

Internet programiranje

Rad sa fajlovima u programskom jeziku PHP

Predavač:
Dr Dražen Drašković, docent

Gde se mogu smeštati fajlovi na serveru?

- Fajlovi na serveru se mogu smeštati:
 - u **bazi podataka**, u kolonama tipa tinyblob (do ~65B), blob (do ~65KB), mediumblob (do ~16MB), longblob (do ~4GB)
 - u **nekom folderu na serveru** (obično u nekom podfolderu foldera u kome se nalazi aplikacija)

Fajlovi koji podležu logičkoj strukturi baze podataka

- Fajlovi koji podležu logičkoj strukturi baze podataka su fajlovi koji su na neki način povezani sa podacima iz baze.
- Primer:
u tabeli *Demonstratori* se čuvaju bitni podaci o studentima, a čuvaju se i fajlovi koji predstavljaju slike tih studenata. Svakom studentu odgovara jedna slika.
- Fajlovi koji podležu logičkoj strukturi BP se mogu smeštati u:
 - bazu podataka ili
 - u neki folder na serveru

Smeštanje fajlova u bazu podataka

● Prednosti:

- Svi podaci se nalaze na jednom mestu (u bazi) i povezani su na standardan način
- Nema potrebe voditi računa o problemu poklapanja imena fajlova
- Jedinstveni back-up
- Atomska (nedeljivo) ažuriranje

● Nedostaci:

- Nepotrebno se opterećuje DBMS što usporava rad, posebno ako se radi o velikim fajlovima

Smeštanje fajlova u folder



● Prednosti

- Ne opterećuje se DBMS što ubrzava rad

● Mane

- Prostorna razdeljenost logički povezanih podataka
- Treba nekako veštački obezbediti povezivanje podataka iz baze i odgovarajućih fajlova
- Treba rešiti potencijalni problem preklapanja imena fajlova
- Odvojeni backup baze i foldera sa fajlovima
- Ažuriranje nije atomsko (nedeljivo)

Fajlovi koji ne podležu logičkoj strukturi baze podataka

- Fajlovi koji ne podležu logičkoj strukturi baze podataka tj. nisu ni na koji način povezani sa podacima u bazi, smeštaju se u folder na serveru.
- Fajlovi u veb aplikacijama koje ne koriste bazu podataka se smeštaju u folder na serveru.

Slanje fajlova na server



- Da bi se u okviru HTML forme obezbedila mogućnost za upload fajlova na server, treba uraditi sledeće:
 - U okviru `<form>` taga postaviti sledeće vrednosti atributa:
`enctype="multipart/form-data"` (tip sadržaja forme)
`method="post"` (način slanja forme)
 - Unutar form taga definisati input element koji ispisuje klasičnu "Browse" strukturu
`<input type="file" name="user_file">`

Gde se čuva datoteka na serveru

- Kada korisnik pošalje sliku na server (klikom na dugme Submit u okviru forme), PHP automatski prebacuje datoteku na server i čuva je u privremenoj lokaciji na serveru.
- Ova lokacija se definiše parametrom `upload_file_dir` u datoteci `php.ini`.

Superglobal \$_FILES (#1)



- Podaci o upload-ovanom fajlu se nalaze superglobalnoj promenljivoj \$_FILES, koja se automatski inicijalizuje po prijemu podataka putem HTTP post metode.
- \$_FILES je po strukturi dvodimenzionalni asocijativan niz, gde je prva dimenzija naziv odgovarajućeg input elementa za upload fajla, a druga dimenzija predstavlja naziv određenog atributa fajla.

Superglobal \$_FILES (#2)



● Za slučaj

```
<input type="file" name="file1">
```

podaci će se nalaziti u `$_FILES["file1"]`

● Atributi fajla:

- `$_FILES["file1"]["tmp_name"]` - lokacija fajla na serveru
- `$_FILES["file1"]["name"]` - originalno ime fajla na računaru korisnika
- `$_FILES["file1"]["size"]` - veličina fajla u B
- `$_FILES["file1"]["type"]` - tip datoteke (ako postoji)
- `$_FILES["file1"]["error"]` - podaci o grešci

Upload velikih fajlova (#1)



- Ako se na server prosleđuju fajlovi veći od nekoliko MB, neophodni su dodatni koraci:
 - U HTML formu treba dodati skriveno polje koje će sadržati maksimalnu dozvoljenu veličinu datoteke (u B) koja se može poslati na server:

```
<INPUT TYPE="hidden"  
  NAME="MAX_FILE_SIZE" VALUE="100000">
```
 - Maksimalna količina memorije koja se može dodeliti PHP skriptu mora biti veća od maksimalne veličine datoteke (parametar `memory_limit` u `php.ini`).

