

11. PHP

1

Uploadovanie súborov

Obsah

2

- Uploadovanie súborov

Uploadovanie súborov

3

- Pri uploadovaní súborov pomocou formulára musíme nastaviť **metódu POST**.
- Musíme nastaviť typ kódovania údajov vo formulári na **multipart/form-data**.
- Vložíme prvok formulára typu **file**
`<input type="file" name="xyz">`
- Pri prvku typu **file** sa nám automaticky zobrazí tlačidlo na nájdenie súboru v našom počítači.
- Prvkov na uploadovanie súborov môžeme mať vo formulári viac.

Formulár na uploadovanie

4

- Do objednávky (objednavka.php) pridáme pole s potvrdením (súborom).
- Do tabuľky tour_objednavky pridáme položku **potvrdenie** na názov súboru (VARCHAR 200)

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

...

```
<input type="file" name="potvrdenie">  
</form>
```

Maximálna veľkosť uploadovaného súboru

5

- Vo formulári môžeme vytvoriť špeciálny skrytý prvok s názvom **MAX_FILE_SIZE**, v ktorom nastavíme maximálnu prípustnú veľkosť uploadovaného súboru (v bytoch).
- Nemusí fungovať vo všetkých prehliadačoch.
- **Netreba sa na neho spoliehať!**
- `<input type="hidden" name="MAX_FILE_SIZE" value="30000">`

Akceptované typy súborov

6

- Pri prvku typu **file** môžeme definovať atribút **accept**, ktorým definujeme akceptované typy (tzv. MIME typ) súborov (www.iana.org/assignments/media-types).
- Viac typov oddeľujeme čiarkou.
- Atribút accept nefunguje vo všetkých prehliadačoch (v IE 9 a menej nefunguje)! **Nespoliehať sa na tento atribút.**
- `<input type="file" name="potvrdenie" accept="typ_súboru">`
- Napr. `<input type="file" name="potvrdenie" accept="image/gif, image/jpeg">`

Čo sa deje po odoslaní formulára so súborom?

7

- Všetky klasické prvky formulára sa správajú ako doteraz, teda sú v poli `$_POST`.
- Uploadované súbory sa správajú inak.
- Sú dostupné cez pole `$_FILES['xyz']`, kde `xyz` je názov prvku typu **file** (`<input type="file" name="xyz">`).

Údaje po odoslaní formulára

8

PHP Variables

Variable	
<code>_POST["dospeli"]</code>	3
<code>_POST["deti"]</code>	0
<code>_POST["id_zajazdu"]</code>	2
<code>_POST["submit"]</code>	Objednaj
<code>_FILES["potvrdenie"]</code>	<pre>Array ([name] => 01-prednaska.zip [type] => application/x-zip-compressed [tmp_name] => C:\wamp\tmp\php610A.tmp [error] => 0 [size] => 373)</pre>

Pole `$_FILES['xyz']`

9

- Je to asociatívne (dvojrozmerné) pole s presne definovanými hodnotami (pozri ďalej).
- Existuje len počas behu skriptu.
- Po úspešnom uploade sa vytvorí dočasný súbor na serveri, ktorý treba presunúť na vhodné miesto (+ premenovať).
- Po skončení skriptu sa dočasný súbor automaticky zmaže.

`$_FILES['xyz']` – hodnoty (1)

10

- **`'name'`** – meno pôvodného súboru (u klienta)
- **`'tmp_name'`** – dočasné meno súboru na strane servera s cestou
- **`'type'`** – MIME typ súboru (nastavený klientom – prehliadačom). Napr. Windows nepoznajú formát ogg tak MIME nenastavia. PHP to neopraví, aj keď daný formát pozná.

`$_FILES['xyz']` – hodnoty (2)

11

- **`'size'`** – veľkosť súboru v bytoch
- **`'error'`** – stav prenosu súboru (podarilo sa/chyba).
Všetky možné hodnoty sú uvedené ďalej.
 - Napr.:

```
if ($_FILES['potvrdenie']['error'] == 0)  
    echo 'Súbor bol úspešne uploadovaný.';
```

`$_FILES['xyz']['error']`

12

- **UPLOAD_ERR_OK (0)**: žiadna chyba, súbor bol úspešne uploadovaný
- **UPLOAD_ERR_INI_SIZE (1)**: uploadovaný súbor je väčší ako hodnota **upload_max_filesize** definovaná v `php.ini`
- **UPLOAD_ERR_FORM_SIZE (2)**: uploadovaný súbor je väčší ako hodnota **MAX_FILE_SIZE** definovaná vo formulári
- **UPLOAD_ERR_PARTIAL (3)**: uploadovaný súbor nebol prenesený celý (nastal problém počas uploadovania)
- **UPLOAD_ERR_NO_FILE (4)**: nebol uploadovaný žiadny súbor
- **UPLOAD_ERR_NO_TMP_DIR (6)**: chýba dočasný priečinok
- **UPLOAD_ERR_CANT_WRITE (7)**: problém so zápisom súboru na disk
- **UPLOAD_ERR_EXTENSION (8)**: problém s príponou súboru (nepovolená prípona)

