

Serveur Web - page upload



Pour créer une page PHP qui reçoit un fichier envoyé depuis un script Bash, nous pouvons utiliser la méthode HTTP POST. Voici un exemple de code qui vous aidera à démarrer :

```
<?php
// Afficher les erreurs PHP
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$allowed_types = array('text/csv');

// Spécifiez le répertoire de destination
$target_dir = "imports";

// Créer le répertoire de destination s'il n'existe pas
if (!file_exists($target_dir)) {
    if (mkdir($target_dir, 0777, true)) {
        echo "Le répertoire de destination a été créé avec succès.\n";
    } else {
        echo "Impossible de créer le répertoire de destination\n";
        exit();
    }
}

if (!isset($_FILES['file'])) {
    echo "Aucun fichier présenté.\n";
    exit();
}

if (in_array($_FILES['file']['type'], $allowed_types)) {
    echo "Le type MIME du fichier n'est pas accepté.\n";
    exit();
}

if ($_FILES['file']['error'] !== UPLOAD_ERR_OK) {
    switch ($_FILES['file']['error']) {
        case UPLOAD_ERR_INI_SIZE:
            echo "La taille du fichier dépasse la limite autorisée.\n";
            break;
        case UPLOAD_ERR_FORM_SIZE:
            echo "La taille du fichier dépasse la limite autorisée par le\n";
            formulaire.\n";
            break;
    }
}
```

```
        break;
    case UPLOAD_ERR_PARTIAL:
        echo 'Le fichier n\'a été que partiellement téléchargé.\n';
        break;
    case UPLOAD_ERR_NO_FILE:
        echo 'Aucun fichier n\'a été téléchargé.\n';
        break;
    case UPLOAD_ERR_NO_TMP_DIR:
        echo 'Le dossier temporaire est manquant.\n';
        break;
    case UPLOAD_ERR_CANT_WRITE:
        echo 'Impossible d\'écrire le fichier sur le disque.\n';
        break;
    case UPLOAD_ERR_EXTENSION:
        echo 'Une extension PHP a empêché le téléchargement du
fichier.\n';
        break;
    default:
        echo 'Une erreur s\'est produite lors du téléchargement du
fichier.\n';
    }
    echo "\n";
    exit();
}

// Vérifie si le fichier a été correctement envoyé
if (isset($_FILES['file']) && $_FILES['file']['error'] == 0) {
    // Spécifiez le chemin de destination pour enregistrer le fichier
    $destination = $target_dir . '/' . $_FILES['file']['name'];

    // Si le fichier existe déjà, le supprimer
    if (file_exists($destination)) {
        unlink($destination);
    }

    // Déplacez le fichier du répertoire temporaire vers le répertoire de
    destination
    if (move_uploaded_file($_FILES['file']['tmp_name'], $destination)) {
        if (file_exists($destination)) {
            $crc = hash_file('sha1', $destination);
            echo $crc;
        } else {
            echo "Une erreur s'est produite lors du téléchargement du
fichier.\n";
            exit;
        }
    } else {
        echo "Une erreur s'est produite lors du téléchargement du
fichier.\n";
    }
}
```

```
        exit;
    }
} else {
    echo "Une erreur s'est produite lors de l'envoi du fichier.\n";
}
?>
```

Le code ci-dessus vérifie si un fichier a été envoyé via la méthode POST et utilise la fonction PHP **move_uploaded_file** pour déplacer le fichier du répertoire temporaire vers un répertoire de destination sur votre serveur.

-

La création du répertoire de destination `imports` avec la fonction **mkdir** si celui-ci n'existe pas crée récursivement le répertoire de destination, en accordant les autorisations de lecture, d'écriture et d'exécution à tous les utilisateurs. Sauf qu'il est peut-être nécessaire de créer ce dossier en ligne de commande dans le dossier public du site : `mkdir imports` Le dossier `imports` a été créé avec un complément de droit dans le Terminal `chmod g+w imports`.

-

Si le fichier de destination existe déjà avec la fonction **file_exists**, nous utilisons la fonction **unlink** pour le supprimer avant de déplacer le nouveau fichier avec la fonction **move_uploaded_file**. Ainsi, le fichier existant sera remplacé par la nouvelle version.

-

La fonction **is_dir** permet de vérifier si le répertoire de destination n'existe pas, alors nous utilisons la fonction **mkdir** pour le créer, puis nous vérifions si le répertoire a été créé avec succès avant de continuer le traitement du fichier téléchargé.

Si le répertoire est créé avec succès, nous continuons avec la vérification et le traitement du fichier téléchargé, sinon nous affichons un message d'erreur et sortons du script avec la fonction **exit**.

-

Nous avons utilisé une structure conditionnelle pour vérifier la valeur de `$_FILES['file']['error']`. Si la valeur n'est pas **UPLOAD_ERR_OK**, nous affichons un message d'erreur correspondant à la valeur d'erreur spécifique. Les constantes **UPLOAD_ERR_INI_SIZE**, **UPLOAD_ERR_FORM_SIZE**, **UPLOAD_ERR_PARTIAL**, **UPLOAD_ERR_NO_FILE**, **UPLOAD_ERR_NO_TMP_DIR**, **UPLOAD_ERR_CANT_WRITE**, **UPLOAD_ERR_EXTENSION** sont des constantes prédéfinies en PHP qui indiquent les différentes erreurs de téléchargement de fichier possibles.

En utilisant ce code, nous pouvons afficher un message d'erreur plus précis pour aider à résoudre le problème de téléchargement de fichier.

-

Un contrôle du CRC sur un fichier téléchargé et renvoyé. Nous utilisons la fonction **hash_file** pour calculer la valeur SHA1 du fichier. La fonction **hash_file** calcule le hash d'un fichier donné à l'aide de l'algorithme de hachage spécifié. Dans notre cas, nous allons utiliser l'algorithme de hachage SHA1 pour calculer la valeur CRC du fichier.

Nous devons également vérifier le paramétrage de PHP.

- La directive **post_max_size** est définie dans un fichier de configuration différent. Nous pouvons utiliser la fonction `phpinfo` pour vérifier le fichier de configuration utilisé par PHP. Cherchez la ligne "Loaded Configuration File" pour trouver le fichier utilisé. La directive **post_max_size** est définie dans un niveau supérieur de la hiérarchie de configuration et ne peut pas être modifiée par la fonction **ini_set**. Par exemple, si la directive est définie dans le fichier **php.ini** global, elle ne peut pas être modifiée par **ini_set** dans un script PHP.

La modification de la directive **post_max_size** nécessite un redémarrage du serveur web pour prendre effet. Assurez-vous que le serveur web a été redémarré après avoir modifié la valeur de **post_max_size** dans le fichier **php.ini**.

- Nous avons également modifié la directive `upload_max_filesize`.

— [Cédric ABONNEL dit Cédrix](#)

From:
<https://www.abonnel.fr/> - notes informatique & technologie

Permanent link:
https://www.abonnel.fr/electronique/teleinformation-compteur-electricite/page_upload

Last update: **2023/02/16 03:42**