Upload velikih fajlova (#2)



- Podesiti maksimalnu veličinu datoteke koja se može prebaciti na server (parametar `upload_max_filesize` u `php.ini`)
- Podesiti maksimalnu veličinu POST zahteva tako da bude veća od maksimalne veličine fajla koji se upload-uje (parametar `post_max_size` u `php.ini`)
- Najduže vreme izvršavanja php skriptova treba podasiti treba podasiti na odgovarajuću vrednost koji omogućava da se postupak slanja datoteke obavi do kraja (parametar `max_execution_time` u `php.ini` - po defaultu 30 sec)

Opšte napomene za smeštanje fajlova u BP

- Upload-ovanom fajlu se pristupa preko atributa "tmp_name", koji sadrži lokaciju (putanju) fajla na serveru
- U kolonu tipa blob (i tipova izvedenih iz blob) upisuje se **binarni sadržaj** upload-ovanog fajla, a sam upis nije ni po čemu specifičan od upisa podataka u kolone drugih tipova
- Podaci iz kolone tipa blob se brišu kao i svi ostali podaci

Redosled koraka pri upisu fajlova u BP

1. Proverava se da li je fajl poslat (upload) na server pomoću funkcije `is_uploaded_file()` ako jeste prelazi se na sledeći korak.
2. Proverava se da li je fajl dozvoljene veličine pomoću funkcije `filesize()` ako jeste prelazi se na sledeći korak.
3. Otvara se fajl pomoću funkcije `fopen()`.
4. Čita se binarni sadržaj poslatog fajla pomoću funkcije `fread()` i pamti u nekoj promenljivoj.
5. Binarni sadržaj poslatog fajla (sadržan u promenljivoj) se prečišćava pomoću funkcije `AddSlashes()`.
6. Upis podataka u bazu, pri čemu se u kolonu tipa blob (ili tipa izvedenog iz blob) upisuje prečišćeni binarni sadržaj poslatog fajla koji se čuva u promenljivoj.

Prikaz slike smeštene u BP (#1)



- Prikaz slike smeštene u BP (u koloni tipa blob) se postiže tako što se okviru `<img>` taga atributu `src` dodeliti vrednost koja predstavlja putanju skripta koji čita i prikazuje sadržaj slike, uz dodatne parametre koji mu se prosleđuju:

```

```

Prikaz slike smeštene u BP (#2)

- Skript `read_and_show_image.php` na osnovu prosleđenog parametra `id_item` (u opštem slučaju se prosleđuje više parametara) čita binarni sadržaj slike iz baze.
- Slika se prikazuje tako što se binarni sadržaj slike dobijen iz baze "štampa" pomoću naredbe `echo`:
`echo $binary_content;`
- Neposredno pre štampanja slike naredbom `echo` veb pregledač se šalje zaglavlje koje definiše MIME tip fajla (resursa) koji se štampa pomoću funkcije `header` (type atribut u `$_FILES["file_name"]`)
`header("Content-Type: image/jpeg");`

Smeštanje fajlova u folder - opšte napomene

- Ideja: u bazi se ne čuva sama slika nego informacija pomoću koje je moguće locirati sliku na serveru (tj. odrediti putanju slike).
- Potrebno je obezbediti:
 - povezivanje podataka iz baze sa odgovarajućim fajlovima i
 - jedinstvenost imenovanja fajlova (kako ne bi došlo do prepisivanja ili do nemogućnosti upload-ovanja)
- Ovi problemi se mogu rešiti primenom određenih konvencija.

Rešenje problema - konvencija



- U tabeli umesto kolone tipa blob staviti kolonu tipa varchar u kojoj se pamti originalno ime fajla.
- Svi fajlovi se čuvaju u jednom folderu db_files.
- Za svaku tabelu u bazi koja "logički" sadrži fajlove kreira se poseban podfolder u folderu db_files čiji je naziv jednak nazivu tabele.
- Za svaku kolonu u tabeli u kojoj se pamti originalno ime fajla kreira se podfolder u folderu koji odgovara toj tabeli čiji je naziv jednak nazivu kolone.
- Fajl se preimenuje, tako što mu se kao prefiks doda vrednost primarnog ključa iz njemu odgovarajućeg reda u tabeli.

Redosled pri upisu podataka



1. Izvrši se upis podataka u bazu;
ako tabela "logički" sadrži fajlove u odgovarajuće kolone se upisuju originalna imena tih fajlova.
Ako je upis uspeo prelazi se na sledeći korak.
2. Smeštanje upload-ovanih fajlova u njima odgovarajuće foldere i izmena imena tih fajlova (u skladu sa uvedenom konvencijom).

