IntegraPay

Business payments made simple.

INTEGRAPAY HOSTED PAYMENT SERVICES v2.1

Table of Contents

IntegraPay

Business payments made simple.

Document Control

Version	Date	Author	Change Summary
v1.0	15/01/2013	JL	First release
v1.1	15/01/2013	TB	Introduction, General Guidelines and Flow
v1.2	16/01/2013	TB	Improved flow diagram
v1.3	25/03/2013	JL	Added transparent redirect options
v1.4	15/4/2013	JL	Added more details to transparent redirect
v1.5	14/12/2015	JL	Added ability to choose HPP template
V1.6	19/05/2017	JL	Added Payer info, transaction info and VerficationStatus response fields to PostHostedRealtimePayment API call.
V2.0	27/05/2017	TB	Added JavaScript interface, and JavaScriptSubmit field to the PreHostedPayment API call. transactionDebitAmountInCents changed to transactionAmountInCents More detail around mandatory fields.
V2.1	24/08/2017	AG	Improved readability

Introduction

Integration with IntegraPay’s Hosted Payment services allows clients to be able to seamlessly and efficiently integrate their Web Applications, such as Invoice Management systems and Shopping Carts, with the IntegraPay Payment Gateway to easily and securely accept credit card payments.

The API works in conjunction with either the Hosted Payment Page or the JavaScript plugin, to provide PCI DSS secure capture and processing of Real Time transactions. Correct implementation of the PCI DSS secure Payment Service greatly reduces the scope of PCI DSS on the calling websites or applications.

The Hosted Payment Page allows some formatting to make it appear similar to the web application that is calling it . The formatting must be set up in conjunction with your IntegraPay consultant.

The Transparent Redirect and JavaScript options allow the developer to paint their own pages, thus maintaining control over the User Interface.

General Guidelines

The Hosted Payment Services will process transactions at any time throughout the day and the money from these transactions will be taken from the customers’ account immediately.

The settlement of the money to the business, minus fees, will vary based on the time of the transaction within the day (banks typically cut off at 6pm), the Merchant being used and the bank where your account resides.

Reports of Hosted Payment Page transactions will reflect the day that the transaction was performed, rather than the date of settlement.

The Hosted Payment Services will from time to time be updated with new functionality. IntegraPay will endeavour to notify businesses in advance of these changes, in particularly if they may affect the operation of the integration.

Choosing the best interface

IntegraPay offers three interfaces for integrating Hosted Payment Services to your application or website.

Each of these methods allow you to process payments from your customers without their credit card information passing through your web server, thus minimizing your scope for PCI compliance. However, the methods differ in the degree of control you have over the process and how it appears, and the way in which the customer interacts with it.

Option 1. JavaScript Payments

This option is the most seamless to your customer as it allows you to display the payment page entirely within your website where you have full control over its appearance and styling, and unlike the other two methods, it doesn't require any redirects of your customer to IntegraPay's hosted payments website. Instead it works by using a client-side jQuery Ajax API call to process the payment. Once the payment is processed the sensitive card information is stripped out of the form before it is submitted back to your server.

Pros

- You have full control over the UI/UX. The Payers browser never leave the web application
- The Credit Card Numbers are never available to your Web Application, therefore reducing PCI scope.
- Validation and presentation of errors are automatically displayed

Cons

- Consumers may be hesitant to enter Credit Cards on your site – you will need to make them confident it is secure and trustworthy.

Option 2. Hosted Payment Page redirect

Using this option, your customer is redirected to a form on IntegraPay's website into which they enter their credit card details and process the transaction. Upon successful completion, the customer is redirected back to your website.

Pros

- Payers can see that the credit card details are being entered on IntegraPay’s secure page
- The Credit Card Numbers are never available on your Web Application pages – therefore PCI scope greatly reduced
- The page is easily branded with Logos and Merchant names
- The page already has inbuilt validation and is built to include other IntegraPay services such as extended Fraud Monitoring
- The page can include alternative payments methods
- Numerous HPP templates can be created by IntegraPay to display and capture different sets of data depending on which HPP they are directed to.

Cons

- UI/UX/Customer Experience. There is little control over ‘look and feel’ – so it will look different to your web application

Option 3. Transparent Redirect

Like option 1, this allows you to create and brand your own payments page. Upon confirming payment, the customer is transparently redirected to IntegraPay's web server where the Credit Card is processed. The customer is almost immediately redirected back to your website, so that it appears they never left.

Generally, IntegraPay would recommend the JavaScript plugin library (option 1) over transparent redirect as it provides better control over error handling and page content.

