

Almacenando y recuperando datos con archivos (I/O)



6.1 Flujos PHP

Los flujos fueron introducidos en PHP 4.3.0 como una manera de generalizar operaciones con ficheros, redes, compresión de datos, y otras operaciones que comparten un conjunto común de funciones y usos. En su definición más simple, un **flujo** es un objeto de tipo **resource** que exhibe un comportamiento similar al de un flujo. Esto es, puede ser leído o escrito de una manera lineal, y puede usar la función **fseek()** para buscar posiciones arbitrarias dentro del flujo.

Una **envoltura** es un código adicional que le indica al flujo cómo manejar protocolos/codificaciones específicos. Por ejemplo, la envoltura *http* sabe cómo traducir un URL a una petición **HTTP/1.0** para un fichero en un servidor remoto. Existen muchas envolturas incluidas en PHP de manera predeterminada, y también se pueden añadir envolturas personalizadas a un script de PHP usando **stream_wrapper_register()**, o directamente desde una extensión utilizando la referencia de la API. Ya que se puede añadir cualquier clase de envoltura a PHP, no existe un límite establecido sobre lo que se puede hacer con ellas. Para acceder a la lista de las envolturas actualmente registradas, use **stream_get_wrappers()**.

6.1.1 Lectura y escritura de archivos

Un flujo está descrito como: **esquema://objetivo**

- **Esquema(string):** El nombre de la envoltura que se va a usar. Algunos ejemplos incluyen: file, http, https, ftp, ftps, compress.zlib, compress.bz2, y php. Si no se especifica una envoltura, se usará la función predeterminada (normalmente *file://*).
- **Objetivo:** Depende de la envoltura usada. Para flujos relacionados con el sistema de ficheros, normalmente es una ruta y un nombre de fichero. Para flujos relacionados con redes, normalmente es un nombre de host, a menudo con una ruta añadida al final.

6.1.3 Manejando posibles excepciones

Como con cualquier función relacionada con un archivo o con un socket, una operación sobre un flujo puede fallar por varias razones normales (esto es: Incapaz de conectarse al host remoto, archivo no encontrado, etc...). Una llamada relacionada con un flujo puede también fallar porque el flujo no está registrado en el sistema en ejecución. Como con la mayoría de la funciones internas de PHP, si ocurre un error se generará un mensaje **E_WARNING** describiendo la naturaleza del error.

6.1.4 Formateando salida de texto

PHP ofrece una serie de flujos de E/S generales que permite acceder tanto a los flujos de entrada y salida de PHP, a la entrada estándar, a descriptores de ficheros de salida y de errores, a flujos de ficheros temporales en memoria y en disco, y a filtros para poder manipular otros recursos de ficheros según se lee o se escribe en ellos.

php://input

php://input es un flujo de sólo lectura que permite leer datos del cuerpo solicitado. En el caso de peticiones POST, es preferible usar **php://input** en vez de *\$HTTP_RAW_POST_DATA* ya que no depende de directivas **php.ini** especiales. Sin embargo, cuando no se genera automáticamente *\$HTTP_RAW_POST_DATA*, se trata de una alternativa que hace un uso menos intensivo de memoria que activando **always_populate_raw_post_data**. **php://input** no está disponible con **enctype="multipart/form-data"**.

php://output

php://output es un flujo de sólo escritura que permite escribir en el buffer de salida tal como lo hacen **print** y **echo**.