

Integrar PayPal Express Checkout con PHP



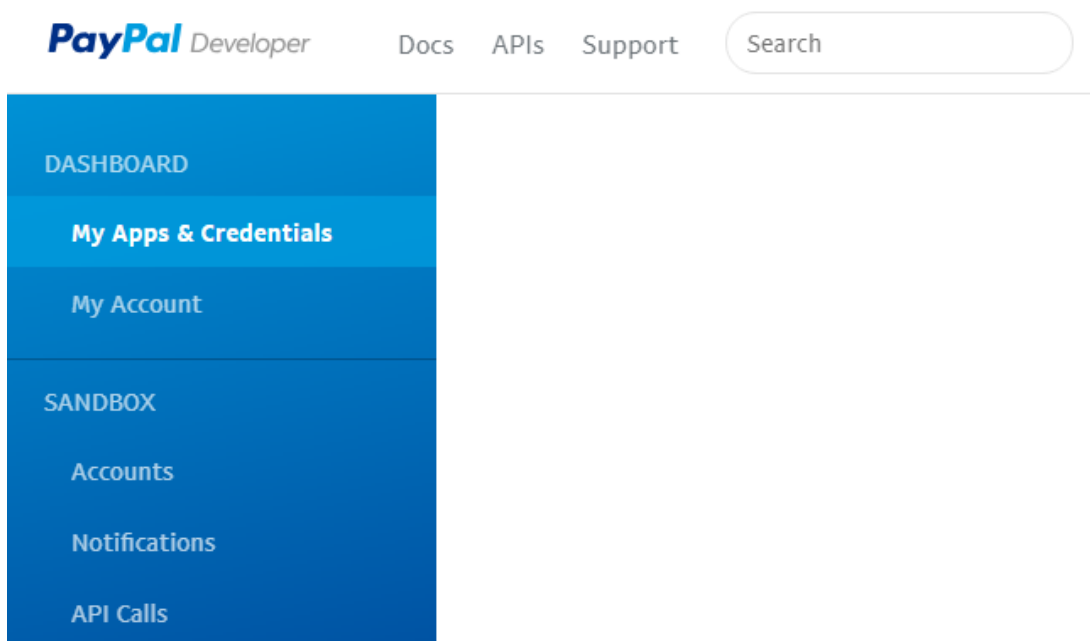
Dentro del mundo del **comercio electrónico**, disponer de varios sistemas de pago nos puede ayudar a conseguir un mayor número de ventas. ¿Por qué centrarnos únicamente en un método si podemos ofrecer varias alternativas a los usuarios? Dentro de todos estos métodos de pago que podemos utilizar, PayPal se ha convertido en uno de los más importantes y que no debería faltar en ninguna tienda online. Se viene utilizando desde hace muchos años, pero no hace mucho que PayPal empezó a ofrecer la posibilidad de utilizar su función PayPal Express Checkout. Se trata de una forma de pago sencilla y seguro que mantiene a los usuarios en el **sitio web** o aplicación móvil durante el proceso de pago.

Como son muchos los beneficios que ofrece este sistema de pago, hoy queremos mostraros el proceso para integrar PayPal Express Checkout en vuestra página web utilizando PHP.

Configurar cuenta de prueba

Lo primero de todo será crear una cuenta de PayPal Sandbox (modo desarrollo). Para ello deberemos registrarnos en la url <https://developer.paypal.com>. También será necesario crear una aplicación REST API desde el panel de control de desarrolladores. Cuando creamos una aplicación de este tipo, obtendremos un ID de cliente y su clave secreta para poder integrar PayPal Express Checkout. Lo recomendable es utilizar el modo Sandbox durante el proceso de desarrollo y una vez que estemos seguros de que todo funciona bien, entonces cambiar a modo live.


Para conseguir estos datos, una vez que hayamos entrado en la url anterior, deberemos pulsar en la opción del menú donde se indica **“My Apps & Credentials”**.



Una vez en dentro de esa sección, pulsaremos en el botón donde se indica “**Create App**”.