Pros

- You have full control over the UI/UX. The Payers never appear to leave the web application
- The Credit Card Numbers are never available to your Web Application, therefore reducing PCI scope.

Cons

- The page must be a single page for the capture Credit Card details for the transaction. As the page is transparently redirected to the IntegraPay page, other data captured in fields will be lost as IntegraPay will not know to capture them.
- Validation – you would have to include your own JavaScript validation on the page (remember you can't post the details back to the web server as you will expose to greater PCI scope.
- Results – If the transaction fails for any reason, the customer would be redirected back to your page. The reason for the failure can be displayed by calling the POST API call, but any details entered on that page such as Credit Card Numbers would need to be re-entered.
- Extra IntegraPay services such as extended Fraud Monitoring or Alternative Payment types cannot be integrated
- Consumers may be hesitant to enter Credit Cards on your site – you will need to make them confident it is secure and trustworthy

Integration Credentials

To use this service your software will need to securely store these settings:

1) The URL to the web service.

2) A username and password to authenticate with the service that is unique to your business.

A test username and password is required for all end user testing. Please contact customerintegration@integrapay.com.au to obtain your username and password.

API Access Method

The API is an "XML over HTTPS" service. You submit an XML string to the service URL and you will receive an XML string back.

Connectivity Requirements:

1) The XML string must be submitted to the service URL via HTTP POST.
2) The "Content-Type" HTTP header of your request should be set to " text/xml".

API URL:

LIVE address:

https://payments.integrapay.com.au/API/API.ashx

TEST address:

https://testpayments.integrapay.com.au/API/API.ashx

Hosted Payment Page URL:

LIVE address:

https://payments.integrapay.com.au/RTP/Payment.aspx?webPageToken=YOUR_TOKEN_HERE

TEST address:

https://testpayments.integrapay.com.au/RTP/Payment.aspx?webPageToken=YOUR_TOKEN_HERE

How the Hosted Payment Process Flows

Option 1 – JavaScript payments

To process a payment with the JavaScript payments you need to perform the following steps:

1. Call our "PreHostedRealTimePayment" API function, with the “javascriptSubmit” parameter set to “1”, from a server-side call on your web site to register all the details of the payment you wish to be processed. This function will return a token.

2. Display a HTML form on your website to your customer with the required html elements wired up to be processed by IntegraPay's JavaScript library, with the token your received embedded in a hidden field on the form.

3. The customer will enter their card details into the form and click the submit button. IntegraPay's JavaScript library will then intercept the card information, perform some validation on it, and if it's ok, submit it directly to IntegraPay's payment API to process the payment.

4. Once the payment is processed, the sensitive card information will be removed from the form, and the form will then be submitted to your server calling whatever action you have specified on your form tag.

5. After the form is submitted you will need to call our "PostHostedRealTimePayment" API function from a server-side call to retrieve the results of the transaction.

Please see the "Using JavaScript payments" section below for details on how to implement this option.

Option 2 - Hosted Page Redirect

Using this option, your customer is redirected to a form on IntegraPay's into which they enter their credit card details, after which they are redirected back to your website. The process goes as follows:

1. Call our "PreHostedRealTimePayment" API function from a server-side call on your web server to register all the details of the payment you wish to be processed. This function will return a token.
2. Redirect to the IntegraPay Hosted Payment Page, passing the token in the URL
3. The user completes the transaction on the IntegraPay page and is redirected back to your website
4. After the form is submitted you will need to call our "PostHostedRealTimePayment" API function from a server-side call to retrieve the results of the transaction.

Option 3 - Transparent Redirect

Using this option your customer never sees IntegraPay's website - your customer enters their credit card details into a form on your own website that is submitted directly to IntegraPay's website. The payment is then processed and the user is immediately redirected back to your website again. The process is as follows:

1. Call our "PreHostedRealTimePayment" API function from a server-side call on your web server to register all the details of the payment you wish to be processed. This function will return a token.
2. You display a page on your web application which has the web token in a hidden field. The customer then inputs their credit card details and submits the payment. The button that submits directly to the IntegraPay web server so the Credit Card details are not passed through your web server.
3. When your redirection hits the IntegraPay web server, the transaction is processed and the customer is immediately redirected back to the page you nominated in the PreHostedRealTimePayment function.
4. After the customer is redirected back to your application, you will need to call our "PostHostedRealTimePayment" API function from a server-side call to retrieve the results of the transaction.

Please see the "Using Transparent Redirect" section below for details on how to implement this option.

Service Functions

Function 1 - PreHostedRealTimePayment

This function is used to register all the details of the payment that you wish your user to make. When your user is redirected to our hosted payment webpage it will be pre-filled in with the details you provided using this function.

