

## TELEVERSEMENT DE FICHIER

### 1. PRINCIPE

Le téléversement de fichier, encore appelé `upload`, permet de transmettre un fichier depuis le client vers le serveur. Cette opération requière un certain nombre de précautions afin d'éviter tout acte malveillant.

#### 1.1. Sélection du fichier à télécharger

La sélection du fichier à transmettre à partir du client se fait au moyen d'une balise `<input type="file">`, insérée dans un formulaire :

Exemple (fichier `upload.html`) :



```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Upload</title>
  <style>
    form {width: 300px;}
  </style>
</head>
<body>
  <form action="upload.php" method="post" enctype="multipart/form-data">
    <fieldset>
      <legend><strong>Déposez votre photo</strong></legend>
      <input type="hidden" name="MAX_FILE_SIZE" value="500000">
      <input type="file" name="photo" accept="image/gif, image/jpeg, image/png">
    </fieldset>
    <input type="submit" name="envoyer" value="Envoyer">
  </form>
</body>
</html>
```

Remarques :

- Un transfert de fichier requière obligatoirement l'emploi de la méthode `post` ainsi que l'encodage `enctype="multipart/form-data"` dans la balise `<form>`.
- Il est possible de limiter la taille maximale du fichier à transférer en insérant une balise `<input type="hidden">` (cachée) nommée `MAX_FILE_SIZE` avant la balise `<input type="file">`.

Exemple :

```
<input type="hidden" name="MAX_FILE_SIZE" value="500000">
```

Dans cet exemple, la taille est limitée à 500000 octets.

A noter : Cette limitation peut facilement être contournée en modifiant `value` dans le navigateur. Il est par conséquent obligatoire de procéder à une vérification également du côté serveur.

- La sélection du fichier à télécharger se fait avec la balise `<input type="file" name="photo">`. Cette balise est matérialisée dans le navigateur par le bouton "Parcourir...". L'attribut `name="photo"` désigne le nom de la variable dans laquelle les informations du fichier téléchargé seront accessibles dans le script PHP, coté serveur.

L'attribut `accept` permet de filtrer les fichiers éligibles au téléchargement dans l'arborescence du client. Dans l'exemple ci-dessus, sont acceptés les fichiers : `.gif`, `.jpg` et `.png`.

- Dans cet exemple, la validation du formulaire appelle l'exécution du script `upload.php`.

## 1.2. Traitement du fichier téléchargé sur le serveur

La validation du formulaire transmet le fichier sélectionné vers le serveur.

Le fichier reçu est stocké provisoirement dans un emplacement temporaire sur le serveur, avant d'être déplacé vers le dossier définitif. (Nous y reviendrons plus loin.)

En plus du fichier, les informations suivantes sont également transmises vers le serveur :

- Le tableau `$_POST` (cf. la méthode choisie) avec les éléments :
  - `'MAX_FILE_SIZE' => string '500000' (length=6)`
  - `'envoyer' => string 'Envoyer' (length=7)`
- Le tableau à 2 dimensions `$_FILES`, contenant un élément nommé `photo` (cf. attribut `name="photo"`) avec la description du fichier téléchargé :
  - `'photo' =>`  

```
array (size=5)
  'name' => string 'fleur.jpg' (length=9)
  'type' => string 'image/jpeg' (length=10)
  'tmp_name' => string 'C:\ ... \Temp\phpA8A0.tmp' (length=48)
  'error' => int 0
  'size' => int 61331
```

Description du contenu du tableau à 2 dimensions `$_FILES` :

Contenu de <code>\$_FILES</code>	Description
<code>\$_FILES["photo"]["name"]</code>	Nom du fichier sur le poste client (sans le nom du dossier)
<code>\$_FILES["photo"]["type"]</code>	Type MIME* initial du fichier (fourni par le navigateur)
<code>\$_FILES["photo"]["size"]</code>	Taille en octets du fichier transféré
<code>\$_FILES["photo"]["tmp_name"]</code>	Nom aléatoire du fichier temporaire créé sur le serveur (chemin complet)
<code>\$_FILES["photo"]["error"]</code>	Code d'erreur éventuel lors du transfert du fichier : 0 : transfert correctement réalisé (→ pas d'erreur), 1 : taille du fichier supérieure à celle définie dans <code>php.ini</code> , 2 : taille du fichier supérieure à celle définie dans <code>MAX_FILE_SIZE</code> , 3 : le fichier a été chargé partiellement, 4 : aucun fichier n'a été chargé.

