# 

# Paycentre Web v.4.0

# User Guide

**Table of Contents**

Contents

[Paycentre Web v.4.0 1](#_Toc455698172)

[User Guide 2](#_Toc455698173)

[1. Introduction 4](#_Toc455698174)

[1.1 Resources 4](#_Toc455698175)

[1.2 Default Settings 4](#_Toc455698176)

[1.3 Implementation Support 5](#_Toc455698177)

[2. Payment Configuration Options 6](#_Toc455698178)

[2.1 : Embedded Iframe 6](#_Toc455698179)

[2.2.1 : Prepopulated Hosted Redirect Page (Prepopulated Hosted Payment Page) 7](#_Toc455698180)

[2.2.2 : Open Field Hosted Redirect Page ( Open Field Hosted Payment Page) 7](#_Toc455698181)

[2.3 : Form Post 7](#_Toc455698182)

[3. Add ons and Additional Features 8](#_Toc455698183)

[3.1 : Tokenization 9](#_Toc455698184)

[3.2 : 3D Secure 9](#_Toc455698185)

[3.3 : Hosted HTML & CSS 9](#_Toc455698186)

[4. Transaction Processing 10](#_Toc455698187)

[4.1 : Payment Work Flow 10](#_Toc455698188)

[4.2 : Payment Initialization message structure 12](#_Toc455698189)

[Form Post Data Fields 14](#_Toc455698190)

[The form post method is only for Merchants who are PCI compliant. 14](#_Toc455698191)

[Fields for form post request 14](#_Toc455698192)

[4.3: Payment Complete Message Structure 14](#_Toc455698193)

[5. Tokenization 17](#_Toc455698194)

[5.1 Storing a card and obtaining a token 17](#_Toc455698195)

[5.2 Performing a purchase and obtaining token simultaneously 20](#_Toc455698196)

[5.3 Tokenisation Reponses 22](#_Toc455698197)

[6. Making Payments Using Tokens 23](#_Toc455698198)

[7. Apendix 25](#_Toc455698199)

[7.1. FAQs 25](#_Toc455698200)

# 1. Introduction

Paycentre Web v.4.0 s allows merchants to process credit card payments in a PCI DSS compliant manner.

#### Document Purpose / Intended Audience

This guide is intended to act a reference point for web/application developers seeking to implement Paycorp transaction functionality into merchant websites and web applications using the Paycorp API for Paycorp's REST web services.

Our API will generate the jason messages for you and handles marshaling and unmarshaling of jason messages so that the merchant does not have to spend time writing code for these tasks.

Our API is currently available in Java, .NET and PHP. However if you are using any other language you may directly integrate into our REST web services and a separate technical reference user guide is available on request

It is assumed that the parties implementing this product will be experienced in the development and implementation of websites and web applications, this document is written principally toward parties with such expertise.

In Particular you should understand:

* JSON based Web Services
* Https protocols

1.1 Resources

When setting up an instance of Payment Pages, you will be provided the following information & resources:

* This Documentation
* Client Id
* Authtoken
* HMAC Secret

We will require from you the following information:

* The Location of any CSS you want to be applied to an iframe or Hosted Page

1.2 Default Settings

By Default, an advanced Payment Page instance is set up with the following Configuration.

|  |  |
| --- | --- |
| **Paramemter** | **Value** |
| Permitted Card Types | Visa, MasterCard, Amex |
| Page Type | iframe |
| Transaction type | Purchase |
| CVC required | True |
| Tokenize | True |
| 3D Secure | Disabled |
| Surcharging | Disabled |

Before processing transactions with our API you need to setup the credentials such as client id , Hmac, Authtoken and a Hmac secret as showen in the online technical reference guide bellow.

<https://s3-ap-southeast-2.amazonaws.com/gatewayapi/1.5/apidoc/index.html#api-1_Basics-GatewayClient>

PLEASE ENSURE YOU USE THE CORRECT CLIENT ID, AUTHTOKEN ,HMAC SECRET AND ENDPOINT AS PROVIDED TO YOU BY PAYCORP.

1.3 Implementation Support

Implementation support is available between 9am and 5pm AEST. IF you require support outside these times, please advise us and we will arrange for a resource to be made available. Email [setup@paycorp.com.au](mailto:setup@paycorp.com.au) or contact via phone on +61 2 9008 5701.