If all the information you provided is correct, the function will return a token that you can then add to the querystring of the URL when you redirect your user to our hosted payment page. The same token is used again with the "PostHostedRealTimePayment" function to look up the results of the transaction when the user is redirected back to your website. Therefore, before redirecting the user, it is important that you store the token within your database and associate it with whatever transaction/order/registration etc. is being performed within your system so you can update it with the results of the transaction.

PLEASE NOTE: The webPageToken will expire after 20 minutes - if no payment is processed within this time the token will become unusable.

XML Request Specification:

- **request** (string) [Required] – The root XML element which contains elements for each of the parameters below.
- **username** (string) [Required] – The API access username that identifies our business.
- **password** (string) [Required] – The API access password assigned to your business to prevent unauthorised access to the service.
- **command** (string) [Required] – Must be set to "PreHostedRealTimePayment".
- **returnUrl** (string) [Conditional – Optional when using JavaScript Payments, required for all other methods] – This is the URL of a page within your website that our hosted payment page will redirect your user to after they have completed the payment. It must be an absolute URL (i.e. contain http:// or https://). If you wish you can add your own querystring values to the URL. When the user is redirected back we will add the "webPageToken" value to the querystring.
- **transactionID** (string – max 100 chars) [Required] – A reference that will uniquely identify this individual transaction within your system. If a transaction already exists in our database with the same ID (this includes scheduled debits and real-time

payments as well as transactions that you have already) then the "DUPLICATE_TRANSACTION" error will be return (see XML response section below).

- **transactionDebitAmountInCents** (long - non-negative) [Required] – The amount that is to be paid by credit card (e.g. for $30.75 the value 3075 would be passed in). When using the test API, the transaction amount gives you the ability to test both successful and failed payments. When the transaction amount has zero cents or a number of cents other than the ones listed below, the test transaction will always be successful. If you provide one of the following numbers for the number of cents you will receive a transaction failed response with the corresponding reason:
  - 31 = Invalid card
  - 54 = Expired card
  - 51 = Declined
  - 61 = Insufficient Funds
  - 96 = Technical failure
- **transactionDescription** (string – max 200 chars) [Optional] – If you wish you can include additional information to be recorded with the transaction that describes the purpose of the transaction (this will also be displayed to the user on the hosted payment page).
- **payerUniqueID** (string – max 100 chars) [Optional] – If you have previously added the person making the payment to IntegraPay's database via IntegraPay's website or other API functions, you can pass in the unique reference you used to create the person. Doing so will causes this real-time transaction to be attached to the transaction history of this person in IntegraPay's database. This parameter is optional – if no value is provided the transaction will be associated only with the business that is receiving the payment. If a value is provided and no payer can be found with the given ID this function will return a "PAYER_NOT_FOUND" error (see XML response section below).
- **payerFirstName** (string – max 50 chars) [Optional (Recommended)/ *Mandatory] – While this parameter is optional, if you have previously captured the payer's name within your system it is recommended that you pass it into this function. This is

because if you don't provide the payer's name the user will be required to enter their name on the hosted payment page (the same applies to "payerLastName" below).

- **payerLastName** (string – max 50 chars) [Optional (Recommended) / *Mandatory]
- **payerAddressStreet** (string – max 255 chars) [Optional (Recommended) / *Mandatory] - This is the payer's billing address, and as with the payerFirstName and payerLastName parameters, it is recommend that you pass through values for each of the address parameters if you have already captured them for your user. If you don't your user will be required to enter their billing address on the hosted payment page.
- **payerAddressSuburb** (string – max 50 chars) [Optional (Recommended) / *Mandatory]
- **payerAddressState** (string – max 50 chars) [Optional (Recommended) / *Mandatory]
- **payerAddressPostCode** (string – max 50 chars) [Optional (Recommended) / *Mandatory]
- **payerAddressCountry** (string – max 50 chars) [Optional (Recommended) / *Mandatory]
- **payerEmail** (string – max 255 chars) [Optional/ *Mandatory]
- **payerPhone** (string – max 50 chars) [Optional/ *Mandatory] – Non-mobile contact number.
- **payerMobile** (string – max 50 chars) [Optional/ *Mandatory] – Mobile phone contact number.
- **transparentRedirect** (integer - either 1 or 0) [Optional - default: 0] – Set to 1 if you wish to use the transparent redirect method (see "Option 2 - Transparent Redirect" section above for details).
- **javascriptSubmit** (integer - either 1 or 0) [Optional - default: 0] – Set to 1 if you wish to use the JavaScript Payments method (see "Option 1 – JavaScript Payments " section above for details).
- **hppTemplateGUID** (string) [Optional] – If your business has set up custom templates for the display and functionality of the hosted payment page, you can pass

the ID of a template into this field to set which template will be displayed to the user. Your IntegraPay representative can advise you on what your template IDs are.

