

PayPal Integration in PHP



1. Create a Database, Tables and Dump Sample Data

To create database run the following query in MySQL.

```
CREATE DATABASE shopping;
```

To create the tables run the following queries.

```
CREATE TABLE `products` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `name` varchar(250) NOT NULL,  
  `code` varchar(100) NOT NULL,  
  `price` double(9,2) NOT NULL,  
  `image` varchar(250) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `code` (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Note: I have already attached the SQL file of this table with dummy data, just download the complete zip file of this tutorial.

```
CREATE TABLE `payment_info` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `item_number` varchar(255) NOT NULL,  
  `item_name` varchar(255) NOT NULL,
```

```

`payment_status` varchar(255) NOT NULL,

`amount` double(10,2) NOT NULL,

`currency` varchar(255) NOT NULL,

`txn_id` varchar(255) NOT NULL,

`create_at` timestamp NOT NULL DEFAULT current_timestamp(),

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Dump sample data into ``products`` table.

```

INSERT INTO `products` (`id`, `name`, `code`, `price`, `image`) VALUES

(1, 'Laptop Core i5', 'Laptop01', 100.00, 'product-
images/laptop.jpg'),

(2, 'Laptop Bag', 'Bag01', 10.00, 'product-images/laptop-bag.jpg'),

(3, 'iPhone X', 'iphone01', 50.00, 'product-images/iphone.jpg');

```

2. Create a config.php File & Make Necessary Changes

Create a **config.php** file, copy paste the following code and make necessary changes like update your database credentials, database name, username, password, also update your PayPal business email, return, cancel, notify URLs etc.

```

<?php

// Database Configuration

define('DB_HOST', 'localhost');

```

```
define('DB_NAME', 'Your Database Name');

define('DB_USERNAME', 'Your Database Username');

define('DB_PASSWORD', 'Your Database Password');

// PayPal Configuration

define('PAYPAL_EMAIL', 'Your PayPal Business Email');

define('RETURN_URL', 'https://www.your-website.com/return.php');

define('CANCEL_URL', 'https://www.your-website.com/cancel.php');

define('NOTIFY_URL', 'https://www.your-website.com/ipn.php');

define('CURRENCY', 'USD');

define('SANDBOX', TRUE); // TRUE or FALSE

define('LOCAL_CERTIFICATE', FALSE); // TRUE or FALSE

if (SANDBOX === TRUE){

    $paypal_url = "https://www.sandbox.paypal.com/cgi-bin/webscr";

}else{

    $paypal_url = "https://www.paypal.com/cgi-bin/webscr";

}

// PayPal IPN Data Validate URL

define('PAYPAL_URL', $paypal_url);
```

3. Create an index.php File & Display Some Products to Buy

Create an **index.php** file, copy paste the following code in it. This page will display all products from `products`` table, so that user can easily buy any product.

```
<?php

require_once('dbclass.php');

$db = new DB;

$db->query("SELECT * FROM `products`");

$products = $db->resultSet();

$db->close();

?>

<html>

<head>

<title>PayPal Payment Integration in PHP</title>

<link rel='stylesheet' href='css/style.css' type='text/css'
media='all' />

</head>

<body>

<div style="width:700px; margin:50 auto;">

<h2>PayPal Payment Integration in PHP</h2>
```

```
<?php
if( !empty($products) )
{
    foreach($products as $product)
    {
?>
        <div class='product_wrapper'>
            <div class='image'><img src='<?php echo $product['image']; ?>'
/>
            </div>
            <div class='name'><?php echo $product['name']; ?></div>
            <div class='price'>$<?php echo $product['price']; ?></div>
            <form method='post' action='<?php echo PAYPAL_URL; ?>'>
                <!-- PayPal business email to collect payments -->
                <input type='hidden' name='business'
                    value='<?php echo PAYPAL_EMAIL; ?>'>
                <!-- Details of item that customers will purchase -->
                <input type='hidden' name='item_number'
                    value='<?php echo $product['code']; ?>'>
    }
}
```

```
<input type='hidden' name='item_name'
      value='<?php echo $product['name']; ?>'>
<input type='hidden' name='amount'
      value='<?php echo $product['price']; ?>'>
<input type='hidden' name='currency_code'
      value='<?php echo CURRENCY; ?>'>
<input type='hidden' name='no_shipping' value='1'>

<!-- PayPal return, cancel & IPN URLs -->
<input type='hidden' name='return'
      value='<?php echo RETURN_URL; ?>'>
<input type='hidden' name='cancel_return'
      value='<?php echo CANCEL_URL; ?>'>
<input type='hidden' name='notify_url'
      value='<?php echo NOTIFY_URL; ?>'>

<!-- Specify a Pay Now button. -->
<input type="hidden" name="cmd" value="_xclick">
<button type='submit' class='pay'>Pay Now</button>

</form>
```

```
        </div>

<?php
    }
}
?>

</div>

</body>

</html>
```