# 2. Payment Configuration Options

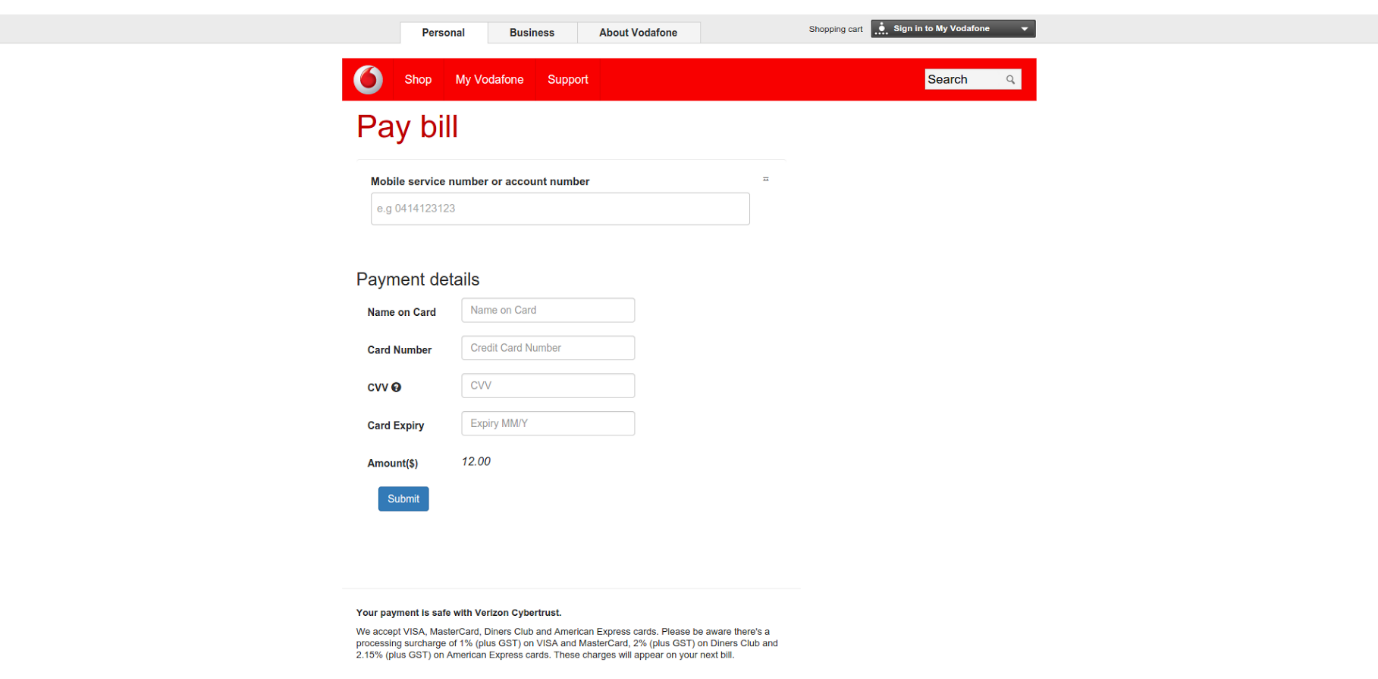
The Paycentre Web v.4.0 supports a wide variety of configurations to suit most Merchant Implementations. Review the below options to consider which implementation style suits your requirements.

2.1 : Embedded Iframe

**Why use an Iframe?**

Iframes are an good blend of security and compliance while causing minimal disruption to page layout. Iframe based payments are compliant according to PCI DSS v3, and can significantly reduce scope for scope for PCI compliance, unlike a Form Post or API. Iframe is the only method to keep your web service out of scope while maintaining a seamless appearance to your payment page.

Paycorp’s iframe option is highly customizable, you can bring your own CSS and even HTML to fully customize the page.



Example page with embedded iframe.

### 2.2.1 : Prepopulated Hosted Redirect Page (Prepopulated Hosted Payment Page)

**Why Use a Prepopulated Hosted Redirect?**

The Prepopulated Hosted Redirect brings the user off the merchant website and onto a discrete secure payment page hosted by Paycorp. the P repopulated Hosted Redirect has many of the same advantages of the iframe, being highly secure, fully PCI complaint and customizable using CSS and HTML. This can be a good option for small merchants wishing to leverage the trust and brand visibility of Paycorp.

### 2.2.2 : Open Field Hosted Redirect Page ( Open Field Hosted Payment Page)

**Why use an Open field Hosted Redirect?**

The Difference between the Open Field Hosted Redirect page and the Prepopulated one is that the metadata fields remain open to be entered by the cardholder. This can be a simpler implementation for small merchants. This is also useful where you need the cardholder to enter information about their payment themselves.

2.3 : Form Post

**Why use a Form post?**

This option is only for PCI complaint merchants only.The Form post give the Merchant the most control over the Customer Experience when making their payment. However, this comes at the significant cost of bringing the entire web server, and its associated devices and servers on the same network into PCI DSS scope. Form post is recommended only where the Merchant is already PCI DSS certified.

# 3. Add ons and Additional Features

In addition the fundamental payment process, the Advanced Payment page also allows for a number of separate features which

3.1 : Tokenization

In addition to processing Payments directly, Paycentre Web v.4.0 includes a tokenization service which enables you to store credit cards in Paycorp’s secure encripted vault . Here the card is stored in Paycorp's encrypted PCI compliant servers and we issue the merchant a code called a Token which represents stored card.