Redosled koraka pri smeštanju upload-ovanih fajlova u foldere



1. Provera da li je korisnik upload-ovao neki fajl, pomoću funkcije `is_uploaded_file()`
2. Provera da li je fajl dozvoljenog tipa i da li poseduje adekvatnu ekstenziju (opciono), ispitivanjem `type` i `name` atributa fajla
3. Provera da li je fajl dozvoljene veličine, ispitivanjem `size` atributa fajla ili korišćenjem `filesize()` funkcije
4. Provera da li odredišni direktorijum postoji (trebalo bi da uvek postoji), korišćenjem funkcija `file_exists()` i `is_dir()`
5. Provera da li već postoji fajl sa istim imenom u odredišnom folderu (ne bi trebalo nikad da postoji), pomoću funkcije `file_exists()`
6. Premestiti fajl u odredišni folder (u skladu sa konvencijom), korišćenjem funkcije `move_uploaded_file()`
7. Proveriti da li se premeštanje fajla izvršilo bez greške, ispitivanjem povratne vrednosti funkcije `move_uploaded_file()`

Prikaz slike smeštene u folderu



- Prikaz slike smeštene u folderu se postiže tako što se okviru `<img>` taga atributu `source` dodeliti vrednost koja predstavlja relativnu (u odnosu na tekući skript) putanju slike

```

```

Redosled pri brisanju podataka



1. Pročita se vrednost primarnog ključa i atributa koji predstavlja originalno ime fajla iz reda koji treba obrisati, a zatim se red obriše.
2. Obriše sa fajl iz odgovarajućeg foldera koji je odgovarao ovom redu. Folder se može jednoznačno odrediti ako se koristi navedena konvencija.

Redosled koraka pri brisanju upload-ovanih fajlova iz foldera

1. Proverava se da li postoji fajl koji odgovara redu koji se briše u odgovarajućem folderu, korišćenjem funkcije `file_exists()`.
Ako postoji prelazi se na sledeći korak, u suprotnom se ništa ne preduzima.
2. Briše se postojeći fajl koji odgovara obrisanom redu u tabeli, korišćenjem funkcije `unlink()`.
3. Proverava se da li je brisanje fajla uspešno izvršeno, ispitivanjem povratne vrednosti funkcije `unlink()`.

Funkcije za rad sa direktorijumima (1)

● Čitanje sadržaja direktorijuma

- `opendir()`, `closedir()` i `readdir()`

- **Primer:**

```
$tekuci_dir = '/uploads/';  
$dir = opendir($tekuci_dir);  
echo "<p>Listing direktorijuma:</p><ul>";  
while ($fajl = readdir($dir)) {  
    echo "<li>$fajl</li>";  
}  
echo "</ul>";  
closedir($dir);
```

Funkcije za rad sa direktorijumima (2)

- Učitvanje podataka o tekućem direktorijumu
 - `dirname ($putanja), basename ($putanja)`
 - Ove funkcije daju deo putanje koji prikazuje sadržaj direktorijuma, odnosno ime datoteke.

Funkcije za rad sa direktorijumima (3)

● Pravljenje i brisanje direktorijuma

- `mkdir()`, `rmdir()`
- Funkciji za kreiranje direktorijuma potrebno je da prosledite dva parametra: putanju ka direktorijumu, koja sadrži i ime novog direktorijuma i ovlašćenja za pristup tom direktorijumu, na primer:
`mkdir("/tmp/testing", 0777);`
- Direktorijum koji nameravate da brišete, mora biti prazan.

Rad sa sistemom datoteka (1)

● Dobijanje podataka o datoteci

- Funkcije `fileatime()` i `filemtime()` vraćaju vremensku oznaku kada je poslednji put pristupano datoteci, odnosno direktorijumu.
- Funkcije `fileowner()` i `filegroup()` vraćaju identif. vlasnika (uid), odnosno vlasničke grupe (gid).
- Funkcija `fileperms()` vraća ovlašćenja za rad sa datotekom.
- Funkcija `filetype()` vraća nekoliko podataka o tipu datoteke koja se ispituje (fifo, char, dir, block, link, file i unknown)
- Funkcija `filesize()` vraća veličinu datoteke u bajtovima.

