

Incarcarea unui fisier pe server se face prin intermediul formularelor. Pentru aceasta facilitate se foloseste un formular putin diferit: are un atribut special (enc-type) si metoda POST. De asemenea, trebuie sa contina un element input de tip file pentru fiecare fisier incarcat (spre exemplu, daca se incarca 2 fisiere, va fi nevoie de 2 elemente input).

Pentru ca functionalitatea de upload sa fie functioneze, trebuie indeplinite urmatoarele conditii:

- directiva file\_uploads din fisierul de configurari al PHP (php.ini) trebuie sa fie 'on'

- directiva upload\_tmp\_dir din php.ini trebuie sa se refere la o cale existenta pe server si cu permisiuni suficiente pentru ca web-serverul sa poata crea fisiere

- directivele upload\_max\_filesize si post\_max\_size din php.ini specifica marimea maxima a fisierului si respectiv a datelor ce pot fi transmise prin intermediul formularului; este recomandat ca aceste valori sa fie revizuite

atributul enctype="multipart/form-data" NU TREBUIE omis, altfel uploadul nu va functiona

Nota: de obicei, interpretorul PHP este configurat deja pentru upload de fisiere, deci primele 3 conditii de mai sus sunt aproape intotdeauna indeplinite fara a fi nevoie de alte configurari. Acesta este si cazul pachetului EasyPHP la care incarcarea de fisiere este activata si configurata implicit.

Un exemplu de formular pentru upload de fisiere este mai jos:

```
<!-- Encoding type, exprimat de atributul enctype, TREBUIE specificat ca mai jos -->
```

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

```
  Incarca un fisier: <input name="fisier" type="file" />
```

```
  <input type="submit" value="Trimite fisier" />
```

```
</form>
```

Nota asupra mecanismului de upload:

atunci cand se face submit la formular, browser-ul web transmite catre server continutul fisierului

serverul web copiaza fisierul transmis intr-o locatie temporara (specificata de directiva upload\_tmp\_dir)

interpretorul PHP este invocat, avand variabila \$\_FILES populata cu informatii despre upload

programatorul este responsabil cu prelucrarea fisierului incarcata pe server (mutare intr-o alta locatie, citire, copiere, etc); prelucrarea ce se face cu ajutorul functiilor puse la dispozitie de PHP. Daca fisierul incarcata nu este mutat (sau redenumit) din locatia temporara, acesta va fi sters automat la terminarea executiei scriptului (la finalul requestului, mai exact).

Fisierul PHP (upload.php) ce va prelucra uploadul contine urmatoarea secventa de cod (include si validare).

```
<?php
```

```
# se verifica daca fisierul incarcata a depasit dimensiunea maxima acceptata
```

```
# daca acest lucru se intampla, variabilele $_POST si $_FILES se golesc automat
```

```
if( empty( $_POST ) && empty( $_FILES ) ) {
```

```
    # a fost facut un POST sau nu?
```

```
    if( isset( $_SERVER['CONTENT_LENGTH'] ) ) {
```

```
        # atunci cand se trimite date prin post variabila $_SERVER['CONTENT_LENGTH']
```

```
        # contine valoarea dimensiunii datelor
```

```
        $POST_MAX_SIZE = ini_get('post_max_size');
```

```
        if( !empty( $POST_MAX_SIZE ) ) {
```

```
            # determin limita maxima (care poate fi in KB, MB sau GB
```

```
            $mul = substr($POST_MAX_SIZE, -1);
```

```
            $mul = ($mul == 'M' ? 1048576 :(
```

```
                $mul == 'K' ? 1024 :($mul == 'G' ? 1073741824 : 1));
```

```
            if ( $_SERVER['CONTENT_LENGTH'] > $mul*(int)$POST_MAX_SIZE &&
```

```
                $POST_MAX_SIZE ) {
```

```
                print "Fisier prea mare! Ati depasit limita maxima permisa";
```

```
            }
```

```
        } else {
```

```
            print "Eroare nespecificata (probabil fisierul este prea mare)";
```

```
        }
```

```

    } else {

        # nu s-a facut inca submit la fisier, afisez un mesaj

        print "Apasati pe 'Trimite fisier' pentru a face upload!";

    }
} else {

    # $_POST si $_FILES sunt setate; verific alte erori ce pot sa apara

    if( $_FILES['fisier']['error'] > 0 ) {

        print "A intervenit o eroare ({$_FILES['fisier']['error']})";

    } else {

        # fisierul uploadat va fi pus in subfolderul 'upload' (care trebuie sa
        # existe deja in aceeasi locatie ca si fisierul upload.php

        $uploaddir = dirname( __FILE__ ). DIRECTORY_SEPARATOR .
            'upload' . DIRECTORY_SEPARATOR;

        $uploadfile = $uploaddir . basename($_FILES['fisier']['name']);

        if (move_uploaded_file($_FILES['fisier']['tmp_name'], $uploadfile)) {

            print "Fisier incarcat cu succes!";

        } else {

            print "Nu s-a putut incarca fisierul";

        }

    }

}

?>

```

Incarcarea mai multor fisiere pe server in acelasi timp (upload de fisiere multiple) 2 !

Cand este nevoie sa se incarce mai multe fisiere in acelasi timp, formularul va contine mai multe elemente de tip INPUT FILE, denumite sub forma unui vector (array):

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

```
<p>Fisiere:
```

```
<input type="file" name="pictures[]" />
```

```
<input type="file" name="pictures[]" />
```

```
<input type="file" name="pictures[]" />
```

```
<input type="submit" value="Incarca" />
```

```
</p>
```

```
</form>
```

Codul PHP pentru prelucrarea uploadului trebuie scris in acelasi fisier ca si formularul (intrucat atributul action nu specifica in mod explicit un fisier PHP).

Nota: acesta este o varianta cu validari minimale.

```
<?php
```

```
# pentru fiecare fisier incarcat, fac o verificare dupa status code
```

```
if( !empty( $_FILES ) )
```

```
foreach ( $_FILES["pictures"]["error"] as $key => $error ) {
```

```
    if ( $error > 0 ) { # echivalent cu ( $_FILES["pictures"]["error"][$key] > 0 )
```

```
        print "Eroare cu fisierul {$_FILES["pictures"]["tmp_name"][$key]}!";
```

```
    } else {
```

```
        $tmp_name = $_FILES["pictures"]["tmp_name"][$key];
```

```
        $name = $_FILES["pictures"]["name"][$key];
```

```
        # mut fisierul din locatia temporara in directorul curent (acelasi
```

```
        # director in care se afla scriptul PHP)
```

```
        move_uploaded_file($tmp_name, "$name");
```

```
    }
```

}

?>