

cURL. Más modos de envío.



En el artículo anterior vimos como subir el contenido de un archivo, situado en la ubicación de nuestra aplicación, a un servidor remoto empleando cURL.

En este artículo vamos a comprender como ocurre ese proceso, y comparar la forma de hacerlo con otro método, para el caso de aquellos archivos que, por su tamaño u otras circunstancias, no puedan ser enviados con tanta facilidad, y como solucionarlo mediante un flujo de datos.

CREAR UN FLUJO DE DATOS

Un flujo de datos, dentro del contexto de PHP, es una manera de canalizar el contenido de un fichero en un archivo temporal, empleando un protocolo nativo del propio lenguaje. Observa el siguiente código:

```
$stream = fopen('php://temp', 'w+'); // Creamos un flujo temporal.  
fwrite($stream, file_get_contents('local_test.txt')); // Grabamos en dicho flujo, el contenido del fichero a subir.  
rewind($stream); // "Rebobinamos" el flujo.  
curl_setopt($curl_handle, CURLOPT_INFILE, $stream); // Asignamos el flujo a CURLOPT_INFILE.  
curl_exec($curl_handle); // Ejecutamos la conexión cURL.  
fclose($stream); // Cerramos el flujo.  
curl_close($curl_handle);
```

Observa que `$stream` es un manejador que crea un archivo, al que llamamos `temp`, bajo una envoltura específica del lenguaje, llamada `php://`. El lenguaje incorpora distintas envolturas para encapsular contenidos de acuerdo con el uso que se les quiera dar. La envoltura `php://` está diseñada, específicamente, para su uso en flujos de datos, como se detalla en [este enlace](#). El uso de `php://temp` crea un archivo temporal (que desaparece cuando se cierra el manejador). La alternativa sería, por ejemplo, usar `php://memory`. Sin embargo, en previsión de que el flujo supere determinado peso (usualmente, 2 Mb), es mejor grabarlo en un archivo temporal.

A continuación se graba el contenido del fichero que vamos a enviar en el flujo de datos, mediante el uso de `fwrite()`. Un flujo de datos se gestiona del mismo modo que un archivo convencional, usando los mismos comandos para grabar y leer en él.

Después nos posicionamos al principio del flujo, con el método `rewind()`. Aunque no es de uso muy habitual, este método, aplicado a un manejador, nos permite posicionar el puntero al principio del archivo al que se refiere dicho manejador.

Luego asignamos el manejador del flujo a la opción `CURLOPT_INFILE`, del mismo modo que, en el artículo anterior, hacíamos con el manejador del fichero. A partir de aquí, el proceso es el mismo, con la única salvedad de cerrar el manejador del flujo.

CONCLUYENDO

Con esto ya tenemos todo lo que necesitamos saber respecto al envío, a servidores remotos de un archivo por cURL. En el próximo artículo procederemos con la lectura desde servidores remotos.