Paycentre Web v.4.0 can be used for storing cards with Paycorp's secure CardVault and obtaining tokens. However in order to process tokens you must use Real Time payments .

The Paycorp tokens are numeric non-mod10 values which retain the same first 6 and last 4 digits of the credit card they replace. There is a 1 to 1 relationship between the token and the card.

This enables the merchant to use the token is the same fields and locations as a credit card, displaying a masked token will look the same as a card and they will fit in the same database fields as a real card.

We will discuss Tokenization and making payments using Tokens in detail in section 8.

3.2 : 3D Secure

**Use with:** Any payment method

3D Secure is an additional security layer which allows the Issuer or the card to check that the transaction about to take place isn’t fraudulent before the actual payment takes place. The other major advantage of 3D secure is that the chargeback liability can be shifted away from the Merchant and back to the issuer. This is a great tool for high-risk payments or where security is paramount. The disadvantage of 3D secure is that the process can disrupt the checkout process.

Speak to your Paycorp Implementation Agent to get more information about how 3D secure can help your business.

3.3 : Hosted HTML & CSS

**Use with:** Iframes or Hosted Pages

One of the key advantages of Paycorp PCI Compliant payment solutions is that they remain PCI compliant whilst being highly customizable and integrate seamlessly into your own website.

**CSS:** The Paycorp page can reference an externally available CSS file hosted in the merchant environment. This way there is no need to contact Paycorp for any minor CSS updates.

**HTML:** Paycorp can also host a HTML template file to further customise the iframe or Hosted page, ensuring the page fits within your style guides and branding. For security purposes, this needs to be hosted by Paycorp, though can be easily updated.

# 4. Transaction Processing

Paycorp advanced payment pages offers three technical methods of processing transaction for merchants.

**Iframe Page**

Merchants will embed a paycorp hosted page as an iframe to their website. Merchants can host css for iframe, so they have full control over look and feel of page. A Merchant may also supply Paycorp with a HTML template to improve page seamlessness.

**Hosted Page**

Merchants will redirect payer to paycorp hosted page. Paycorp url will be visible in the page. Like iframe page, this method also enable merchants to have full control over look and feel of page.

**Form post**

Merchants who are already PCI DSS compliant can manage their own html and post payer's card information to Paycorp.

**Note: Paycorp Recommends the Iframe or Hosted solution for most Merchants**

4.1 : Payment Work Flow

The page is a highly customizable PCI compliant Payment that uses 2 JSON based web service calls and one https call in order to complete the Payment Request.

The **First** call is the ***Payment Initialization.***

Please see: <https://s3-ap-southeast-2.amazonaws.com/gatewayapi/1.5/apidoc/index.html#api-2_Payment-PaymentInit>

The response to this message is the **Paymentinit Response**.This response contains the **paymentPageUrl**. This is the URL you would embed in the iframe or re direct to in the case of re direct method (Hosted payment pages). For the iframe option you will embed the URL as follows <iframe src="value of the paymentPageUrl goes here "></iframe>

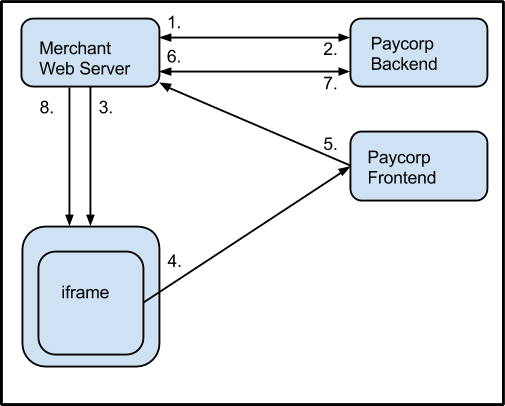
After the card data is entered by the customer in the capture page and submitted, the merchant will receive a response to the Receipt URL specified in the Payment Initialization request 'Redirect' parameter. This response will be via http GET with a parameter called ReqID .

The **Second** Call is the *the* ***Payment Complete call.***

This call is the trigger to submit the full bundle comprised of the first two calls to the Merchant’s acquiring Bank . The Merchant submits the ReqID and is returned with a full transaction response, including any/all metadata sent in the first Call, the Token, if it was requested, sanitized version of the cardholder data (truncated card, expiry date, no CVC). At this point the payment is complete.

Please see:

https://s3-ap-southeast-2.amazonaws.com/gatewayapi/1.5/apidoc/index.html#api-2\_Payment-PaymentComplete



1. **Merchant submit Payment\_Init call**
2. **Paycorp responds back with ReqID & Payment URL**
3. **Merchant’s web server delivers parent page to Browser, including the Payment URL embedded as an iframe**
4. **Cardholder completes payment, cardholder data captured by Paycorp.**
5. **Response returned to Receipt URL specified in Call 1.**
6. **Merchant submits Payment\_complete call**
7. **Paycorp confirms payment processed, returned full transaction bundle, including token if any**
8. **Merchant Web server serves browser with receipt page**

4.2 : Payment Initialization message structure

This method initializes a payment and returns a unique request id.

