

Téléversement de fichiers

Principe de fonctionnement

L'idée est d'utiliser un formulaire html avec une entrée input de type file avec la syntaxe :

```
<input type="file" name="nomFichier">
```

A la différence de toutes les autres entrées d'un formulaire classique, l'entrée de type file ne peut être associée qu'à la méthode POST. De plus, l'envoi des données de type file du formulaire conduit conduit à la construction de la variable globale `$_FILES` dont la clé est 'nomFichier'. Ce tableau associatif comprend plusieurs entrées construites de façon automatique :

- `$_FILES['nomFichier']['name']` qui contient le nom du fichier téléchargé par le client avec son extensions
- `$_FILES['nomFichier']['size']` qui contient la taille en octets du fichier
- `$_FILES['nomFichier']['type']` qui contient le type de contenu du fichier (par exemple image/jpeg). Il s'agit de l'un des différents types MIME (*Multipurpose Internet Mail Extensions*) qui décrivent le contenu d'un fichier (voir par exemple : [https://fr.wikipedia.org/wiki>Type de médiatique](https://fr.wikipedia.org/wiki>Type_de_m%C3%A9diatique))
- `$_FILES['nomFichier']['tmp_name']` qui contient le nom du fichier tel qu'il est stocké temporairement sur le serveur, le temps de le placer dans le dossier qui convient.
- `$_FILES['nomFichier']['error']` qui contient le code erreur résultant de l'envoi du fichier.

```
<p>
    <label for="nomFichier">Indiquez le chemin du fichier à téléverser (une
        image de type jpeg, png ou gif d'une taille maximale de 2Mo) </label>
    <input type="file" name="nomFichier" id="nomFichier" form="form_upload">
</p>
<p align="center">
    <input type="submit" value="Upload" form="form_upload">
</p>
```

```
Array
(
    [name] => P1010837.JPG
    [type] => image/jpeg
    [tmp_name] => C:\xampp\tmp\php2360.tmp
    [error] => 0
    [size] => 457349
)
```

Mise en pratique

Comme on ne maîtrise absolument pas ce que l'utilisateur va tenter de téléverser sur le serveur il faut prendre de grandes précautions quant aux fichiers transmis par formulaire. L'idée est donc d'informer l'utilisateur du type de fichier attendu puis de filtrer ces fichiers en fonction de ce qui aura été effectivement téléchargé.

Pour cela, on va ajouter un attribut à la balise form pour que le navigateur s'occupe de transmettre au serveur le type de fichier téléchargé :

```
<form action="upload.php" method="POST" enctype="multipart/form-data" id="form_upload">
```

Ensuite, dans le fichier PHP cible, on va placer le fichier téléchargé dans le bon dossier du site web que s'il

correspond à ce qui est attendu. C'est pour cette raison que le téléversement n'est pas automatique et passe par le dossier /tmp/ du serveur web.

Voici le déroulement séquentiel de la procédure de « sécurisation » du téléversement :

1. On récupère les informations transmises par le formulaire :

```
$nom=$_FILES['nomFichier']['name'];
$size=$_FILES['nomFichier']['size'];
$type=$_FILES['nomFichier']['type'];
$nom_tmp=$_FILES['nomFichier']['tmp_name'];
```

2. On filtre le fichier téléchargé en fonction de son type en créant la chaîne de caractère correspondant au chemin du futur fichier sur le serveur :

```
switch ($type) {
    case 'image/jpeg':
        $nom_upload="image".time().".jpg";
        break;
    case 'image/png':
        $nom_upload="image".time().".png";
        break;
    case 'image/gif':
        $nom_upload="image".time().".gif";
        break;
    default:
        $nom_upload="";
        break;
}
```

Notez que :

- Il s'agit d'une situation où on ne veut récupérer que des fichiers d'image des trois types jpg, png et gif
- Le cas par défaut permet de créer un nom de fichier vide dans le cas où le fichier n'est pas du format voulu.
- On a concaténé un timestamp au nom du fichier afin d'éviter les doublons de nom de fichier
- Le renommage est essentiel afin de pouvoir maîtriser le nom des fichiers stockés sur le serveur.