- **auditUsername** (string) [Optional] – You can set this to be the user or system ID etc. of your software that initiated the call to the API so it is logged along with the transaction audit history that we keep in our database.

*Mandatory – Depending on intended use, some fields may be set up as mandatory by IntegraPay at it's discretion.

This is an example of the XML string with only the required parameters:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <username>1234</username>
  <password>X7{F?Hk92/Qj}n7Lvv6</password>
  <command>PreHostedRealTimePayment</command>
  <returnUrl>HTTP://www.test.com/TransactionResult.aspx</returnUrl>
  <transactionID>INV1234567</transactionID>
  <transactionAmountInCents>11095</transactionAmountInCents>
</request>
```

This is an example of the XML string which includes the optional parameters:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <username>1234</username>
  <password>X7{F?Hk92/Qj}n7Lvv6</password>
  <command>PreHostedRealTimePayment</command>
  <returnUrl>HTTP://www.test.com/TransactionResult.aspx</returnUrl>
  <transactionID>INV1234567</transactionID>
  <transactionAmountInCents>11095</transactionAmountInCents>
  <transactionDescription>12 months membership starting
15/01/2013</transactionDescription>
  <payerUniqueID></payerUniqueID>
  <payerFirstName>John</payerFirstName>
  <payerLastName>Smith</payerLastName>
  <payerAddressStreet>1 Test St.</payerAddressStreet>
  <payerAddressSuburb>Testville</payerAddressSuburb>
  <payerAddressState>QLD</payerAddressState>
  <payerAddressPostCode>4000</payerAddressPostCode>
  <payerAddressCountry>AUSTRALIA</payerAddressCountry>
  <payerEmail>test@test.com</payerEmail>
  <payerPhone>0733331234</payerPhone>
  <payerMobile>0411123123</payerMobile>
  <auditUsername>WEBSYS-101</auditUsername>
</request>
```

XML Response Specification:

The function will return an XML string. When parsing this XML, it is important to check the "result" element first. This element will tell you if the API call succeeded or whether an error occurred (if successful the result element will have a value of "OK", if it failed it will have a value of "ERROR"). The other elements in the XML response will differ depending on whether the function call was successful or not as detailed below:

SUCCESS RESPONSE

- **response** (string) [Required] – The root XML element which contains elements for each of the parameters below.
- **result** (string) - will always be "OK" for a success response.
- **webPageToken** (string - 64 characters) – this is the token string that you use to redirect your user to our hosted payment page (see the introduction to this function above for more details on using the token).

Example success response XML:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <result>OK</result>
<webPageToken>d1eee786222e49539a1b31ee17593b60f7e3bf953da345a0b06dad01c3292dab</webPageToken>
</response>
```

ERROR RESPONSE

- **response** (string) [Required] – The root XML element which contains elements for each of the parameters below.
- **result** (string) - will always be "ERROR" for a error responses.
- **errorType** (string) - specifies the type of error. Will be one of the following strings:
  - "XML_FORMAT_ERROR"
  - "PARAMETER_INVALID"
  - "DUPLICATE_TRANSACTION"
  - "PAYER_NOT_FOUND"
  - "ACCESS_DENIED"
  - "SERVICE_UNAVAILABLE"
- **errorMessage** (string) - this will contain a detailed explanation of the cause of the error.

Example error response XML:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <result>ERROR</result>
  <errortype>PARAMETER_INVALID</errortype>
  <errormessage>The 'transactionID' parameter is missing</errormessage>
</response>
```

Function 2 - PostHostedRealTimePayment

Once our hosted payment webpage redirects the user back to your website, this function can be used to lookup the result of the transaction.

XML Request Specification:

- **request** (string) [Required] – The root XML element which contains elements for each of the parameters below.
- **username** (string) [Required] – The API access username that identifies your business.
- **password** (string) [Required] – The API access password assigned to your business to prevent unauthorised access to the service.
- **command** (string) [Required] – Must be set to "PostHostedRealTimePayment".
- **webPageToken** (string - 64 chars) [Required] – The token that you orignally received from the "PreHostedRealTimePayment" function and used to redirect your user to our hosted payment page.