Operation: PAYMENT\_INIT

**Payment Init Request Fields**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| clientId | Integer | ClientId provided by paycorp  **Required**. |
| type | String | Options: PURCHASE, AUTHORISATION, TOKEN  Default: PURCHASE |
| tokenize | Boolean | Flag to tokenize card with PURCHASE and AUHTORISATION. If you only want to tokenize use type as TOKEN. |
| tokenReference | String | Reference used in conjunction with tokenization request.. |
| amount | Object | Transaction amount and currency.  paymentAmount: Type decimal  currency: 3 char ISO code  Required. |
| redirect | Object | Set of redirect URLs you provide for paycorp to rediect payer to merchant website.  **returnUrl:**The payer is redirected to this URL after submitting the payment.  ReturnMethod: |
| clientRef | String (50 Chars Max) | Merchant specified field |
| comment | String (100 Chars Max) | Merchant specified field |
| extraData | Map(100 Chars Max) | Merchant specified key value pair metadata. |

**Sample PAYMENT\_INIT Request** **generated by the API for you**

|  |
| --- |
| POST /paycorp-webservice/InterfaceServlet HTTP/1.1  Host: test-merchants.paycorp.com.au  AUTHTOKEN: d5c1b696-b86a-4613-92f8-62dd412b1a56  Content-Type: application/json  Cache-Control: no-cache  {  "operation":"PAYMENT\_INIT",  "requestDate":"2015-02-17 14:35:55",  "validateOnly":false,  "requestData":{  "clientId":11000010,  "type":"PURCHASE",  "tokenize":true,  "tokenReference":”egiwuegfiuwebf”,  "amount":{  "paymentAmount":12.0,  "currency":"NZD"  },  "redirect":{  "returnUrl":"http://some\_merchant.com/confirm",  "returnMethod":"GET"  },  "clientRef":"testRef",  "comment":"testComment",  "extraData":{  "msisdn":"0432384947",  "sessionId":"x2d2323r23r23"  }  }  } |

**PAYMENT\_INIT Response fields**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| reqid | Integer | A unique payment id provided by paycorp. |
| expireAt | String | Time at which this request will expire. Currently valid for 30 mins. |
| paymentPageUrl | Object | Url to get Iframe and hosted payment pages, not used for form post solution. |

**Sample PAYMENT\_INIT Response raw jason message.**

|  |
| --- |
| {  "operation":"PAYMENT\_INIT" ,  "msgProcessId":"f91f4a8f-e889-44e8-871e-a81ab3ff1399",  "responseData":{  "reqid":"5bDqiptxShatUDb5kzIY",  "expireAt":"2015-02-18T14:39:37.937+1100",  "paymentPageUrl":"[http://test-merchants.paycorp.com.au/webinterface/app/payment](http://localhost:9017/webinterface/app/payment)? reqid=5bDqiptxShatUDb5kzIY"  }  } |

## Form Post Data Fields

## The form post method is only for Merchants who are PCI compliant.

## Fields for form post request

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| cardType | String | The card type entered by the customer .  Acepted cardType:  VISA, MASTERCARD, AMEX, JCB, DINERS |
| cardHolderName | String | The card holder name entered by the customer. |
| cardNo | String | The card number entered by the customer. |
| cardExpiry | String | The card expiry in format MM/YY. |
| cardSecureId | String | The CVC2/CVV2 number entered by the customer. |

After submitting, paycorp will store payment details and redirect payer to return url using return method as provided in payment init request. Only parameter passed in the return url will be reqid.

Example: http://some\_merchant.com/confirm?reqid= 5bDqiptxShatUDb5kzIY

4.3: Payment Complete Message Structure

Completes payment and returns final transaction response.

Operation: PAYMENT\_COMPLETE

**Payment Complete Request Fields**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| clientId | Integer | ClientId provided by paycorp  **Required**. |
| reqid | String | Request Id provided by paycorp in payment\_init request. |

**Sample PAYMENT\_COMPLETE Request** **generated by the API for you**

|  |
| --- |
| POST /paycorp-webservice/InterfaceServlet HTTP/1.1  Host: test-merchants.paycorp.com.au  AUTHTOKEN: d5c1b696-b86a-4613-92f8-62dd412b1a56  Content-Type: application/json  Cache-Control: no-cache  {  "operation":"PAYMENT\_COMPLETE",  "validateOnly":false,  "requestData":{  "clientId":11000010,  "reqid": "5bDqiptxShatUDb5kzIY"  }  } |