Rad sa sistemom datoteka (2)

- Dobijanje podataka o datoteci

- Funkcije koje ispituju vrednost određenog atributa datoteke vraćaju `true` ili `false`:

```
is_dir()  
is_executable()  
is_file()  
is_link()  
is_readable()  
is_writable()
```

Rad sa sistemom datoteka (3)

● Pravljenje, brisanje i premeštanje datoteka

- Pomoću funkcije `touch` možete napraviti novu datoteku ili izmeniti vreme kada je poslednji put menjan sadržaj postojeće datoteke.
- Datoteke se brišu pomoću funkcije `unlink($ime_datoteke)`. Prototip funkcije:
`unlink($ime_datoteke)`
- Datoteke možete kopirati i premeštati pomoću funkcija `copy` i `rename`. Prototip tih funkcija izgleda ovako:
`copy($izvorna_putanja, $ciljna_putanja)`
`rename($staro_ime, $novo_ime)`
- Funkcija `rename()`, osim preimenovanja, radi premeštanje datoteke s jednog mesta na drugo, jer PHP nema posebnu funkciju za premeštanje!



Upotreba mrežnih funkcija i protokola

Funkcija header()



- `void header($string)`
- Šalje sirovo HTTP zaglavlje.
- Funkcija mora biti pozvana pre nego što se pošalje bilo koji izlaz - HTML tagovi, blanko linije u fajlu ili iz PHP.
To je vrlo česta greška kada se koriste `include/require` ili neke druge funkcije za pristup fajlu.
- Postoje 2 specijalna slučaja pozivanja ove funkcije.

Pozivanje funkcije header()

- Prvi slučaj:

Parametar koji se prosleđuje funkciji počinje sa **HTTP/**, a nakon toga se piše HTTP statusni kod za slanje:

```
header("HTTP/1.0 404 Not Found");
```

ili kraće:

```
header("Status: 404 Not Found");
```

- Drugi slučaj:

Kao parametar se navodi ključna reč **Location:**, a nakon toga lokacija na koju se prelazi (tačnije veb čitač redirektuje na drugu veb stranicu):

```
header("Location: welcome.php");
```

Slanje i primanje poruka e-poštom

- Upotreba jednostavne funkcije `mail()`
- Šalje poruke preko protokola SMTP (*Simple Mail Transfer Protocol*)
- Protokol SMTP omogućava samo slanje poruka
- Protokoli IMAP (*Internet Message Access Protocol*) i POP (*Post Office Protocol*) omogućavaju čitanje poruka sa servera za e-poštu, ali ne i slanje poruka

Funkcija mail()



Metod	Opis
To	Adresa e-pošte na koju se šalje poruka.
Subject	Naslov poruke.
Message	Tekst (sadržaj) poruke.
Additional Headers	Opciono: dodatna zaglavlja (From, Reply-To,...)
Additional Parameters	Opciono: bilo koji dodatni parametri koje želite da pošaljete mejl serveru

Primer - Slanje mejla (1)



`//posalji_mejl.php`

```
<?php
    if (!array_key_exists('Submitted', $_POST)) {
?>
<form method="post" action="Mail.php">
    <input type="hidden" name="Submitted" value="true"/>
    Mail Server: <input type="text" name="Host" size="25"/>
    <br/>
    To: <input type="text" name="To" size="25"/>
    <br/>
    From: <input type="text" name="From" size="25"/>
    <br/>
    Subject: <input type="text" name="Subject" size="25"/>
    <br/>
    <textarea name="Message" cols="50" rows="10">
    </textarea>
    <br/>
    <input type="submit" value="Send Email"/>
</form>
```

Primer - Slanje mejla (2)



`//posalji_mejl.php`

```
<?php
} else {
    ini_set('SMTP', $_POST['Host']);
    $To = $_POST['To'];
    $From = 'From: ' . $_POST['From'];
    $Subject = $_POST['Subject'];
    $Message = $_POST['Message'];

    if(mail($To, $Subject, $Message, $From)) {
        echo "Message Sent";
    } else {
        echo "Message Not Sent";
    }
}
?>
```

- Da bi ovaj primer radio morate imati podešen mejl server. Funkcija `ini_set()` postavlja adresu mejl servera.

Ograničenja mail() funkcije

- Nema podrške za SMTP autentifikaciju.
- Teško se šalju poruke formatirane u HTML-u.
- Teško se prosleđuju prilozi u poruci (*attachments*).

Šta je rešenje?

Koristiti mnogobrojne besplatne klase koje služe za slanje mejlova!



- Vrlo dobra ekstenzija za slanje e-pošte je PHPMailer. Dostupan je besplatno:
<http://phpmailer.sourceforge.net>
- U sledećem primeru je pokazano kako se radi sa PHPMailer klasom.