3. On déplace le fichier dans le dossier adéquat :

```
if ($nom_upload!="") {
    if (move_uploaded_file($nom_tmp,"images/".$nom_upload)) {
        echo "<h1>Le fichier a bien été transféré sur le serveur !</h1>";
    }
    else {
        echo "<h1>Une erreur est survenue lors du téléchargement !</h1>";
    }
}
else {
    echo "<h1>Le type de fichier n'est pas valide !</h1>";
}
```

Notez que :

- la fonction `move_uploaded_file` permet, comme son nom l'indique, de déplacer le fichier de la zone temporaire vers le dossier `images/` présent sur le serveur.
- Cette fonction renvoie un booléen qui « valide » le transfert.
- Aucun fichier n'est transféré s'il n'est pas du type attendu.

Pour finir, la taille maximale de téléversement de fichier est fixé par une constante contenue dans le fichier de configuration `php.ini`, il s'agit de la constante `upload_max_filesize` visible sur la page `phpinfo` du serveur (dans xampp, cette constante vaut 40Mo par défaut !!). Enfin, les dimensions effectives des images ne sont pas gérées par ce transfert. Pour cela, il faut utiliser d'autres outils permettant de maîtriser ces dimensions : c'est la librairie GD par exemple mais là, c'est une autre histoire ...

Créez un formulaire de dépôt d'image dans un dossier du serveur ne récupérant que des fichiers images d'un type et d'un « poids » maximal que vous définirez dans votre formulaire.

Pour les plus aventureux, vous pouvez chercher à permettre un téléversement de fichiers multiples en ajoutant l'attribut `multiple` à l'input de type file. Il vous faudra alors déclarer le `name` en tant que tableau et traiter le téléversement dans une boucle dont le compteur correspondra au nombre d'éléments du tableau `$_FILES`.

Lister les éléments d'un dossier

Il peut être tout à fait pertinent de pouvoir afficher toutes les images d'un dossier spécifique pour former une galerie d'images. Pour cela, il existe deux fonctions PHP qui permettent d'ouvrir et de lister le contenu d'un dossier :

```
if($dossier = opendir('./images'))
{
    while(($fichier = readdir($dossier)) != false)
    {
        echo "<p>$fichier</p>";
    }
}
```

Le premier test permet de s'assurer des droits nécessaires à l'accès au `$dossier` et la boucle `while` tourne sur un test le lecture successives de chaque élément du `$dossier` stocké dans la variable `$fichier`.

Une autre solution consiste à utiliser la fonction PHP, `scandir()` :

```
$fichiers=scandir('images/');
foreach ($fichiers as $fichier) {
    echo "<p>$fichier</p>";
}
```

Si vous testez ce code vous verrez apparaître deux éléments particuliers dans la liste. A quoi correspondent-ils ? Comment les « filtrer » du contenu ?

Faire afficher les images des images avec un framework de galerie

Il existe de très nombreux framework de galerie open source sur le net. Je vous en propose un qui est basique mais paramétrable de façon simple.

Il repose sur un fichier CSS pour la mise en forme et sur un fichier JS utilisant Jquery pour l'interaction avec l'utilisateur, les deux sont à récupérer sur UniversiTICE. La galerie se trouvera dans une div de classe gallery et chaque image dans une div de classe img-w, ce qui donne quelque chose comme ça :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="style.css">
    <script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js'></script>
    <script type="text/javascript" src='gallery.js'></script>
  </head>

  <body>

    <div class="gallery">
      <div class="img-w">
        
      </div>
      <div class="img-w">
        
      </div>
    </div>

  </body>
```

Tâche finale : réaliser une galerie dynamique

Fusionner les différents travaux réalisés pour obtenir une galerie dynamique dans laquelle on peut téléverser des fichiers images filtrés selon des critères qui vous sont propres dans un dossier dont le contenu s'affiche au fur et à mesure du téléchargement.