**PAYMENT\_COMPLETE Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| clientId | Integer | ClientId provided by paycorp. |
| type | String | Transaction type as provided in payment init request |
| tokenize | Boolean | Flag to tokenize card with PURCHASE and AUTHORISATION transaction type |
| tokenReference | String |  |
| creditCard | object | Creditcard info submitted by payer  type  holderName  number  expiry |
| amount | object | Transaction amount in **CENTS** and currency type as submitted by merchant in payment init request.  E.g:  initRequest.transactionAmount = new TransactionAmount(1010,”AUD”); Please see currency codes given below. |
| clientRef | String | Merchant specified field |
| comment | String | Merchant specified field |
| extraData | String | Merchant specified key value pair metadata. |
| txnReference | Integer | Response Field - Paycorp transaction reference |
| responseCode | String | Response Field - Bank Response Code |
| responseText | String | Response Field - Bank Response Test |

**Sample PAYMENT\_COMPLETE Response**

|  |
| --- |
| {  "msgProcessId":"7d385fa3-a91b-4223-a6ca-b8b97dd1f973",  "operation":"PAYMENT\_COMPLETE"  "responseData":{  "clientId":11000010,  "transactionType":"PURCHASE",  "tokenize":true,  "tokenReference":"egiwuegfiuwebf",  "creditCard":{  "type":"VISA",  "holderName":"Rohit",  "number":"456445\*\*\*\*\*\*4564",  "expiry":"1218"  },  "token":{  "token":"4564456856784564",  "responseText":"Token already exists, card expiry updated"  },  "amount":{  "paymentAmount":12.0,  },  "clientRef":"testRef",  "comment":"testComment",  "extraData":{  "sessionId":"x2d2323r23r23",  "msisdn":"0432384947"  },  "txnReference":2015300000000327,  "responseCode":"00",  "responseText":"Transaction Successful",  }  } |

# 5. Tokenization

5.1 Storing a card and obtaining a token

Tokenization Request

//// JAVA ////

PaymentInitRequest initRequest = new PaymentInitRequest();

initRequest.setClientId(10000000);

initRequest.setTransactionType(TransactionType.TOKEN);

initRequest.setTransactionAmount(new TransactionAmount(0));

initRequest.setClientRef("merchant\_\_token\_reference");

initRequest.setRedirect(new Redirect("https://merchant.com"));

PaymentInitResponse paymentInitResponse = client.payment().init(initRequest);

//// .NET ////

PaymentInitRequest initRequest = new PaymentInitRequest();

initRequest.clientId = 10000000;

initRequest.transactionType = Enums.TransactionType.TOKEN.ToString();

// $10.10 dollars

TransactionAmount transactionAmount = new TransactionAmount(0);

initRequest.transactionAmount = transactionAmount;

initRequest.clientRef = "merchant\_\_token\_reference";

Redirect redirect = new Redirect("https://merchant.com");

initRequest.redirect = redirect;

PaymentInitResponse paymentInitResponse = client.payment.init(initRequest);

//// PHP ////

initRequest = new PaymentInitRequest();

initRequest->setClientId(10000000);

initRequest->setTransactionType(TransactionType::$TOKEN);

// $10.10 dollars

$transactionAmount = new TransactionAmount(0);

initRequest->setTransactionAmount($transactionAmount);

initRequest->setClientRef("merchant\_\_token\_reference");

$redirect = new Redirect("https://merchant.com");

initRequest->setRedirect($redirect);

$initResponse = $client->getPayment()->init(initRequest);

5.2 Performing a purchase and obtaining token simultaneously

[PURCHASE with tokenize Request:](https://s3-ap-southeast-2.amazonaws.com/gatewayapi/1.5/apidoc/index.html#parameter-examples-2_Payment-PaymentInit-1_5_0-2)

//// JAVA ////

PaymentInitRequest initRequest = new PaymentInitRequest();

initRequest.setClientId(10000000);

initRequest.setTransactionType(TransactionType.PURCHASE);

// $10.10 dollars as the amount must be entered in cents

//r”AUD” is the currency type . If this is not set then the default currency based

// on the country of the acquiring bank will apply

initRequest.setTransactionAmount(new TransactionAmount(1010,”AUD”));

initRequest.setClientRef("merchant\_payment\_reference");

initRequest.setRedirect(new Redirect("https://merchant.com"));

initRequest.setTokenize(true);

initRequest.setTokenReference("merchant\_token\_reference");

PaymentInitResponse paymentInitResponse = client.payment().init(initRequest);

//// NET ////

PaymentInitRequest initRequest = new PaymentInitRequest();

initRequest.clientId = 10000000;

initRequest.transactionType = Enums.TransactionType.PURCHASE.ToString();

// $10.10 dollars as the amount must be entered in cents

//r”AUD” is the currency type . If this is not set then the default currency based

