PHP : Les fichiers

L3 Informatique - UE Développement Web

David Lesaint

david.lesaint@univ-angers.fr

Janvier 2020

Gestion des fichiers

Notion de flux

Flux (stream)

- Une ressource (type `resource`) accessible linéairement en lecture ou en écriture : fichier, ressource Web, données de BDD, ...
- Identifié avec la syntaxe `scheme://target`
  - `file://*` : accès aux fichiers locaux
  - `http://*` : accès HTTP aux URL
  - `ftp://*` : accès FTP aux URL
  - `zlib://*` : compression de données
  - `phar://*` : archive PHP
  - `ssh2://*` : SSH 2
  - ...

Gestionnaire de protocole (wrapper)

Code PHP pour gérer certains protocoles/encodages liés à un flux.

- eg. le wrapper HTTP convertit une URL en requête HTTP.

Contexte de flux

Contexte de flux

- Un ensemble de paramètres et d'options spécifiques à un wrapper pour contrôler le comportement d'un flux.
- Se crée avec la fonction `resource stream_context_create([array $options [, $params]]`.
- Se passe ensuite en argument aux fonctions de création et d'accès aux flux : `fopen`, `file_get_contents`, `file_put_contents`, ...

Exemple : accès HTTP

Récupération d'une ressource Web par son URL (file_get_contents3.php)

```php
// création des options du contexte de flux
$opts = array(
  'http'=>array(
    'method'=>"GET",
    'header'=>"Accept-language: en\r\n" .
              "Cookie: foo=bar\r\n"
  )
);
// création du contexte de flux
$context = stream_context_create($opts);
// accès au flux "http://www.univ-angers.fr"
// avec envoi par le wrapper PHP pour HTTP
// d'une requête HTTP/1.1
// contenant les entêtes HTTP ci-dessus
$file = file_get_contents('http://www.univ-angers.fr/', false, $context);
echo $file;
```

Gestion des fichiers

Lecture et écriture de fichiers

A l'aide de

- Fonctions : `fopen`, `fread`, `fwrite`, `fclose`, `file_get_contents`, `file_put_contents`, ...
- Deux classes SPL : `SPLFileInfo` et `SPLFileObject`.

Différentes modalités d'accès

- Lecture seule, écriture seule, ou les deux.
- Ecrasement de fichier ou ajout de données par écriture.
- Lecture intégrale, ligne par ligne, caractère par caractère.
- Lecture automatique de données formattées.
- Positionnement de pointeur de lecture/écriture.
- Verrouillage d'accès.
- Copie, renommage, effacement, méta-données.

Lecture de fichier avec `file_get_contents`

`string file_get_contents(string $f [,...])`

Lit le contenu d'un fichier en totalité et le retourne dans une chaîne.

- `$f` : nom du fichier.
- Drapeaux :
  - recherche ou non du fichier dans le `$PATH`.
  - contexte de flux valide (`NULL` pour les fichiers).
  - positionnement du pointeur de lecture.
  - nombre d'octets à lire ou bien lecture intégrale.

Retourne `FALSE` en cas d'erreur.
Emet un `E_WARNING` si fichier introuvable.

Lecture de fichier avec `file_get_contents`

Lecture d'un fichier local (file_get_contents1.php)

```php
$filename=__DIR__.'/html.txt';
$str=file_get_contents($filename);
if ($str===false) {
    throw Exception('could not read');
} else {
    echo $str;
}
```

Lecture d'une page HTML (file_get_contents2.php)

```php
$file=file_get_contents("http://www.univ-angers.fr");
echo $file;
```

Ecriture de fichier avec `file_put_contents`

`file_put_contents(string $f, mixed $d [,...])`

Ecrit dans le fichier spécifié par `$f` la donnée `d` :

- `$d` : chaîne, tableau ou ressource de flux.
- Drapeaux :
  - ou-logique entre constantes : `FILE_USE_INCLUDE_PATH` (recherche dans le `$PATH`), `FILE_APPEND` (concaténation vs. écrasement), `LOCK_EX` (verrou exclusif).
  - contexte de flux (`NULL` pour les fichiers)

Crée le fichier s'il n'existe pas, sinon l'écrase ou y ajoute les données selon l'option `FILE_APPEND`.
Retourne le nombre d'octets écrits ou `FALSE` en cas d'erreur.