4. Create a notify.php File, Handle IPN & Insert Payment Record in DB

Create a **notify.php** file, copy paste the following code in it. This page will handle all PayPal Instant Payment Notification (IPN), validate the data and add record into `payment_info` table.

```
<?php

require_once('dbclass.php');

/*
Read POST data

reading posted data directly from $_POST causes serialization
issues with array data in POST.

Reading raw POST data from input stream instead.
*/
```

```
define("IPN_LOG_FILE", "ipn.log");

$raw_post_data = file_get_contents('php://input');

$raw_post_array = explode('&', $raw_post_data);

$myPost = array();

foreach ($raw_post_array as $keyval) {

    $keyval = explode ('=', $keyval);

    if (count($keyval) == 2)

        $myPost[$keyval[0]] = urldecode($keyval[1]);

}

// Build the body of the verification post request,
// adding the _notify-validate command.

$req = 'cmd=_notify-validate';

if(function_exists('get_magic_quotes_gpc')) {

    $get_magic_quotes_exists = true;

}

foreach ($myPost as $key => $value) {

    if($get_magic_quotes_exists == true && get_magic_quotes_gpc()
== 1) {

        $value = urlencode(stripslashes($value));

    } else {
```



```
        $value = urlencode($value);

    }

    $req .= "&$key=$value";
}

/*
Post IPN data back to PayPal using curl to
validate the IPN data is valid & genuine
Anyone can fake IPN data, if you skip it.
*/

$ch = curl_init(PAYPAL_URL);

if ($ch == FALSE) {

    return FALSE;

}

curl_setopt($ch, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);

curl_setopt($ch, CURLOPT_POST, 1);

curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);

curl_setopt($ch, CURLOPT_POSTFIELDS, $req);

curl_setopt($ch, CURLOPT_SSLVERSION, 6);

curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 1);
```

```
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);

curl_setopt($ch, CURLOPT_FORBID_REUSE, 1);

/*
This is often required if the server is missing a global cert
bundle, or is using an outdated one.

Please download the latest 'cacert.pem' from
http://curl.haxx.se/docs/caextract.html
*/

if (LOCAL_CERTIFICATE == TRUE) {

    curl_setopt($ch, CURLOPT_CAINFO, __DIR__ . "/cert/cacert.pem");

}

// Set TCP timeout to 30 seconds

curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 30);

curl_setopt($ch, CURLOPT_HTTPHEADER, array(

    'Connection: Close',

    'User-Agent: PHP-IPN-Verification-Script'

));
```

```
$res = curl_exec($ch);

// cURL error

if (curl_errno($ch) != 0){

    curl_close($ch);

    exit;

} else {

    curl_close($ch);

}

/*

 * Inspect IPN validation result and act accordingly

 * Split response headers and payload, a better way for strcmp

 */

$tokens = explode("\r\n\r\n", trim($res));

$res = trim(end($tokens));

if (strcmp($res, "VERIFIED") == 0 || strcasecmp($res, "VERIFIED") == 0) {

    // assign posted variables to local variables

    $item_number = $_POST['item_number'];

    $item_name = $_POST['item_name'];
```

```
$payment_status = $_POST['payment_status'];

$amount = $_POST['mc_gross'];

$currency = $_POST['mc_currency'];

$txn_id = $_POST['txn_id'];

$receiver_email = $_POST['receiver_email'];

// $payer_email = $_POST['payer_email'];

// check that receiver_email is your PayPal business email
if (strtolower($receiver_email) != strtolower(PAYPAL_EMAIL)) {

    error_log(date('[Y-m-d H:i e] ').

        "Invalid Business Email: $req" . PHP_EOL, 3,
IPN_LOG_FILE);

    exit();

}

// check that payment currency is correct
if (strtolower($currency) != strtolower(CURRENCY)) {

    error_log(date('[Y-m-d H:i e] ').

        "Invalid Currency: $req" . PHP_EOL, 3,
IPN_LOG_FILE);

    exit();
```

```

}

//Check Unique Transcation ID

$db = new DB;

$db->query("SELECT * FROM `payment_info` WHERE
txn_id=:txn_id");

$db->bind(':txn_id', $txn_id);

$db->execute();

$unique_txn_id = $db->rowCount();

if(!empty($unique_txn_id)) {

    error_log(date('[Y-m-d H:i e] ').

        "Invalid Transaction ID: $req" . PHP_EOL, 3,
IPN_LOG_FILE);

    $db->close();

    exit();

}else{

    $db->query("INSERT INTO `payment_info`

        (`item_number`, `item_name`, `payment_status`,

            `amount`, `currency`, `txn_id`)

            VALUES