Funkcie pri uploadovaní (1)

13

- **is_uploaded_file(dočasný_súbor)** – vráti TRUE, ak bol zadaný súbor naozaj uploadovaný pomocou formulára (**nie je to podvrh**)

```
if (is_uploaded_file($_FILES['xyz']['tmp_name']))  
... // je to uploadovaný súbor, spracujem ho
```

Funkcie pri uploadovaní (2)

14

- **move_uploaded_file(dočasný_súbor, nový_súbor)** – presunie dočasný uploadovaný súbor na zadané miesto (robí automatickú kontrolu `is_uploaded_file()`)

```
move_uploaded_file($_FILES['xyz']['tmp_name'],  
"nova_cesta/novy_subor.pripona")
```

- **Používateľ, pod ktorým sa vykonávajú php skripty, musí mať právo zápisu do priečinka nova_cesta.**
- Ak súbor na serveri už existuje, prepíše sa!

Úprava odoslania objednávky

15

```
if (isset($_FILES['potvrdenie'])) {  
    $novy_nazov = "";  
    if ($_FILES['potvrdenie']['error'] == UPLOAD_ERR_OK) {  
        if (is_uploaded_file($_FILES['potvrdenie']['tmp_name'])) {  
            $novy_nazov = 'potvrdenia/ . $_SESSION['id_pouz'] . '-' .  
$_FILES['potvrdenie']['name'];  
  
            $podarilosa =  
move_uploaded_file($_FILES['potvrdenie']['tmp_name'],  
$novy_nazov);  
  
            if ($podarilosa) { echo '<p>Súbor bol nahratý na server.</p>'; }  
            else { echo '<p class="chyba">Súbor NEbol nahratý na server.</p>'; }  
        }  
    } else {  
        echo '<p class="chyba">Nastal problém pri uploadovaní súboru</p>';  
    } } }
```

Otázky a problémy

16

- Musí byť vytvorený priečinok potvrdenie s možnosťou zápisu.
- Nekontrolujeme typ súboru (dá sa pomocou MIME typu alebo aspoň pomocou prípony)
- Najprv sa uploaduje súbor a až potom sa vkladá do tabuľky. Čo ak zlyhá INSERT? Na serveri bude zbytočný súbor.
- Ak by sa zmenila objednávka a vložil nový súbor, pôvodný súbor by zostal na serveri – treba vyriešiť pomenovanie súborov.
- Pri mazaní objednávky sa nemaže súbor.

Dôležité premenné v súbore php.ini

17

- **file_uploads** – či je možný upload cez PHP
- **upload_tmp_dir** – kde sa umiestňujú uploadované súbory
- **upload_max_filesize** – max. veľkosť jedného uploadovaného súboru
- **post_max_size** – max. veľkosť POST sekcie (teda všetkých súčastí formulára odosielaného metódou POST)
- **memory_limit** – max. veľkosť pamäte, ktorú vie PHP skript alokovať

Čo musí platiť pri uploadovaní

18

- Ak napr. uploadujeme 3 súbory naraz, potom musí platiť:
 $3 * \text{upload_max_filesize} < \text{post_max_size}$
- Ďalej musí platiť:
 $\text{post_max_size} < \text{memory_limit}$
- $\text{MAX_FILE_SIZE} \leq \text{upload_max_filesize}$
(dodržiavanie `MAX_FILE_SIZE` je na prehliadači => nespoliehať sa na to)

Ďalšie dôležité premenné v súbore php.ini

19

- **max_execution_time** – max. čas vykonávania PHP skriptu (regulovanie vyťaženia servera)
- **max_input_time** – max. čas strávený prenosom "PHP" údajov (POST, GET, upload, ...)
 - Upozornenie: klienti so slabším pripojením môžu na tomto "stroskotať".

Premenná v Apache (httpd.conf)

20

- **LimitRequestBody** – veľkosť "balíka", ktorú Apache akceptuje. (vrátane POST, GET, ...)
- V hierarchii predchádzajúcich (obmedzujúcich) premenných má táto najvyššiu prioritu, lebo je definovaná priamo v Apache (webovom serveri).

Upozornenie

21

- Všetky doterajšie príklady a ukážky fungujú len vtedy, ak neposielame na server príliš veľký súbor (väčší ako niektorá z konštánt **upload_max_filesize** a **post_max_size**).
- **Pri príliš veľkom súbore nebude existovať pole `$_FILES` !**
- Ošetrovanie a spracovanie takéhoto formulára nebudeme preberať. Odporúčame pozrieť rôzne príručky.

Pripomenutie

22

- Projekt č. 2 - termín odovzdania **11.5.2013 do 23:55**
- Skúška – prihlasovanie cez AIS
- Predtermín 16.5 – bude (max. 30 ľudí), prihlasovanie cez AIS
- Všetky podrobnosti sú v systéme Moodle

Ďakujem za pozornosť 😊

23

Priestor na vaše otázky