

Inicializar cURL en PHP



Ahora que ya tenemos una idea de lo que es cURL, vamos a ver como emplearlo desde PHP para leer o escribir datos desde nuestra aplicación en un servidor remoto al que podamos acceder por alguno de los protocolos que cURL emplea. Como hemos visto en el artículo anterior, son suficientes para cubrir casi cualquier necesidad que se nos pueda presentar. Y digo casi, porque siempre habrá algún repositorio de datos antiguo que use algún protocolo descatalogado, en alguna parte. Con eso no se puede luchar. Afortunadamente, no es el caso habitual. En desarrollo web se tiende, cada vez más, a usar estándares aprobados por el consorcio y aceptados por la comunidad (por suerte).

INICIALIZAR UN RECURSO cURL

Para poder usar la librería, debemos crear un recurso que haga uso de las prestaciones de PHP para la gestión de cURL. Un recurso es un manejador, como los que se emplean para lectura o escritura de ficheros, o para conexión con bases de datos, etc, sólo que orientado, específicamente, a la librería cURL.

Para crear este recurso, y configurarlo adecuadamente para que funcione necesitamos contar con los siguientes datos:

Protocolo. El protocolo por el que se establecerá la conexión con el servidor remoto.

Servidor. El nombre del servidor remoto en el que vamos a leer o escribir, o su IP, si no está bajo un nombre de dominio.

Puerto. El número del puerto por el que escucha el servidor.

Ruta. La ruta, dentro del servidor remoto, en la que vamos a leer y/o escribir.

Modo pasivo. Una indicación booleana (**true** o **false**) acerca de si el servidor remoto es pasivo o no. Un servidor FTP (o FTPS, entre otros) puede ser activo (lo que es lo normal), o pasivo. No vamos a entrar aquí en detalles a este respecto, ya que nos llevaría mucho contenido, y se sale del objetivo de estos artículos. No obstante, puedes leer al respecto en [este enlace](#).

Username. El nombre de usuario con el que nos vamos a conectar al servidor remoto.

Password. La contraseña necesaria para acceder al servidor remoto.

Llegados a este punto, podemos empezar definiendo estos valores, que usaremos para la conexión, así:

```
<?php
$protocol = 'ftps://'; // Protocolo del servidor remoto
$server = '245.245.245.245'; // IP del servidor remoto
$port = 990; // Puerto de datos del servidor remoto
$path = ''; // Ruta del servidor donde leeremos o escribiremos
$passive = true; // Un servidor pasivo
$username = 'UsuarioAutorizado'; // El nombre de usuario con el que conectar al servidor remoto
$password = 'password'; // La contraseña de acceso al servidor remoto
$initial_path = '' // La ruta base en la que vamos a trabajar en el servidor remoto
```

El siguiente paso es determinar la URL del servidor remoto, a partir del protocolo, el servidor y la ruta dentro de este en la que trabajaremos, así:

```
$url = $protocol.'/{'.$server.'}/{'.$initial_path.'}';
```

Llegados a este punto, creamos el manejador, a través del cual gestionaremos la librería cURL, así:

```
$curl_handle = curl_init();
```

El método `curl_init()` es nativo de PHP. La sintaxis permite escribirlo sin argumentos, o poner, como único argumento, la variable `$url` que hemos definido antes. Sin embargo, esta también puede fijarse como opción, si no se incluye como argumento. Nosotros vamos a hacerlo así, por lo que el método lo ponemos vacío. En cualquier caso, ya se ha creado el manejador, y se ha almacenado en la variable `$curl_handle`.

CONFIGURANDO cURL

Una vez creado el manejador debemos configurar la conexión por cURL. Para ello existen una serie de opciones proporcionadas por PHP. Aquí vamos a ver qué son esas opciones, y cómo establecerlas.

