3 - Invalid customer number 4 - Invalid schedule Identifier 5 - Schedule is not active 99 - Exception encountered
Response/DeclinedMessage	Alpha Numeric	Invalid schedule identifier	DeclinedMessage value is present only when ResponseCode is 1. This contains a textual description of the declined code.

Example DeletePaymentSchedule XML Response:

```
<Response>
        <ResponseCode>0</ResponseCode>
        <DeclinedCode></DeclinedCode>
        <DeclinedMessage></DeclinedMessage>
</Response>
```

5 Appendix

5.1 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).

5.2 API Overview

5.2.1 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.

5.2.2 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 < scheduleXML > 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">
<scheduleXML >string</ scheduleXML >
  </ SOAP METHOD >
 </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>

]]>
```

5.2.3 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.

5.3 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

5.4 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