

iranphp articles

عنوان مقاله : کلاس Archive_Tar از مجموعه PEAR
نگارنده : میلاد راستیان
آدرس پست الکترونیک : miladmovie@yahoo.com
تاریخ نگارش :

کلاس Archive_Tar از مجموعه PEAR :

با این کلاس به آدرس http://pear.php.net/package/Archive_Tar خیلی حال کردم یک جا بود که من باید چندتا فایل رو کپی گرفتم و در یک بسته میگذاشتمش یک گوشه !
همون طور که می دونید gz فقط یک فایل رو فشرده می کنه ! یعنی باید اولش اگر یک شاخه یا چند تا فایل رو دارید تبدیل به یک بسته آرشیو کنید (tar) بعد اون رو زیپ کنید البته راه های هم هست که هر دو کار رو با هم انجام داد که آآن بحث ما یک چیز دیگست .
11 تا متد داره که مهمترینش عبارت اند از :

سازنده کلاس	<code>Archive_Tar::Archive_Tar()</code>
اضافه کردن فایل یا شاخه به فایل	<code>Archive_Tar::add()</code>
ایجاد فایل آرشیو	<code>Archive_Tar::create()</code>
باز کردن فایل آرشیو	<code>Archive_Tar::extract()</code>
لیست فایل های موجود در فایل آرشیو را بر می گرداند	<code>Archive_Tar::listContent()</code>

خوشبختانه این کلاس بسیار ارزش مند اصلا نیازی به هیچ کلاسی نداره ولی خوب باید حتما فایل `PEAR.php` در سرور وجود داشته باشه (که معولا وجود داره). خب اینجاست که دیگه باید از خودم یک مثال از خودم در بیارم که همه متدها رو در یک کد توضیح میدم.
فرض کنیم که یک شاخه داریم شبیه شکل زیر :

```
web/index.html
web/images/header.gif
web/images/footer.gif
web/css/style.css
```

```
<?php
//testTAR.php
require_once 'Archive/Tar.php';

//constructor of Archive_Tar
//Parameters $p_compress (2) by default is false, if we set it true the archive file will be
compressed
$file_archive=new Archive_Tar("tarname.tar.gz",true);

//set $files_name as array
$files_name=array();
//set file to archive
$files_name[0]="web/index.html";
//set directory to archive
$files_name[1]="web/css/";

//create tarname.tar.gz with
$file_archive->create($files_name);
//now tarname.tar.gz contain 1 files and 1 directory

//now we add 1 new directory to our archive
$files_name=array();
$files_name[0]="web/images/";

//add that directory to tarname.tar.gz
$file_archive->add($files_name);

//extract file to selectiob folder
$result = $file_archive->extract('extract');
```

```
//print array of file content  
print_r($file_archive->listContent());  
?>
```

فکر نکنم نیاز به توضیح داشته باشه دیگه!؟

در صورتی که ورودی دوم سازنده این کلاس رو مقدار دهی نکنیم این فایل فقط آرشیو می شود و فشرده نمی شه برای اینکه فشرده هم بشه باید مقدار `true` رو برای اون بفرستیم

یک لیست از فایل ها و شاخه های رو که می خواهیم آرشیو بشه در یک آرایه قرار می دهیم و بعد این آرایه رو به متد `create` می دهیم تا یک فایل آرشیو درست کنه. همچنین در صورتی که بخواهیم بعد از ایجا فایل آرشیو، فایل یا شاخه ای رو به فایل آرشیو اضافه کنیم از متد `add` استفاده می کنیم

و در پایان هم با استفاده از متد `extract` و ورودی که بهش می دهیم مشخص می کنیم فایل آرشیو در کجا باز بشه

خط آخر هم به ما آرایه ای می ده که شامل مشخصات فایل های موجود در فایل آرشیو است که نمونه ای از اون رو می تونید در زیر ببینید:

```
Array  
(  
  [0] => Array  
  (  
    [checksum] => 3452  
    [filename] => web/index.html  
    [mode] => 33206  
    [uid] => 0  
    [gid] => 0  
    [size] => 0  
    [mtime] => 1126089004  
    [typeflag] =>  
  )  
  
  [1] => Array  
  (  
    [checksum] => 2854  
    [filename] => web/css/  
    [mode] => 16895  
    [uid] => 0  
    [gid] => 0  
    [size] => 0  
    [mtime] => 1126089594  
    [typeflag] => 5  
  )  
  
  [2] => Array  
  (  
    [checksum] => 3749  
    [filename] => web/css/style.css  
    [mode] => 33206  
    [uid] => 0  
    [gid] => 0  
    [size] => 0  
    [mtime] => 1126089596  
    [typeflag] =>  
  )  
  
  [3] => Array  
  (  
    [checksum] => 3152  
    [filename] => web/images/  
    [mode] => 16895  
    [uid] => 0  
  )  
)
```

```
[gid] => 0
[size] => 0
[mtime] => 1126089010
[typeflag] => 5
)

[4] => Array
(
[checksum] => 4086
[filename] => web/images/header.gif
[mode] => 33206
[uid] => 0
[gid] => 0
[size] => 0
[mtime] => 1126089014
[typeflag] =>
)

[5] => Array
(
[checksum] => 4111
[filename] => web/images/footer.gif
[mode] => 33206
[uid] => 0
[gid] => 0
[size] => 0
[mtime] => 1126089232
[typeflag] =>
)

)
```