

PHP- Le téléchargement de fichiers

Les formulaires HTML offrent la possibilité de télécharger un fichier : par exemple une photo, la notice d'un produit, un C.V. sur un site d'emploi (donc au format Word ou PDF).

Tout d'abord, un point sur le mot *télécharger* qui peut désigner aussi bien les opérations suivantes :

- enregistrement sur un PC d'un fichier présent sur un serveur distant (site web); il s'agit de **download**.
- envoi vers un serveur distant d'un fichier qui se trouve sur un PC : il s'agit de **l'upload**.

C'est ce second cas qui nous intéresse ici.

Formulaire HTML

Pour que le téléchargement soit possible, il faut ajouter l'attribut `enctype` à la balise `<form>`. La valeur doit être `multipart/form-data` :

```
<form action="post.php" method="post" enctype="multipart/form-data">
```

Ensuite, on a besoin d'un champ de type `file`, qui fera apparaître un bouton contenant le texte *Parcourir* avec lequel on pourra choisir un fichier présent sur le PC :

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

Traitement en PHP

Dans le fichier de traitement PHP (celui assigné comme valeur à l'attribut `action`, lorsque le formulaire est soumis, on récupère les informations sur le fichier via la variable superglobale `$_FILES`. Comme les 7 autres superglobales, cette variable se comporte comme un tableau PHP.

Exercice : créer un formulaire d'upload et le fichier PHP de traitement correspondant, dans le fichier PHP écrivez juste `var_dump($_FILES)`; et observez le résultat.

Vous devriez avoir quelque chose du genre :

- 'name' => string 'monfichier.jpg' (length=10) = **nom du fichier d'origine**
- 'type' => string 'img/jpeg' (length=10) = **type MIME du fichier**
- 'tmp_name' => string 'C:\tmp\phpC1CD.tmp' (length=23) = **nom et chemin du fichier temporaire**
- 'error' => int 0 = **erreurs (s'il y en a, elles sont retournées via un tableau PHP)**
- 'size' => int 100813 = **taille du fichier, en octets**

Gestion des erreurs

Si le téléchargement échoue, les erreurs sont retournées dans `$_FILES["fichier"]["error"]`, les cas d'erreur sont prédéfinis dans un tableau.

- S'il n'y a pas d'erreur, `$_FILES["fichier"]["error"]` vaut 0,
- S'il y en a, la taille du tableau est supérieure à 0

Pour savoir si le tableau contient des erreurs, il faut donc faire :

```
if (sizeof($_FILES["fichier"]["error"]) > 0) { ... }
```

--> [Voir ici la documentation complète de PHP sur la gestion des erreurs d'upload](#)

Sécurité

Le problème principal de l'upload d'un fichier est la sécurité : c'est l'utilisateur qui envoie un fichier présent sur son PC, et comme il ne faut jamais faire confiance aux actions de l'utilisateur, il faut vérifier que le fichier reçu est bien du type attendu et ne comporte pas de code malicieux (tentative d'attaque via un fichier exécutable, virus, logiciel espion etc.).

Il faut ensuite s'assurer des droits sur ce fichier (écriture, lecture, exécution) et le stocker correctement sur le serveur (s'agit-il d'un fichier accessible publiquement à tous les internautes ou d'un [contenu confidentiel](#) ?).

Vérifier le type

On doit tout d'abord s'assurer de points basiques :

- un fichier a-t-il bien été téléchargé ?
- le type du fichier envoyé par l'utilisateur est-il celui attendu (image, document Word, PDF...) ?

Les fausses bonnes idées, car les informations retournées ne sont pas fiables :

- tester uniquement l'extension comme chaîne de caractère
- tester **le type MIME** retourné par le navigateur (celui dans `$_FILES["fichier"]["type"]`).

PHP fournit une extension nommée `_FILEINFO` qui fait référence en termes de sécurité. Voici comment l'utiliser, pour un type :

```
// On met les types autorisés dans un tableau (ici pour une image)
$aMimeTypes = array("img/gif", "img/jpeg", "img/pjpeg", "img/png", "img/x-png",
"img/tiff");

// On extrait le type du fichier via l'extension FILEINFO
$finfo = finfo_open(FILEINFO_MIME_TYPE);
$mimetype = finfo_file($finfo, $_FILES["fichier"]["tmp_name"]);
finfo_close($finfo);

if (in_array($mimetype, $aMimeTypes))
{
    /* Le type est parmi ceux autorisés, donc OK, on va pouvoir
    déplacer et renommer le fichier */
}
else
{
    // Le type n'est pas autorisé, donc ERREUR

    echo "Type de fichier non autorisé";
    exit;
}
```

Déplacer et renommer le fichier

Par défaut, le fichier téléchargé est stocké dans le répertoire `tmp` de votre serveur. Mais ce fichier devra sans doute se trouver dans un répertoire de votre projet, il faut donc le déplacer.

Il est nécessaire aussi de renommer le fichier, pour répondre à une éventuelle charte de nommage et surtout pour que l'utilisateur ne puisse tenter d'exécuter le fichier via l'url (le nom sur le serveur sera différent de celui qu'il connaissait).

Pour cela, PHP propose une fonction "2 en 1" : `move_uploaded_file()`.

Exemple : déplacer et renommer un fichier (de type image) de `tmp` vers un répertoire nommé `images` d'un projet :

```
move_uploaded_file($_FILES["fichier"]["tmp_name"], "img/photo.jpg");
```

La logique veut que les contrôles de sécurité ait été réalisés avant le déplacement.

Spécifier des droits sur le fichier

Sur les systèmes d'exploitation (Windows, Linux...), les fichiers possèdent des droits (ou permissions) de lecture, d'exécution et d'écriture accordés aux utilisateurs. Il s'agit d'un système un peu complexe mais qui participe grandement à la sécurité.

Lire ces ressources :

- [Modifier les permissions sur un fichier](#)
- [La fonction `chmod\(\)` de PHP](#)