```

```
        (:item_number, :item_name, :payment_status,
        :amount, :currency, :txn_id)");

$db->bind(":item_number", $item_number);

$db->bind(":item_name", $item_name);

$db->bind(":payment_status", $payment_status);

$db->bind(":amount", $amount);

$db->bind(":currency", $currency);

$db->bind(":txn_id", $txn_id);

$db->execute();

/* error_log(date('[Y-m-d H:i e] ').
"Verified IPN: $req ". PHP_EOL, 3, IPN_LOG_FILE);

*/

}

$db->close();

} else if (strcmp($res, "INVALID") == 0) {

    //Log invalid IPN messages for investigation
    error_log(date('[Y-m-d H:i e] ').
        "Invalid IPN: $req" . PHP_EOL, 3, IPN_LOG_FILE);

}
```

```
?>
```

5. Create a cancel.php File

Create a **cancel.php** file, copy paste the following code in it. Customer will be redirected here if he/she cancels payment from PayPal payment page.

```
<h1>Sorry! Your PayPal Payment has been cancelled.</h1>
```

6. Create a return.php File

Create a **return.php** file, copy paste the following code in it. Customer will be redirected here when the payment is successful.

```
<html>

<head>

<title>Payment Confirmed</title>

</head>

<body>

    <div style="width:700px; margin:50 auto;">

        <h1>Your paymeny has been received successfully.<br /> Thank
you!</h1>

    </div>

</body>

</html>
```

7. Create a style.css File

Create a **style.css** file, copy paste the following code in it. This is the stylesheet of your index page where all products are displayed nicely.

```
body {  
  
    font-family: Arial, sans-serif;  
  
    line-height: 1.6;  
  
}  
  
.product_wrapper {  
  
    float:left;  
  
    padding: 10px;  
  
    text-align: center;  
  
    }  
  
.product_wrapper:hover {  
  
    box-shadow: 0 0 0 2px #e5e5e5;  
  
    cursor:pointer;  
  
    }  
  
.product_wrapper .name {  
  
    font-weight:bold;  
  
    }  
  
.product_wrapper .pay {  
  
    text-transform: uppercase;
```



```
background: #F68B1E;

border: 1px solid #F68B1E;

cursor: pointer;

color: #fff;

padding: 8px 40px;

margin-top: 10px;
}

.product_wrapper .pay:hover {

    background: #f17e0a;

border-color: #f17e0a;
}
```