Leer archivos por cURL

Ya sabemos como podemos enviar archivos vía cURL a un servidor remoto, que, conceptualmente, era la parte más compleja. En este artículo vamos a aprender como leer un archivo remoto y grabarlo en el servidor donde tenemos nuestra aplicación.
Los cambios necesarios son mínimos, pero necesarios para poder llevar la lectura a buen puerto.

PRECAUCIONES

Para leer un archivo desde un servidor remoto y grabarlo en el servidor de nuestra aplicación, debemos tomar dos precauciones específicas:
En primer lugar, la ubicación de nuestro servidor donde vayamos a grabar los contenidos que leamos del remoto debe tener permiso de escritura. Esto parece una obviedad, pero, a menudo, se tiende a olvidarlo.
La opción CURLOPT_UPLOAD debe desactivarse (valor false). De lo contrario, se perderá el contenido del archivo en el servidor remoto, y se grabará un archivo vacío. Esto es importante porque esa pérdida del contenido afecta, directamente, al archivo del servidor remoto, y es irreversible. Por lo tanto, antes de iniciar la lectura, y después de inicializar el manejador de cURL, emplearemos el siguiente comando:

```
curl_setopt($curl_handle, CURLOPT_UPLOAD, false);
```

Con estos dos detalles en mente, no tendremos problemas en leer archivos remotos, y grabarlos en nuestro servidor, como se describe en este artículo.

LEYENDO ARCHIVOS REMOTOS

Una vez creada y configurada la conexión cURL, con las precauciones que acabamos de mencionar, viene el momento de efectuar la lectura del contenido de un archivo remoto, y crear, en nuestro propio servicor, un archivo con dicho contenido.
El primer paso es establecer la URL, con la ruta y el nombre del archivo cuyo contenido queremos leer, tal como hacíamos cuando establecíamos un archivo de destino. Lo haremos así:

```
// Valores y opción para la URL de origen
$remote_path = 'origen/'; // La ruta base en la que vamos a trabajar en el servidor remoto
$source_file_name = 'remote_test.docx'; // El nombre del archivo en el servidor remoto cuyo contenido queremos leer.
$url = $protocol.'//'.$server.'/'.$remote_path.$source_file_name; // URL final del archivo en el servidor remoto.
curl_setopt($curl_handle, CURLOPT_URL, $url); // Establecemos la URL final para el archivo en el servidor remoto en cURL.
```

Ahora tenemos que establecer una opción de cURL que, hasta ahora, no habíamos empleado. Se trata de CURLOPT_RETURNTRANSFER, a la que deberemos asignarle el valor true. Lo hacemos así:

```
curl_setopt($curl_handle, CURLOPT_RETURNTRANSFER, true); // Permitimos que se transfieran datos desde el servidor remoto al de nuestra aplicación.
```

Esto permite que los datos que se vayan a recibir puedan alojarse en una cadena, en lugar de ser volcados a la terminal de salida.
Por último, recibimos los datos, con curl_exec(), en una cadena, tal como hemos establecido, y los grabamos en un fichero en nuestro servidor, así:

```
$content = curl_exec($curl_handle);
$dest_file = 'local_test.docx';
$file_handle = fopen($dest_file, "w+");
fputs($file_handle, $content);
fclose($file_handle);
curl_close($curl_handle);
```

ERRORES Y HORRORES

A menudo, cuando se usa cURL, como cuando se emplea cualquier otro recurso de PHP, se producen errores, que debemos ser capaces de identificar y resolver. Los más comunes tienen que ver con el protocolo de comunicación, los puertos, la IP del servidor y, por supuesto, la autenticación. Para identificar un error cuando cURL no funciona como esperamos, el lenguaje nos ofrece dos funciones: curl_errno() y curl_error(). Estas reciben un único argumento, que es el manejador cURL obtenido con curl_init(). La primera función nos devuelve un código numérico del error producido, y la segunda nos devuelve un mensaje descriptivo.
La lista completa de posibles errores está en este enlace, donde puedes investigar cuál es la causa de un posible fallo.

CONCLUYENDO

En este artículo hemos aprendido a usar cURL para leer contenidos de un servidor remoto, y grabarlos en el servidor de nuestra aplicación. En el próximo artículo te mostraré una clase de PHP para gestionar lecturas y escrituras por cURL.