Example request XML:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <username>1234</username>
  <password>X7{F?Hk92/Qj}n7Lvv6</password>
  <command>PostHostedRealTimePayment</command>
<webPageToken>d1eee786222e49539a1b31ee17593b60f7e3bf953da345a0b06dad01c3292
dab</webPageToken>
</request>
```

XML Response Specification:

The function will return an XML string. When parsing this XML, it is important to check the "result" element first. This element will tell you if the API call succeeded or whether an error occurred (if successful the result element will have a value of "OK", if it failed it will have a value of "ERROR"). The other elements in the XML response will differ depending on whether the function call was successful or not as detailed below:

SUCCESS RESPONSE

- **response** (string) [Required] – The root XML element which contains elements for each of the parameters below.

- **result** (string) - will always be "OK" for a success response.

- **transactionID** (string) - the unique transaction ref that you passed into the PreHostedRealTimePayment function.

- **resultID** (string – 1 character) – A single character that specifies the result of the transaction:

  - "S" – Successful – the transaction was processed successfully and the funds have been transferred.

  - "R" – Retry later – the transaction was not able to be processed, possibly due to bank unavailability or technical malfunction. The transaction may be able to be processed at a later time (please note that if you wish to re-attempt to process the transaction again later you will need to provide a different TransactionID).

  - "F" – Failed – The transaction was processed by the bank but the funds were not able to be obtained and the payment was dishonoured. See the “resultRejectionTypeID” below for the reason for the transaction failing.

  - "C" – Cancelled – The user cancelled the transaction by clicking the Cancel button on our hosted payment webpage.

  - "N" – Not Processed – the transaction was not processed. This can occur if the user has not completed the payment on our hosted payment webpage.

  - "E" – Form Error – This response can occur only if you are using the Transparent Redirect method. It means that the credit card data your user entered into your website was missing or invalid. When this occurs check the "formErrorType" and "formErrorMessage" fields to find out what the problem was (see below).

- **resultRejectionTypeID** (integer) – If the transaction was processed successfully this parameter will be set to zero, if the transaction failed it will have a number greater than zero. Possible values for this parameter are:

  - 0 – Successful transaction
  - 1 - Insufficient Funds
  - 3 - Invalid Credit Card
  - 4 - Expired Credit Card
  - 5 - Technical Failure
  - 6 - Transaction Declined
  - 7 - Authority Revoked By Payer
  - 8 - Payer Deceased
  - 11 - Invalid Payer Contact Details

- **resultDescription** (string) – Contains a text description of the reason for the transaction failing (this will be empty if the transaction was successful).
- **resultBankReceiptID** (string) – Contains a receipt transaction ID that was assigned to the transaction by the bank.
- **resultBankSettlementDate** (string – format: YYYY-MM-DD) – Depending on the bank processing the transaction, this may contain the date on which the bank will deposit the transaction amount into the business' bank account. If the transaction failed, or no settlement date was given by the bank, this value will be empty.
- **creditCardNumberMasked** (string) – Contains the credit card number used by the user to make the payment. It will be masked so no more than the first 5 and last 4 digits will be visible.
- **creditCardName** (string) – The name of the credit card that the user used to make the payment.
- **creditCardType** (string) – The type of credit card the user used to make the payment. Will have one of the following values:
  - MASTERCARD
  - VISA
  - AMEX

  - DINERS CLUB
  - DISCOVER CARD
  - JCB CARD
  - CHINA UNIONPAY

- **formErrorType** (string) – This will only have a value when using transparent redirect and the credit card data your user entered was missing or invalid. This fields gives a code for the type of error and will be set to one of the following values:
  - MISSING_CARD_NUMBER
  - MISSING_CARD_NAME
  - MISSING_EXPIRY_YEAR
  - MISSING_EXPIRY_MONTH
  - MISSING_CCV
  - INVALID_CCV
  - INVALID_EXPIRY
  - INVALID_CARD_NUMBER
  - EXPIRED_CARD
  - UNSUPPORTED_CARD
- **formErrorMessage** (string) – This is a more detailed message describing the problem that occurred with the data entered by your user when using the transparent redirect method (it will only have a value when the "formErrorType" parameter has a value).
- **payerFirstName** (string) – First name entered by the customer if the field is enabled on the form.
- **payerLastName** (string) – Last name entered by the customer if the field is enabled on the form.

- **payerAddressStreet** (string) – Address entered by the customer if the field is enabled on the form.
- **payerAddressSuburb** (string) – Address entered by the customer if the field is enabled on the form.
- **payerAddressState** (string) – Address entered by the customer if the field is enabled on the form.
- **payerAddressPostCode** (string) – Address entered by the customer if the field is enabled on the form.
- **payerAddressCountry** (string) – Address entered by the customer if the field is enabled on the form.
- **payerEmail** (string) – Email entered by the customer if the field is enabled on the form.
- **payerPhone** (string) – Phone entered by the customer if the field is enabled on the form.
- **payerMobile** (string) – Mobile number entered by the customer if the field is enabled on the form.
- **transactionAmount** (string) – Final transaction amount that was processed by the bank (this can differ from the request transaction amount if you are using payer paid fees).
- **transactionDateTime** (string) – Date/Time the transaction was processed.
- **verficationStatus** (string) – This field will be blank if your transaction was processed successfully and passed all security monitoring and fraud detection checks. If your transaction was flagged by one of these checks this field will be set to “UR” meaning Under Review. Transactions that are under review will not be settled to you until our risk department has checked the transaction. If the transaction looks ok it will be released and settled to you normally, if fraud is suspected the transaction may be refunded immediately before any funds are settled to you.

Example success response XML - successful payment:

```
<?xml version="1.0" encoding="utf-16"?>
<response>
```

```
  <result>OK</result>
  <transactionID>INV333549183</transactionID>
  <resultID>S</resultID>
  <resultDescription />
  <resultRejectionTypeID>0</resultRejectionTypeID>
  <resultBankReceiptID>RT46182</resultBankReceiptID>
  <resultBankSettlementDate />
  <creditCardNumberMasked>411111xxxxxx1111</creditCardNumberMasked>
  <creditCardName>J SMITH</creditCardName>
  <creditCardType>VISA</creditCardType>
  <formErrorType />
  <formErrorMessage />
  <payerFirstName />
  <payerLastName />
  <payerAddressStreet />
  <payerAddressSuburb />
  <payerAddressState />
  <payerAddressPostCode />
  <payerAddressCountry />
  <payerEmail />
  <payerPhone />
  <payerMobile />
  <transactionAmount>20.50</transactionAmount>
  <transactionDateTime>2017-05-19 13:45:17</transactionDateTime>
  <verificationStatus />
</response>
```

Example success response XML - rejected payment:

```
<?xml version="1.0" encoding="utf-16"?>
<response>
  <result>OK</result>
  <transactionID>TXN35672</transactionID>
  <resultID>F</resultID>
  <resultDescription>REJECTED - INSUFFICIENT FUNDS
</resultDescription>
  <resultRejectionTypeID>5</resultRejectionTypeID>
  <resultBankReceiptID>RT46184</resultBankReceiptID>
  <resultBankSettlementDate />
  <creditCardNumberMasked>411111xxxxxx1111</creditCardNumberMasked>
  <creditCardName>MR J SMITH</creditCardName>
  <creditCardType>VISA</creditCardType>
  <formErrorType />
  <formErrorMessage />
  <payerFirstName />
  <payerLastName />
  <payerAddressStreet />
  <payerAddressSuburb />
  <payerAddressState />
  <payerAddressPostCode />
  <payerAddressCountry />
  <payerEmail />
  <payerPhone />
  <payerMobile />
  <transactionAmount>30.96</transactionAmount>
  <transactionDateTime>2017-05-19 13:48:21</transactionDateTime>
  <verificationStatus />
</response>
```

ERROR RESPONSE

- **response** (string) [Required] – The root XML element which contains elements for each of the parameters below.

- **result** (string) - will always be "ERROR" for a error responses.

- **errorType** (string) - will always be "ERROR" for a error responses. This function can return the following error types:

  - "XML_FORMAT_ERROR"

  - "PARAMETER_INVALID"

  - "ACCESS_DENIED"

  - "SERVICE_UNAVAILABLE"

- **errorMessage** (string) - this will contain a more detailed explanation of the cause of the error.

Example error response XML:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <result>ERROR</result>
  <errortype>PARAMETER_INVALID</errortype>
  <errormessage>The webPageToken you supplied was not
found</errormessage>
</response>
```

Implementing JavaScript Payments

How It Works

The JavaScript library works in conjunction with the "PreHostedRealTimePayment" and "PostHostedRealTimePayment" API functions which are detailed above.

You call the PreHostedRealTimePayment function, with the “javascriptSubmit” parameter set to “1”, to obtain a token which you then place in a hidden input field within a form you create within your website. This form is wired up to connect to IntegraPay's payments API via a client-side API call using jQuery ajax.

How to Configure Your Website

Setting up your website to support JavaScript payments simply requires you to link to IntegraPay's JavaScript library and create a html form with each of the html element/tags described below. Each element requires you to include a "data-integrapay" attribute which wires up your form to our JavaScript library.

Each of the required HTML elements are listed below. As long as the "integrapay-data" attributes are set correctly, you are free to add any additional attributes to the tags, such as "class" and "style" tags etc., to suit your website.

1) Scripts

The standard jQuery library is required for IntegraPay's JavaScript library to work. If you don't already have jQuery referenced in your website you will need to add it in a script tag. e.g. to load it from Google's CDN you can use:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

You will then need to create a script reference to IntegraPay's JavaScript library.