// on the country of the acquiring bank will apply

initRequest.transactionAmount = new TransactionAmount(1010,”AUD”);

initRequest.clientRef = "merchant\_reference";

initRequest.comment = "merchant\_additional\_data";

initRequest.redirect = new Redirect("https://merchant.com");

initRequest.tokenize = true;

initRequest.tokenReference = "merchant\_token\_reference";

PaymentInitResponse initResponse = client.payment.init(initRequest);

//// PHP ////

initRequest = new PaymentInitRequest();

initRequest->setClientId(10000000);

initRequest->setTransactionType(TransactionType::$PURCHASE);

// $10.10 dollars as the amount must be entered in cents

//r”AUD” is the currency type . If this is not set then the default currency based

// on the country of the acquiring bank will apply

$transactionAmount = new TransactionAmount(1010,”AUD”);

initRequest->setTransactionAmount($transactionAmount);

initRequest->setClientRef("merchant\_payment\_reference");

$redirect = new Redirect("https://merchant.com");

initRequest->setRedirect($redirect);

initRequest->setTokenize(TRUE);

initRequest->setTokenReference("merchant\_token\_reference");initRequest = $client->getPayment()->init($paymentInitRequest);

**Please note that PayemntInitResponse ,PaymentCompleteRequest and PaymentCompleteResponse message structure and procedure is the same as the previous des****cription in section 4 and referenced online techincal guide page.**

**Please see the online technical reference guide given below guide for further information on storing tokens.**

[**https://s3-ap-southeast-2.amazonaws.com/gatewayapi/1.5/apidoc/index.html#api-1\_Basics-GatewayClient**](https://s3-ap-southeast-2.amazonaws.com/gatewayapi/1.5/apidoc/index.html#api-1_Basics-GatewayClient)

5.3 Tokenisation Reponses

The following responses can be returned upon successful token generation.

|  |  |
| --- | --- |
| **Response Code** | **Response Text** |
| 00 | SUCCESS |
| 01 | TOKEN ALREADY EXISTS IN DATABASE - CARD EXPIRY UPDATED. |
| 02 | PREAUTH TRANSACTION FAILED, HOWEVER TOKEN WAS STILL  GENERATED. |

The following responses indicate a non-successful result:

|  |  |
| --- | --- |
| **Response Code** | **Response Text** |
| 20 | INSUFFICIENT PARAMETERS SUPPLIED TO PERFORM FORM SUBMISSION |
| 21 | INTERNAL ERROR; PLEASE CONTACT SUPPORT. |
| 22 | INVALID CLIENTID USED WITH PROVIDED SSL CLIENT CERTIFICATE. |
| 23 | UNABLE TO FIND AND DELETE STORED CARD WITH SUPPLIED TOKEN. |
| 24 | UNABLE TO UPDATE STORED CARD - INVALID EXPIRY OR FAILED PREAUTH. |
| 25 | UNABLE TO FIND AND UPDATE STORED CARD WITH SUPPLIED TOKEN |
| 26 | UNABLE TO FIND STORED CARD WITH SUPPLIED TOKEN |
| 27 | PREAUTH TRANSACTION FAILED, NO TOKEN GENERATED |
| 28 | ACTION NOT PERMITTED FOR SUPPLIED CLIENTID. |
| 29 | TOO MANY RETRIEVAL ATTEMPTS - PLEASE CONTACT SUPPORT |
| 30 | INVALID CARD DATA SUPPLIED. |

# 6. Making Payments Using Tokens

In order to make payments using tokens you need to use Real Time Payments option provided by the same API.

Please note that you need to setup your credentials in the same manner as decribed in Section 1 page 3.

//// JAVA ////

PaymentRealTimeRequest realTimeRequest = new PaymentRealTimeRequest();

CreditCard creditCard = new CreditCard();

// Enter TOKEN instead of actual card number

creditCard.setNumber("4564456687594564");

creditCard.setExpiry("1225"); \*

realTimeRequest.setClientId(10000000);

realTimeRequest.setTransactionType(TransactionType.PURCHASE);

realTimeRequest.setCreditCard(creditCard);

// $10.10 dollars as the amount has to entered in cents

//r”AUD” is the currency type . If this is not set then the default currency based

// on the country of the acquiring bank will apply

realTimeRequest.setTransactionAmount(new TransactionAmount(1010,”AUD”));

realTimeRequest.setClientRef("merchant\_reference");

realTimeRequest.setComment("merchant\_additional\_data");

PaymentRealTimeResponse realTimeResponse = client.payment().realTime(realTimeRequest);

//// .NET ////

PaymentRealTimeRequest realTimeRequest = new PaymentRealTimeRequest();

CreditCard creditCard = new CreditCard();

// TOKEN

creditCard.number = "4564456687594564";

creditCard.expiry = "1225";

realTimeRequest.clientId = 10000000;

realTimeRequest.transactionType = Enums.TransactionType.PURCHASE.ToString();

realTimeRequest.creditCard = creditCard;

