

Comprimir un directorio con PHP



En un aplicativo web con frecuencia se van acumulando, en el servidor, archivos enviados por los usuarios de la aplicación. Pueden ser imágenes, documentos en pdf o rtf, hojas de cálculo, o cualesquiera otros, dependiendo de la finalidad de la aplicación.

Como sea, el caso es que estas aplicaciones que actúan como repositorios dan como resultado, si el sitio tiene muchos usuarios y un elevado tráfico que, en poco tiempo, el volumen de documentos almacenados pueda llegar a comprometer el servicio, o requiera aumentar el costoso espacio de almacenamiento.

La solución puede pasar por empaquetar todos esos documentos en un archivo comprimido, y mover ese archivo a otra ubicación, como puede ser un ordenador local.

También puede ser que un usuario decida recuperar todos los archivos que ha ido colocando en el repositorio a lo largo del tiempo en un único paquete.

En ambos casos, existe una clase que puede ayudarnos a realizar esta tarea de empaquetado. Se trata de DirCompress.

QUÉ ES DirCompress]

DirCompress es una clase de PHP que permite recorrer un directorio que le indicamos de forma recursiva, es decir, recorriendo todos los subdirectorios que contenga, y almacenar todos los archivos que encuentre en un archivo comprimido, en el directorio que nosotros elijamos también. El comprimido (en formato [.zip](#)) mantendrá la estructura original de subdirectorios y archivos que hay en el directorio que queremos empaquetar.

DirCompress es open source, de libre uso y distribución, y es muy eficiente, al tiempo que muy fácil de usar (gratis, eficiente y fácil son tres palabras que me gustan). Puedes descargarla, junto con un ejemplo de uso, [en este enlace](#).

REQUISITOS PREVIOS]

Para usar DirCompress tienes, por supuesto, que tener instalado PHP, preferiblemente en la versión 7.0 o superior. Aunque también funciona con la versión 5, esta última está ya obsoleta en muchas otras funcionalidades que, aunque no vienen al caso, me hacen aconsejarte la actualización. Tu intérprete de PHP debe estar compilado con la librería ZipArchive (o importarla por configuración). En la versión 7 eso no es ningún problema, ya que existe por defecto. Si quieres leer más sobre esta librería, puedes visitar [este enlace](#).

Además, debes asegurar de que el directorio donde vayas a depositar el comprimido tenga los adecuados permisos de escritura para que PHP pueda grabar en el mismo el paquete final. Esto, que parece de cajón, se le pasa por alto a mucha gente, y me han llegado comentarios diciendo que "no funciona". Una vez establecidos los permisos, funciona perfectamente. En la versión actual de la clase DirCompress ni siquiera necesitas, en principio, preocuparte por esto, ya que la clase le otorga permisos al directorio de destino (o lo crea con tales permisos, si no existe), pero no está de más que te asegures de que, efectivamente, tienes tales permisos.

USO DE LA CLASE]

Una vez que dispongas de la clase, bien sea incluida en tu script o cargada con un autoloader (como en el ejemplo de GitHub), necesitarás instanciarla en un objeto. El constructor recibe dos argumentos: el directorio de origen (aquél cuyo contenido quieres empaquetar) y el de destino (aquél dónde deseas almacenar el empaquetado final. Ambos directorios deben ser rutas relativas al directorio donde está el script desde el que estás trabajando, no al que contiene la clase. De este modo, es mucho más fácil establecerlos en cada caso.

Una vez creado el objeto, debemos usar el método [setZipFileName\(\)](#), pasándole, como argumento, el nombre que queremos darle al empaquetado. Podemos añadir la extensión [.zip](#), o no hacerlo. En este caso, la clase lo hará por nosotros.

Por último, emplearemos el método [createZip\(\)](#), lo que recorrerá el directorio de origen, creará el paquete comprimido, y lo

almacenará, con el nombre que le dijimos en el método anterior, en el directorio de destino que le indicamos al constructor.

Opcionalmente, podemos usar el método `makeLink()` para crear un enlace de descarga. Este método recibe, como argumento, el texto que queremos que aparezca en el enlace. Este enlace nos permitirá descargar el comprimido.

La clase es muy sencilla de usar, y está profusamente documentada. Entre eso, y la simplicidad que verás en el ejemplo de GitHub, te será muy sencillo usar esta clase en tu propio trabajo. Y si tienes cualquier duda, pon un mensaje en este artículo, y cualquiera de los miembros de este blog estaremos encantados de ayudarte.