



PesaPal Integration - PHP

1. Include

OAuth.php – Class that helps in constructing the OAuth Request

```
<?php
include_once('OAuth.php');
?>
```

2. Assign Variables

1. *token* and *params* – null
2. *consumer_key* – merchant key issued by PesaPal to the merchant
3. *consumer_secret* – merchant secret issued by PesaPal to the merchant
4. *signature_method* (leave as default) – `new OAuthSignatureMethod_HMAC_SHA1()` ;
5. *iframelink* – the link that is passed to the iframe pointing to the PesaPal server

```
<?php
$token = $params = NULL;
$consumer_key = 'Your PesaPal Merchant Consumer Key'; //Register a merchant account on
//demo.pesapal.com and use the merchant key for testing.
//When you are ready to go live make sure you change the key to the live account
//registered on www.pesapal.com!
$consumer_secret = 'Your PesaPal Merchant Consumer Secret'; // Use the secret from your
// test account on demo.pesapal.com. When you are ready to go live make sure you
//change the secret to the live account registered on www.pesapal.com!
$signature_method = new OAuthSignatureMethod_HMAC_SHA1();
$iframelink = 'http://demo.pesapal.com/api/PostPesapalDirectOrderV4'; //change to
//https://www.pesapal.com/API/PostPesapalDirectOrderV4 when you are ready to go live!
?>
```

3. Assign Form Details

Assign form details passed to ***pesapal-iframe.php*** from ***shopping-cart-form.php*** to the specified variables.

```
<?php
//get form details
$amount = $_POST['amount'];
$amount = number_format($amount, 2); //format amount to 2 decimal places
$desc = $_POST['description'];
$type = $_POST['type']; //default value = MERCHANT
$reference = $_POST['reference']; //unique order id of the transaction, generated by merchant
$first_name = $_POST['first_name']; //optional
$last_name = $_POST['last_name']; //optional
$email = $_POST['email'];
$phonenumber = ''; //ONE of email or phonenumber is required
?>
```

4. Define the *callback_url*



This is the full url pointing to the page the iframe redirects to after processing the order on pesapal.com

5. Construct the *post_xml* variable

The format is standard so no editing is required. Encode the variable using `htmlentities`.

```
<?php
$callback_url = 'http://www.test.com/redirect.php'; //redirect url, the page that will handle
the response from pesapal.
$post_xml = "<?xml version='1.0' encoding='utf-
8'><PesapalDirectOrderInfo xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance' xmlns:xsd='http://www.w3.org/2001/XMLSchema'
Amount='\".$amount.\"' Description='\".$desc.\"' Type='\".$type.\"'
Reference='\".$reference.\"' FirstName='\".$first_name.\"'
LastName='\".$last_name.\"' Email='\".$email.\"'
PhoneNumber='\".$phonenumber.\"' xmlns='http://www.pesapal.com' />";
$post_xml = htmlentities($post_xml);
?>
```

6. Construct the OAuth Request url

Using the OAuth class included construct the oauth request url using the parameters declared above (the format is standard so no editing is required).

```
<?php
$consumer = new OAuthConsumer($consumer_key, $consumer_secret);
//post transaction to pesapal
$iframe_src = OAuthRequest::from_consumer_and_token($consumer, $token, "GET",
$iframe_link, $params);
$iframe_src->set_parameter("oauth_callback", $callback_url);
$iframe_src->set_parameter("pesapal_request_data", $post_xml);
$iframe_src->sign_request($signature_method, $consumer, $token);
?>
```

7. Display the iframe

Pass `$iframe_src` as the iframe's src.

```
<iframe src="<?php echo $iframe_src;?>" width="100%" height="620px"
scrolling="auto" frameborder="0">
<p>Unable to load the payment page</p>
</iframe>
```

8. Store

Once the payment process has been completed by the user, PesaPal will redirect to your site using the url you assigned to `$callback_url`, along with the following query string parameters:

1. *pesapal_merchant_reference* – this is the same as `$reference` (unique order id) that you posted to PesaPal
2. *pesapal_transaction_tracking_id* – this is a unique id for the transaction on PesaPal that you can use to track the status of the transaction later

Store the *pesapal_transaction_tracking_id* in your database against the order.



```
<?php
$reference = null;
$pesapal_tracking_id = null;

if(isset($_GET['pesapal_merchant_reference']))
    $reference = $_GET['pesapal_merchant_reference'];

if(isset($_GET['pesapal_transaction_tracking_id']))
    $pesapal_tracking_id = $_GET['pesapal_transaction_tracking_id'];

//store $pesapal_tracking_id in your database against the order with orderid = $reference
...
?>
```

9. Query for Status

Once a transaction has been posted to PesaPal, you can query for its status. A cron job is recommended for this, as the status has to be queried periodically until you no longer receive a PENDING status for the transaction.

Below is sample code that can be used for the cron job. Note: you will have to query all orders for which the payment status is unknown or PENDING, and execute the following code on them.

```
<?php
$token = $params = NULL;
$statusrequest = 'http://demo.pesapal.com/api/querypaymentstatus'; //change to
// https://www.pesapal.com/api/querypaymentstatus when you are ready to go live!
$consumer = new OAuthConsumer('/*your consumer key*/', /*your consumer secret*/);

$reference=/* the unique order id that was passed in as reference when posting the order*/;
$transaction_id= /* the pesapal_transaction_tracking_id from the database for this order*/;

//get transaction status
$request_status = OAuthRequest::from_consumer_and_token($consumer, $token,
"GET", $statusrequest, $params);
$request_status->set_parameter("pesapal_merchant_reference", $reference);
$request_status->set_parameter("pesapal_transaction_tracking_id",
$transaction_id);

$request_status->sign_request($this->signature_method, $consumer, $token);

//curl request
$ajax_req = new XMLHttpRequest();
$ajax_req->open("GET", $request_status);
$ajax_req->send();

//if curl request successful
if($ajax_req->status == 200){
    $values = array();
    $elements = split("=", $ajax_req->responseText);
    $values[$elements[0]] = $elements[1];
}
```



```
//transaction status
$status = $values['pesapal_response_data'];
switch($status){
    case "PENDING":
        //update order record in database
        break;
    case "COMPLETED":
        //update order record in database
        break;
    case "FAILED":
        //update order record in database
        break;
}
?>
```