Las opciones de cURL se identifican mediante unos nombres que empiezan con el prefijo **CURLOPT_**. A estas opciones se les asignan unos valores cuya naturaleza, tipo o estructura dependerá de cada opción específica. Sin embargo, la forma de establecer los valores necesarios en cada opción es, siempre, mediante el método `curl_setopt()`, que recibe tres argumentos: el primero es el recurso, o manejador, que hemos creado con `curl_init()`. Los otros dos argumentos son el nombre de la opción a establecer, y el valor que queremos darle.

La primera opción que tenemos que establecer (y hablamos de "opción" pero esta no es, en modo alguno, opcional), es **CURLOPT_USERPWD**. Le pasaremos el nombre de usuario y la contraseña de acceso al servidor remoto. Con las variables de ejemplo que hemos puesto más arriba, nos quedaría algo así:

```
curl_setopt($curl_handle, CURLOPT_USERPWD, $username.'!'.$password);
```

Otras opciones que se emplean con mucha frecuencia, y que vamos a ver en los ejemplos de estos artículos, son las siguientes:

CURLOPT_SSL_VERIFYPEER. El valor por omisión es `true`. Si le ponemos el valor `false` evitamos que se haga una verificación por SSL en la transferencia de datos. Esto es adecuado cuando el servidor remoto no encripta los datos bajo SSL, porque, si esta opción estuviera activada, eso impediría la transferencia.

CURLOPT_SSL_VERIFYHOST. Esta opción es muy parecida a la anterior, refiriéndose a la comprobación SSL del nombre del servidor. El valor por omisión es `true` pero, si no usamos SSL, o el nombre del servidor no puede identificarse mediante SSL, lo ponemos a `false`. En caso de duda, el valor `false` evitará conflictos en conexiones TLS o de otros tipos.

CURLOPT_FTP_SSL. Esta opción se usa para indicar si se requiere una capa adicional de seguridad (que la biblioteca libcurl gestionará, si la conexión la admite) tanto para las conexiones de control como de datos. Las posibles opciones son: **CURLFTPSSL_NONE** (no se intentará usar SSL); **CURLFTPSSL_TRY** (se intentará usar SSL, pero se ignorará si no se puede emplear); **CURLFTPSSL_CONTROL** (se requiere SSL para la conexión de control); **CURLFTPSSL_ALL** (se requiere SSL para control y transferencia).

CURLOPT_FTPSSLAUTH. Controla el método de autenticación con el servidor remoto. Los posibles valores son: **CURLFTPAUTH_SSL** (comprueba, en primer lugar, la autenticación por SSL); **CURLFTPAUTH_TLS** (comprueba, en primer lugar, la autenticación por TLS); **CURLFTPAUTH_DEFAULT** (es la opción más adecuada, ya que deja que sea cURL el que decida la mejor forma de autenticarse).

CURLOPT_UPLOAD. Permite que cURL suba ficheros al servidor remoto, si le establecemos el valor `true`. En caso contrario la transferencia sólo podrá ser desde el servidor remoto (lectura), pero no se podrán hacer envíos.

CURLOPT_PORT. Es un valor numérico con el puerto por el que viajan los datos al servidor remoto. Se refiere al puerto para transferencia, no para control.

CURLOPT_TIMEOUT. Un valor numérico con el tiempo máximo, en segundos, que debe durar una operación cURL. Si se excede este tiempo, la operación se abortará.

CURLOPT_FTPPORT. El valor que se usará para indicarle a la instrucción FTP PORT la IP a la que se tiene que conectar el servidor remoto. Si se usa un servidor ftp pasivo, este valor deberá indicárnoslo el proveedor de servicios o, en su caso, simplemente no la estableceremos. Si no es ftp pasivo, el valor por defecto ('') es el adecuado para esta opción.

Existen muchas más opciones que podemos configurar, pero las que hemos visto aquí son las más habituales en la mayoría de los casos. La lista completa puedes verla [en este enlace](#).

Este artículo se hace ya demasiado largo. En el siguiente continuaremos viendo el uso de cURL en PHP, así como algunas opciones más que hemos aplazado a propósito, para hablar de ellas en su contexto.