For our Sandbox site the reference is:

```
<script src="https://testpayments.integrapay.com.au/JS/"></script>
```

For our Production site the reference is:

```
<script src="https://payments.integrapay.com.au/JS/"></script>
```

2) Form Tag

A standard POST form tag needs to be added which must contain all the other html elements listed below. The form "action" can be set to post to any location on your website.

```
<form action="INSERT YOUR ACTION URL HERE" method="POST" data-integrapay="PaymentForm">
```

3) Input Tags

A hidden input field needs to be added to store the token that you received when you called the "PreHostedRealTimePayment" API function. You will need to set the value of this hidden field to the token from your server when you render the page:

```
<input type="hidden" id="apiToken" name="apiToken" value="INSERT YOUR

TOKEN HERE" data-integrapay="ApiToken" />
```

Add the following input fields for capturing the credit card details:

```
<input type="text" data-integrapay="CardName" />

<input type="text" data-integrapay="CardNumber" />

<input type="text" data-integrapay="CardCcv" />
```

4) Select Tags

For capturing the card expiry date information the following two select dropdown lists need to be added:

```
<select data-integrapay="CardExpiryMonth">
        <option value="01">January</option>
        <option value="02">February </option>
        <option value="03">March</option>
        <option value="04">April</option>
        <option value="05">May</option>
        <option value="06">June</option>
        <option value="07">July</option>
        <option value="08">August</option>
        <option value="09">September</option>
        <option value="10">October</option>
        <option value="11">November</option>
        <option value="12">December</option>
</select>
<select data-integrapay="CardExpiryYear">
        <option value="2017">2017</option>
        <option value="2018">2018</option>
        <option value="2019">2019</option>
        <option value="2020">2020</option>
        <option value="2021">2021</option>
        <option value="2022">2022</option>
        <option value="2023">2023</option>

        <option value="2024">2024</option>

        <option value="2025">2025</option>

</select>
```

5) Div tags

These two div tags are used to display messages and status information to the customer.

If the validation of the card information fails, or an error occurs during the processing of the payment, a description message will be inserted into this div:

```
<div data-integrapay="Errors"></div>
```

While the payments is being processed, a loading spinner gif will be placed into this div to give the customer feedback to wait while the payment is being processed:

```
<div data-integrapay="Processing"></div>
```

6) Button tag

The following button tag will need to be included in the form which the customer will click to process the payment.

IMPORTANT: This button must not be a submit button as this could cause the sensitive data to be inadvertently sent to your server - i.e. it must have *type="button"* instead of *type="submit"* so that the IntegraPay library can control when the form will be posted.

```
<button type="button" data-integrapay="SubmitButton">Process Payment</button>
```

Example Form

Below is an example of a complete form using Bootstrap for UI styling:

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">


<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Credit Card JavaScript Demo</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa"
crossorigin="anonymous"></script>
    <script src="https://localhost/PaymentWeb/JS/"></script>
</head>


<body>
    <div class="container">


        <h4>Enter Payment Method</h4>


        <form action="[INSERT YOUR ACTION URL HERE]" method="POST" class="form-horizontal"
id="PaymentForm" data-integrapay="PaymentForm">
            <input type="hidden" value="[INSERT YOUR TOKEN HERE]" id="apiToken"
name="apiToken" data-integrapay="ApiToken" />


            <div class="form-group">
                <label for="cardName" class="col-sm-2 control-label">Name on Card</label>
                <div class="col-sm-3">
                    <input id="cardName" type="text" class="form-control" data-integrapay="CardName"
/>
                </div>
            </div>


            <div class="form-group">
                <label for="cardNumber" class="col-sm-2 control-label">Card Number</label>
                <div class="col-sm-3">
                    <input id="cardNumber" type="text" class="form-control" data-
integrapay="CardNumber" />
```

```
                </div>
            </div>

            <div class="form-group">
                <label for="cardCvv" class="col-sm-2 control-label">CCV</label>
                <div class="col-sm-1">
                    <input id="cardCvv" type="text" class="form-control" data-integrapay="CardCcv" />
                </div>
            </div>

            <div class="form-group">
                <label for="cardExpiryMonth" class="col-sm-2 control-label">Expiry Month</label>
                <div class="col-sm-2">
                    <select id="cardExpiryMonth" class="form-control" data-
integrapay="CardExpiryMonth">
                        <option value="01">January</option>
                        <option value="02">February </option>
                        <option value="03">March</option>
                        <option value="04">April</option>
                        <option value="05">May</option>
                        <option value="06">June</option>
                        <option value="07">July</option>
                        <option value="08">August</option>
                        <option value="09">September</option>
                        <option value="10">October</option>
                        <option value="11">November</option>
                        <option value="12">December</option>
                    </select>
                </div>
            </div>