// $10.10 dollars

//r”AUD” is the currency type . If this is not set then the default currency based

// on the country of the acquiring bank will apply

realTimeRequest.transactionAmount = new TransactionAmount(1010,”AUD”);

realTimeRequest.clientRef = "merchant\_reference";

realTimeRequest.comment = "merchant\_additional\_data";

PaymentRealTimeResponse realTimeResponse = client.payment.realTime(realTimeRequest);

//// PHP ////

$realTimeRequest = new PaymentRealTimeRequest();

$creditCard = new CreditCard();

// TOKEN

$creditCard->setNumber("4564456687594564");

$creditCard->setExpiry("1225");

$realTimeRequest->setClientId(10000000);

$realTimeRequest->setTransactionType(TransactionType::$PURCHASE);

$realTimeRequest->setCreditCard($creditCard);

// $10.10 dollars

//r”AUD” is the currency type . If this is not set then the default currency based

// on the country of the acquiring bank will apply

$transactionAmount = new TransactionAmount(1010,”AUD”);

$realTimeRequest->setTransactionAmount($transactionAmount);

$realTimeRequest->setClientRef("merchant\_reference");

$realTimeRequest->setComment("merchant\_additional\_data");

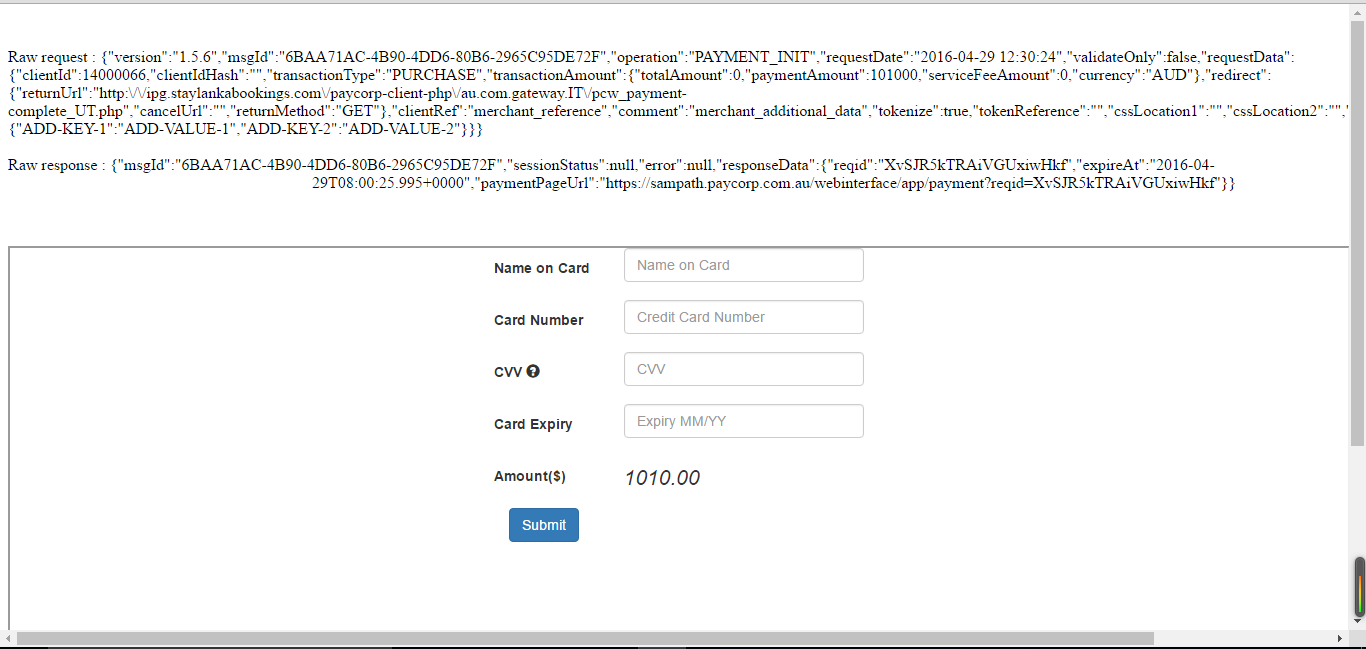
$realTimeResponse = $client->getPayment()->realTime($realTimeRequest);