Ecriture de fichier avec `file_put_contents`

Ecriture dans un fichier local (file_put_contents1.php)

```php
$fname=__DIR__.'/file_put_contents1.txt';
$tab=array('odyssée', 'de', 'l\'espace', 2001);
file_put_contents($fname,implode(' ',$tab)."\n");
```

Utilisation de `fopen`, `fwrite`, `fclose`

```
resource fopen(string $f, string $mode [,...])
```

- Ouvre un fichier ou une URL spécifiée par `$f`.
- Drapeaux :
  - `$mode` : lecture seule `"r"`, lecture et écriture `"r+"`, écriture seule avec écrasement `"w"`, lecture et écriture avec écrasement `"w+"`, écriture seule par ajout `"a"`, lecture et écriture par ajout `"a+"`.
  - recherche dans le `$PATH`.
  - contexte de flux valide (`NULL` pour les fichiers).

```
int fwrite(resource $h, string $s [,int $n])
```

- Ecrit le contenu de `$s` (maximum `$n` octets) dans le fichier identifié par `$h`.

```
bool fclose(resource $h)
```

- Ferme le fichier identifié par `$h`.

Utilisation de `fopen`, `fwrite`, `fclose`

Exemple de compression de fichier (fopen.php)

```php
/* création d'un fichier compressé (binaire)
 * Le fichier peut être décompressé
 * et lu avec le wrapper compress.zlib stream
 * ou en ligne de commande 'gzip -d foo-bar.txt.gz'
 */
$fp = fopen("compress.zlib://foo-bar.txt.gz", "wb");
if (!$fp) die("Unable to create file.");
fwrite($fp, "This is a test.\n");
fclose($fp);
```

Autres fonctions

Nom	Description
`flock`	dé-/verrouillage d'accès.
`fgetc`	lecture d'un caractère à la fois.
`fgets`	lecture d'une ligne à la fois (délimiteur `\n`).
`fread`	lecture de blocs de taille prédéfinie.
`fseek`	positionnement du pointeur.
`rewind`	positionner le pointeur en début de fichier.
`ftell`	position du pointeur.
`fgetcsv`	lecture d'une ligne à la fois et conversion sous forme de tableau de mots basée sur séparateur prédéfini.
`readfile`	lecture intégrale et affichage sur la sortie standard.

Autres fonctions

Nom	Description
`copy`	copie.
`rename`	renommage.
`unlink`	suppression.
`file_exists`	existence.
`filesize`	taille.
`filetype`	de type fichier ou répertoire.
`is_file`	vérification.
`is_readable`	accessibilité en lecture.
`is_writable`	accessibilité en écriture.
`is_uploaded_file`	test de téléversement.
`fileatime`	dernier accès.
`filemtime`	dernière modification.
`filectime`	dernière modification des permissions.
`realpath`	chemin d'accès.
`basename`	extraction du nom à partir d'un chemin.

La classe SPLFileInfo

Classe SPLFileInfo

- Instanciée avec le nom d'un fichier.
- Ses méthodes donnent accès aux méta-données système sur le fichier : date d'accès, extension, nom, groupe, noeud d'index (inode), fichier cible (lien), propriétaire et permissions, chemin, taille, type ...

splfileinfo.php

```
1 $file=new SPLFileInfo(__DIR__.'/html.txt');
2 echo $file->getExtension().".\n";
3 echo $file->getSize().".\n";
4 echo $file->isReadable()  .".\n";
```

La classe SPLFileObject

Classe SPLFileObject

- Hérite de SPLFileInfo et implémente RécursiveIterator et SeekableIterator.
- Instanciée avec le nom d'un fichier.
- Offre un large éventail de méthodes pour manipuler des fichiers.

Envoi d'une image en réponse HTTP (splfileobject.php)

```php
// ouvre le fichier en mode lecture binaire
$file = new SplFileObject(__DIR__."/imagesvg.png", "rb");
// envoie les bonnes en-têtes
header("Content-Type: image/png");
header("Content-Length: " . $file->getSize());
// affiche sur le tampon de sortie et termine le script
$file->fpassthru();
exit;
```

Parcours de répertoires (SPL)

spl-directory.php

```php
try {
    foreach (new DirectoryIterator('./') as $item) {
        echo $item."\n";
    }
} catch(Exception $e) {
    echo "No files found!\n";
}
```