REST API apps

Create an app to receive REST API credentials for testing and live transactions.

 **Note:** Features available for live transactions are listed in your [account eligibility](#).

Create App

App Name	Type	Actions
----------	------	---------

Create your first app to view it here.

Se nos pedirá un nombre para la aplicación. Pondremos el que queramos.

Application Details

App Name

Sandbox developer account

Pulsaremos en el botón “**Create App**” que aparece en la parte inferior.

SANDBOX API CREDENTIALS

Sandbox account
[REDACTED]

Client ID
[REDACTED]

Secret
[Hide](#)

Note: When you generate a new secret, you still maintain the original secret. The maximum number of client secrets is two. A client secret is either in enabled or disabled state.

Created	Secret	Status	Action
Sep 03, 2018	[REDACTED]	Enabled	...

[Generate New Secret](#)

Por pantalla se nos mostrará los datos asociados a la aplicación creada para integrar PayPal Express Checkout.

Crear archivo configuración

El siguiente paso será crear el archivo config.php donde definiremos la configuración de pruebas y producción para integrar el pago vía PayPal. Para ello, definiremos la constante “**ProPayPal**” que podrá tener los valores 0 y 1. Cuando se indica el primero, significará que está en modo desarrollo. Si por el contrario le asignamos el valor 1, entonces estaremos indicando que estamos en modo real. Lo deberemos tener en valor 0 mientras estemos desarrollando la integración y hayamos hecho todas las pruebas.

El código de este archivo será el siguiente:

```
<?php
define('ProPayPal', 0);
if(ProPayPal) {
    define("PayPalClientId", "*****");
    define("PayPalSecret", "*****");
    define("PayPalBaseUrl", "https://api.paypal.com/v1/");
    define("PayPalENV", "production");
} else {
    define("PayPalClientId", "XXXXXXXXXXXXXXXXXXXXXXXX");
    define("PayPalSecret", "XXXXXXXXXXXXXXXXXXXXXXXX");
}
```

```

define("PayPalBaseUrl", "https://api.sandbox.paypal.com/v1/");
define("PayPalENV", "sandbox");
}
?>

```

En las variables “**PayPalClientId**” y “**PayPalSecret**” sustituiremos las “**X**” por la información que nos haya proporcionado PayPal al crear la aplicación, tal y como hemos visto en el paso anterior.

Mostrar detalles del artículo y botón de pago

En nuestro archivo index.php lo que haremos será mostrar los detalles del producto, su precio y el botón de pago con PayPal Express. Esto último lo tendremos dentro del archivo “**paypalCheckout.php**” que explicaremos en el punto siguiente.

El código del index.php quedaría de la siguiente forma:

```

<?php
include('config.php');
$productName = "Producto demostración";
$currency = "EUR";
$productPrice = 10;
$productId = 587965;
$orderNumber = 567;
?>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-
theme.min.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scri
pt>

<title>Demo Paypal Express Checkout</title>
</head>

<body>
<div class="container">
    <h2>Demo Paypal Express Checkout Demo con PHP</h2>
    <br>
    <table class="table">
        <tr>
            <td style="width:150px"></td>
            <td style="width:150px"><?php echo $productPrice; ?> €</td>
            <td style="width:150px">
                <?php include 'paypalCheckout.php'; ?>
            </td>
        </tr>
    </table>

```

```
</div>
</body>
```

Si os fijáis al principio del archivo, hemos declarado varias variables con la información del producto, precio, id del producto y el número del pedido. Con estas variables lo que pretendemos es simular la realización de un pedido.