PHPMailer metode



Metoda	Opis
AddAddress ()	Dodaje polje "to".
AddAttachment ()	Dodaje datoteku za slanje zadavanjem putanje u fajl sistemu.
AddBCC ()	Dodaje polje "bcc".
AddCC ()	Dodaje polje "cc".
AddReplyTo ()	Dodaje polje "Reply-to".
IsHTML ()	Postavlja tip poruke na HTML.
IsSMTP ()	Kaže svojoj klasi da šalje poruke preko SMTP.
Send ()	Funkcija koja kreira poruku, šalje je i vraća true vrednost, a u slučaju da ne pošalje poruku uspešno vraća false.

PHPMailer atributi



Atribut	Opis
AltBody	Postavljanje teksta kao tela poruke.
Body	Postavljanje tela poruke, može biti tekst ili HTML.
ErrorInfo	Poslednje poruke o greškama.
From	Postavlja adresu e-pošte pošiljaoca.
FromName	Dodaje ime pošiljaoca u poruci koja se šalje.
Host	Postavlja SMTP host. Svi hostovi moraju biti razdvojeni sa ;
Password	Postavlja SMTP lozinku.
SMTPAuth	Postavlja SMTP autentifikaciju, koristeći username i password.
Subject	Dodaje naslov poruke.
Username	Postavlja SMTP korisničko ime.
WordWrap	Postavlja odsecanje teksta u telu poruke, nakon zadatog broja znakova.

Primer - Korišćenje PHPMailer(1)

```
<?php                                     // posalji_gmejl.php
    if (!array_key_exists('Submitted',$_POST)) { ?>
<form method="post" action="posalji_gmejl.php">
<input type="hidden" name="Submitted" value="true"/><br/> Mail
    Server: <input type="text" name="Host" size="25"/>
<br/>
    If authentication is required:<br/>
    Username: <input type="text" name="Username" size="25"/><br/>
    Password: <input type="password" name="Password" size="10"/>
<hr/>
    To: <input type="text" name="To" size="25"/><br/>
    From Email: <input type="text" name="From" size="25"/><br/>
    From Name: <input type="text" name="FromName" size="25"/>
<br/> Subject: <input type="text" name="Subject"
size="25"/><br/>
    <textarea name="Message" cols="50" rows="10"></textarea><br/>
    Using HTML: <input type="checkbox" name="HTML"/>
    <input type="submit" value="Send Email"/>
</form>
```

Primer - Korišćenje PHPMailer(2)

```
<?php }
    else { require("class.phpmailer.php");
    $To = $_POST['To'];
    $From = $_POST['From'];
    $FromName = $_POST['FromName'];
    $Subject = $_POST['Subject'];
    $Message = $_POST['Message'];
    $Host = $_POST['Host'];
    if (array_key_exists('HTML', $_POST)) {
        $HTML = true;
        $Mail->Username=$_POST['Username']; //Gmail nalog
        $Mail->Password=$_POST['Password']; //Gmail sifra
    } else {
        $HTML = false;
    }
    $Mail = new PHPMailer();
    $Mail->IsSMTP(); // send via SMTP
    $Mail->Host = $Host; //SMTP server
```

Primer - Korišćenje PHPMailer(3)

```
if (array_key_exists('Username',$_POST)) {  
    $Mail->SMTPAuth=true; // dozvola SMTP autent.  
} else { $Mail->SMTPAuth=false; }  
  
$Mail->SMTPSecure = "ssl";//postavljanje prefiksa serveru  
$Mail->Host        = "smtp.gmail.com"; // GMail SMTP  
$Mail->Port        = 465;              //GMail SMTP port  
  
$Mail->From = $From;           //e-mail posaljioca  
$Mail->FromName = $FromName; //ime posiljaoca  
$Mail->AddAddress($To);  
$Mail->AddReplyTo($From);  
$Mail->WordWrap = 50;  
$Mail->IsHTML($HTML);  
$Mail->Subject = $Subject;  
$Mail->Body = $Message;
```

Primer - Korišćenje PHPMailer(4)

```
if($Mail->Send())
{
    echo "Poruka poslata";
}
else
{
    echo "Poruka nije poslata<br/>";
    echo "GRESKA: " . $Mail->ErrorInfo;
}
}
?>
```

Upotreba FTP protokola



- FTP (*File Transfer Protocol*) je protokol za prenošenje datoteka između umreženih računara.
- PHP podržava i FTP veze, slično kao i HTML veze.
- FTP se koristi za izradu rezervnih kopija datoteka koje čine veb prezentaciju ili pri preslikavanju (engl. *mirroring*) tih datoteka na drugu veb lokaciju.



Pitanja?

Hvala!