# 

# 7. Apendix

7.1. FAQs

1. How to remove echo / raw request in PHP API ?

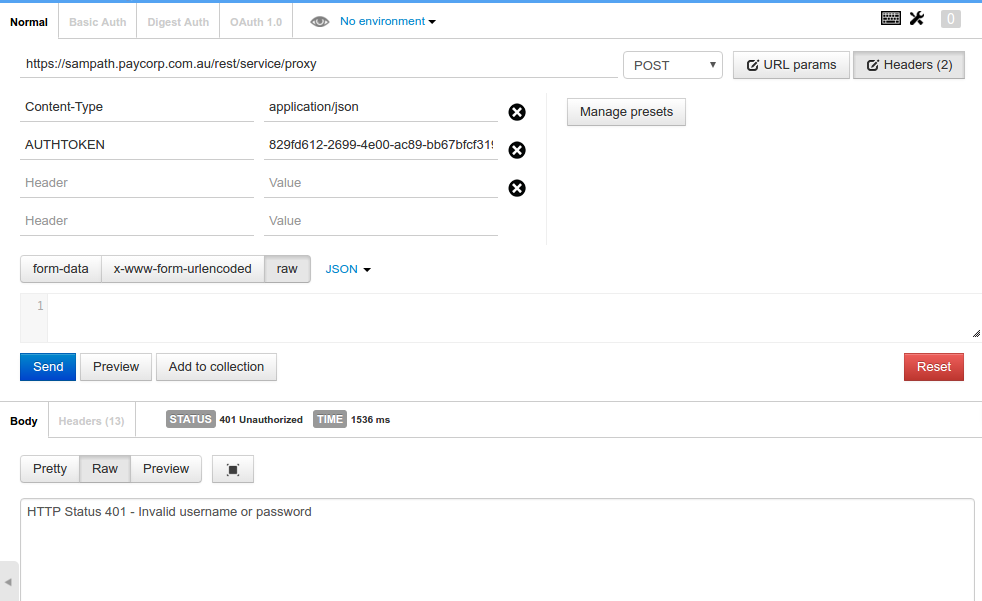


**Solution :** Please comment all the echo's in following php file which can be found on Api

**Step 1 :** Go to directory 'au.com.gateway.client.facade'

**Step 2 :** Comment all echo's in this file 'BaseFacade.php'

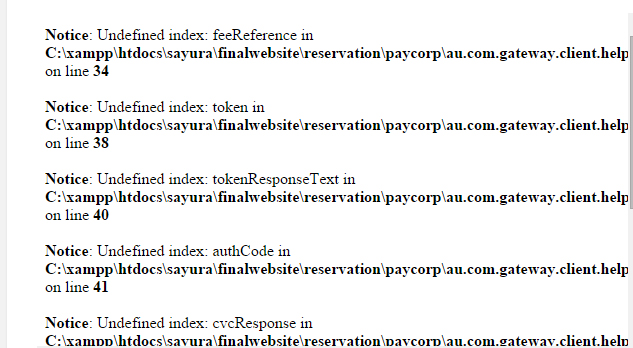
2. Invalid user name or Password ?



**Solution :** This issue occurred when entered details was wrong recheck if the following client details and proceed with the transaction

* Clientid
* HMAC
* Authtoken

3. How to remove notices in Response page



**Solution :**

**step 1** Go to directory au.com.gateway.IT

**step 2** open file : pcw\_payment-complete\_UT.php

**step** **3** open file : pcw\_payment-init\_UT.php

**step 4** Edit line **ini\_set('display\_errors', 1); replace 1 with 0** inpayment init and payment complete files

4. Overcome access denied error from the gateway.

**Raw request :** {"version":"1.5.6","msgId":"5446CCEC-0758-4574-912B-9FFAC7013199","operation":"PAYMENT\_INIT","requestDate":"2016-03-14 10:38:01","validateOnly":false,"requestData":{"clientId":14000192,"clientIdHash":"","transactionType":"PURCHASE","transactionAmount":{"totalAmount":0,"paymentAmount":1010,"serviceFeeAmount":0,"currency":"LKR"},"redirect":{"returnUrl":"http:\/\/localhost\/NEWTEST\/paycorp-paycorp-client-php\/[au.com.gateway.IT](http://au.com.gateway.it/)\/pcw\_payment-complete\_UT.php","cancelUrl":"","returnMethod":"GET"},"clientRef":"merchant\_reference","comment":"merchant\_additional\_data","tokenize":false,"tokenReference":"","cssLocation1":"","cssLocation2":"","useReliability":true,"extraData":{"ADD-KEY-1":"ADD-VALUE-1","ADD-KEY-2":"ADD-VALUE-2"}}}  
  
**Raw response :** {"msgId":null,"sessionStatus":null,"error":{"code":"**ACCESS\_DENIED**","text":"Access denied"},"responseData":null}

**Solution :**

This error generate when user try with incorrect client details

* ClientID
* HMAC
* Authtoken

Yous can Avoid access denied error while using correct user credentials

5. How to remove 22 invalid client ID used with provided SSL client certificate?

* Please set your tokenization parameter as false to over come this issue

6. How to pass the amount in Iframe and redirection .eg with cents or without?

7. How to set CSS in I frame ?

**step 1** Go to directory au.com.gateway.IT

**step 2** open file :pcw\_payment-init.php

initRequest.setCssLocation1("https://testres.gabos.com/css/common/pg/paycorp/sbipg.css");