\* : Le type MIME (Multipurpose Internet Mail Extensions) est un standard, initialement utilisé pour les courriers électroniques, permettant d'indiquer la nature et le format d'un document :

- Un type MIME est composé de 2 parties : `type/sous-type`.
- Il est possible de sélectionner tous les sous-types avec `*`.
- Exemple : `audio/*` désigne tous les fichiers audios.
- L'attribut `accept` de la balise `<input type="file">` utilise les types MIME.

Si la transmission du fichier s'est bien terminée, il est provisoirement enregistré à l'emplacement :

`$_FILES["photo"]["tmp_name"]`

Le traitement du fichier reçu nécessite plusieurs vérifications :

- Est-ce qu'une erreur s'est produite lors du transfert (cf. `$_FILES["photo"]["error"]`) ?
- Est-ce que la taille du fichier (cf. `$_FILES["photo"]["size"]`) ne dépasse pas la limite fixée ?
- Est-ce que le type (ou l'extension) du fichier est autorisé ?

Pour achever le traitement, il faudra déplacer le fichier vers son emplacement définitif avec la fonction :

`move_uploaded_file()`.

Exemple (fichier upload.php) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Upload</title>
</head>
<body>
<?php
// Test si le fichier a été transféré
if (isset($_FILES['photo']))
{
    echo 'Contenu de <b>$_FILES</b>:<br>';
    var_dump($_FILES);

    // Test s'il n'y a pas d'erreur
    if ($_FILES['photo']['error'] == 0)
    {
        // Test si la taille du fichier uploadé est conforme
        if ($_FILES['photo']['size'] <= 500000)
        {
            // Test si l'extension du fichier uploadé est autorisée
            $infosfichier = new SplFileInfo($_FILES['photo']['name']);
            $extension_upload = $infosfichier->getExtension();
            $extensions_autorisees = array('jpg', 'gif', 'png');
            if (in_array($extension_upload, $extensions_autorisees))
            {
                // Stockage définitif du fichier photo dans le dossier uploads
                move_uploaded_file($_FILES['photo']['tmp_name'],
                                   'uploads/' . $_FILES['photo']['name']);
                echo "Transfert du fichier <b>".$_FILES['photo']['name']. "</b> effectué !";
            }
            else
                echo "Echec du transfert : Type de fichier non autorisé.";
        }
        else
            echo "Echec du transfert : Fichier trop volumineux.";
    }
    else
        echo "Echec du transfert avec le code d'erreur : ".$_FILES['photo']['error'];
}
?>
</body>
</html>
```

Remarques :

- Le fichier reçu est déplacé dans le dossier uploads à l'aide de la fonction `move_uploaded_file()`.
- Il faut que le dossier uploads ait été créé préalablement au même endroit que le script.
- La classe `SplFileInfo` fournit une interface de haut niveau avec les informations de fichiers.

## 2. CONFIGURATION DU SERVEUR

Les paramètres conditionnant le téléchargement de fichiers sont fixés sur le serveur, dans le fichier de configuration du module PHP : `php.ini`.

(Pour Laragon, ce fichier se situe dans le dossier : `laragon\bin\php\php-8.2.8-Win32-vs16-x64`.)

Quelques paramètres :

- Autorisation du téléchargement de fichier : `file_uploads = On`
- Dossier utilisé pour le stockage temporaire des fichiers téléchargés : `upload_tmp_dir = ...`
- Taille maximale d'un fichier téléchargé : `upload_max_filesize = 20M`
- Nombre maximal de fichiers pouvant être téléchargés simultanément : `max_file_uploads = 20`
- ...

## 3. GESTION DES FICHIERS ET DES DOSSIERS AVEC PHP

Le langage PHP propose une liste de fonctions permettant de gérer les fichiers et les dossiers dans l'arborescence du serveur.

Fonctions pour manipuler des fichiers :

- `is_file()` : Vérifie si un fichier existe.
- `file()` : Lit un fichier et renvoie son contenu, ligne par ligne, dans un tableau.
- `fopen()` : Ouvre un fichier
- `fgets()` : Récupère la ligne courante d'un fichier.
- `fwrite()` : Ecrit du contenu dans un fichier.
- `fclose()` : Ferme un fichier.
- `unlink()` : Supprime un fichier.

Fonctions pour manipuler des dossiers :

- `is_dir()` : Vérifie si un dossier existe.
- `mkdir()` : Crée un dossier.
- `rmdir()` : Supprime un dossier.
- `scandir()` : Liste le contenu d'un dossier.

## 4. EXERCICE

Rendez le formulaire ci-dessus auto-invoquant.

Enregistrez les images dans le dossier `mesImages` (si ce dossier n'existe pas, crée-le avec le script PHP).

Affichez l'image téléchargée à la suite du message "Transfert du fichier ... effectué !"

**Déposez votre photo**

Aucun fichier sélectionné.

Transfert du fichier photo5.jpg effectué !