            <div class="form-group">
                <label for="cardExpiryYear" class="col-sm-2 control-label">Expiry Year</label>
                <div class="col-sm-2">
                    <select id="cardExpiryYear" class="form-control" data-integrapay="CardExpiryYear">
                        <option value="2017">2017</option>
                        <option value="2018">2018</option>
                        <option value="2019">2019</option>
                        <option value="2020">2020</option>
                        <option value="2021">2021</option>
```

```html
                    </select>
                </div>
            </div>

            <div class="form-group">
                <div class="col-sm-offset-2 col-sm-5">
                    <div class="text-danger" data-integrapay="Errors"></div>
                    <div data-integrapay="Processing"></div>
                </div>
            </div>

            <div class="form-group">
                <div class="col-sm-offset-2 col-sm-5">
                    <button type="button" class="btn btn-default" data-integrapay="SubmitButton">Process Payment</button>
                </div>
            </div>

        </form>
    </div>

</body>

</html>
```

Implementing Transparent Redirect

To use transparent redirect, you must add a special html form to your website which posts its data directly to IntegraPay's website. This form contains fields for all of your customer's credit card details.

IMPORTANT: To use this method and maintain your PCI compliance, you must ensure that your form posts directly to IntegraPay's website without any of the form data passing through your web server. .NET users in particular must make sure that their form tag does not contain runat="server", and that it's not contained within any AJAX functionality or other server controls that may cause the page to automatically post form data to your server before submitting it to IntegraPay's site.

IMPORTANT: A weakness of the transparent redirect method is that, because this form must post directly to IntegraPay's website, it's not possible for you to perform any server-side validation on the data entered by your user to check that they have entered correct details. IntegraPay will validate all the data the user entered before attempting to process the payment, and if invalid or incomplete card details have been submitted the payment will not be able to be processed automatically. If this occurs IntegraPay will immediately redirect your user back to your website and you will need to redisplay your form so they can enter their card details again. You are able to find out if this occurred by checking if the value of the "resultID" field is set to "E" when you call the PostHostedRealTimePayment function (see above for details).

You may wish to consider using some client-side javascript on your website to validate the user's input to minimize the risk of this occurring.

Form Tag Specification:

Your form tag must contain

- method="post"
- action=One of the following addresses:

LIVE address:

"https://payments.integrapay.com.au/RTP/Payment.aspx?webPageToken=YOUR_TOKEN_HERE"

TEST address:

"https://testpayments.integrapay.com.au/RTP/Payment.aspx?webPageToken=YOUR_TOKEN_HERE
"

Form Fields Specification:

- **TransparentRedirect** (input - hidden - must be set to "1") [Required] – Hidden form field that must be set to "1" to signify that you are using transparent redirect.

- **CreditCardNumber** (input - text - integer 14 to 16 digits) [Required] – The person's credit card number.

- **CreditCardName** (input - text - max 50 characters) [Required] – The name on the person's credit card.

- **CreditCardExpiryMonth** (input - text - integer 2 digits) [Required] – The month of the expiry date on the person's card.

- **CreditCardExpiryYear** (input - text - integer 4 digits) [Required] – The year of the expiry date on the person's card.

- **CreditCardSecurityCode** (input - text - integer 3 or 4 digits) [Required] – The 3 or 4 digits CCV/CVC number on the person's credit card.

Example Form Html:

```
<form action=" https://testpayments.integrapay.com.au/RTP/Payment.aspx?webPageToken=YOUR_TOKEN_HERE" method="post">

        <input name="TransparentRedirect" type="hidden" value="1" />

        <label>Card number:</label>

        <input name="CreditCardNumber" type="text" maxlength="16" size="18" />

        <label>Name on card:</label>

        <input name="CreditCardName" type="text" maxlength="50" size="30" />

        <label>Expiry date:</label>

        <input name="CreditCardExpiryMonth" type="text" maxlength="2" size="2" />

        /
```

```
    <input name="CreditCardExpiryYear" type="text" maxlength="4" size="4" />
    (mm/yyyy)

    <label>Security code:</label>
    <input name="CreditCardSecurityCode" type="text" maxlength="4" size="4" />
    <input name="formSubmit" type="submit" value="Process Payment" />
</form>
```

CONTACT US

For more information, please contact:

TONY BRADEN - CIO
INTEGRAPAY
E: CUSTOMERINTEGRATION@INTEGRAPAY.COM
M: 0419313296
P: 07 3040 4320