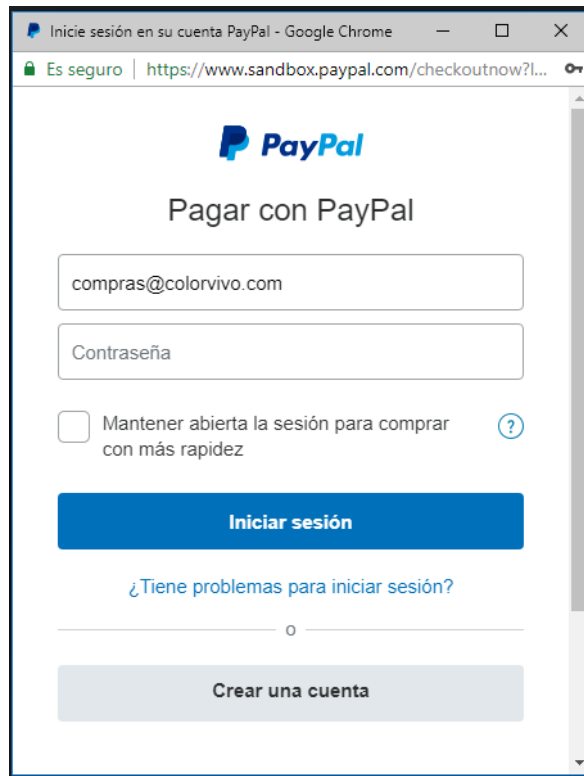
Configuración PayPal Express Checkout

En el archivo “**paypalCheckout.php**”, será donde configuremos PayPal Express Checkout que se encargará de mostrar el botón para realizar el pago y la respuesta del proceso. En este archivo también se hará la llamada al fichero “**orderDetails.php**” que devolverá los detalles del proceso de pago con los valores requeridos para verificarlo.

El código será el siguiente:

```
<div id="paypal-button-container"></div>
<div id="paypal-button"></div>
<script src="https://www.paypalobjects.com/api/checkout.js"></script>
<script>
paypal.Button.render({
  env: '<?php echo PayPalENV; ?>',
  client: {
    <?php if(ProPayPal) { ?>
    production: '<?php echo PayPalClientId; ?>'
    <?php } else { ?>
    sandbox: '<?php echo PayPalClientId; ?>'
    <?php } ?>
  },
  payment: function (data, actions) {
    return actions.payment.create({
      transactions: [{
        amount: {
          total: '<?php echo $productPrice; ?>',
          currency: '<?php echo $currency; ?>'
        }
      }]
    });
  },
  onAuthorize: function (data, actions) {
    return actions.payment.execute()
      .then(function () {
        window.location = "<?php echo PayPalBaseUrl
?>orderDetails.php?paymentID="+data.paymentID+"&payerID="+data.payerID+"&token="
+data.paymentToken+"&pid=<?php echo $productId; ?>";
      });
    }
  }, '#paypal-button');
</script>
```

Verificar los detalles del proceso de pago



El archivo “**orderDetails.php**” será el encargado de recoger los detalles devueltos por el proceso de pago para conocer si se ha completado de forma correcta o no. Los datos que recibiremos serán el token de la operación, el id del pago, el id del pagador y el id del producto. El código de este archivo es el siguiente:

```

if(!empty($_GET['paymentID']) && !empty($_GET['payerID']) &&
!empty($_GET['token']) && !empty($_GET['pid'])) {
    $paymentID = $_GET['paymentID'];
    $payerID = $_GET['payerID'];
    $token = $_GET['token'];
    $pid = $_GET['pid'];
    ?>
    <div class="alert alert-success">
        <strong>Success!</strong> Your order processed successfully.
    </div>
    <table>
        <tr>
            <td>Payment Id: <?php echo $paymentID; ?></td>
            <td>Payer Id: <?php echo $payerID; ?></td>
            <td>product Id: <?php echo $pid; ?></td>
            <td><?php echo ; ?></td>
        </tr>
    </table>
    <?php
} else {
    header('Location:index.php');
}

```

Aquí lo que hacemos es comprobar que las variables “**paymentID**”, “**payerID**”, “**token**” y “**pid**” no estén vacías. Si tienen valor, significará que el pago se ha realizado correctamente. Si por el contrario las variables son devueltas vacías, se ha producido un error y el pago no se ha llevado a cabo.

A lo largo de nuestro WhitePaper, hemos visto como configurar el método de pago PayPal Express Checkout en nuestra [página web](#) utilizando lenguaje de programación PHP.