

# Building Web Applications – Advanced PHP Programming

PHP

***ITAcademy***

# Sadržaj kursa

- OOPHP
- **MySQL**
- Kolačići i sesije
- Obrada binarnih podataka / POST metod
- Design pattern-i

# Building Web Applications – Advanced PHP Programming

MySQL

***ITAcademy***

# Sadržaj

- PHP API za rad sa bazom podataka
- Konekcija sa bazom
- Obrada podataka
- Obrada rezultata

# PHP API

- PHP sadrži veliki broj predefinisanih funkcija
- PHP odlikuje jednostavna integracija sa velikim brojem postojećih sistema za upravljanjem baza podataka
- Za detalje posetite <http://php.net>

# PHP API

- API
  - Application Programming Interface
  - Definiše klase, metode, funkcije, promenljive koje poziva aplikacija pri radu
  - API može da bude proceduralan ili OO
    - Proceduralan – pozivaju se funkcije API-ja
    - OO – pozivaju se metode instanciranih klasa API-ja

# PHP API

- Konektori (*Connector*)
  - MySQL pod konektorom podrazumeva softver koji omogućuje aplikaciji da se poveže sa MySQL serverom
  - MySQL omogućuje konektore za različite jezike, uključujući i PHP

# PHP API

- Drajer (*Driver*)
  - Softver koji služi za konekciju sa konkretnim tipom baza podataka
  - Sadrže implementaciju (na niskom nivou) protokola za konekciju sa baznim serverom
  - Primeri u PHP-u za rad sa MySQL-om su *MySQL Client Library* i *MySQL Native Driver*



# PHP API

- Ekstenzija (*Extension*)
  - Deo PHP-a koji omogućuje pristup osnovnim *PHP* funkcionalnostima
  - Najčešće izlaže direktno API programeru (ali ne i uvek – može da omogući i samo interfejse za pristup API-ju)

# PHP API

- Za rad sa MySQL-om PHP omogućuje tri API-ja:
  - MySQL
  - mysqli
  - PDO

# PHP API

- **PHP MySQL**

- Originalna *PHP* ekstenzija koja omogućuje rad sa *MySQL* bazom podataka
- Proceduralni pristup
- Do PHP verzije 4.1.3 ovo je jedina MySQL ekstenzija

# PHP API

- **PHP mysqli**
  - MySQL improved
  - Pojavljuje se od verzije 5

# PHP API

- **PHP mysqli**

- Brojna unapređenja u odnosu na MySQL

- Objektno-orijentisani interfejs
    - Omogućuje upite sa parametrima (prepared statements)
    - Omogućuje posebnu metodu za izvršenje više upita iz jednog MySQL poziva (multiple statements)
    - Podržava korišćenje transakcija

# PHP API

- **PDO**
  - PHP Data Object
  - Nivo apstrakcije između aplikacije i baze podataka
  - Labavo vezan (slabo spregnut, eng. *loosely coupled*) – aplikacija ne zna koji DBMS se koristi

# PHP API

- **PDO**

- Prednosti

- Čist kod, lak za korišćenje
    - Jednostavnija promena DBMS-a

- Mane

- Zbog portabilnosti, ne dozvoljava korišćenje naprednijih MySQL funkcionalnosti

# PHP API

- **PDO MySQL Driver**
  - Nivo ispod PDO-a
  - Implementira konkretne MySQL funkcionalnosti
  - Aplikacija poziva PDO koji koristi PDO MySQL Driver za komunikaciju sa serverom
  - Predstavlja jedan od PDO drajvera. Drugi se koriste za npr. Postgre server



# PHP API

- **PHP MySQL Client Library** (*libmysqlclient*)
  - Implementira potrebne protokole
  - Za komunikaciju sa MySQL serverom, sve PHP ekstenzije za MySQL koriste ovaj drajver
  - Ranije je bio jedini MySQL drajver
  - Stari drajver, originalno dizajniran za C aplikacije

# PHP API

- **PHP MySQL Native Driver (*mysqlnd*)**
  - Unapređena verzija *libmysqlclient-a*
  - Sadrži poboljšanja u radu sa memorijom (omogućuje se korišćenje memorije do dozvoljenog PHP limita). Brži je
  - Dostupan od verzije 4.1.3

# PHP API

Comparison of MySQL API options for PHP			
	PHP's mysqli Extension	PDO (Using PDO MySQL Driver and MySQL Native Driver)	PHP's MySQL Extension
PHP version introduced	5.0	5.0	Prior to 3.0
Included with PHP 5.x	yes	yes	Yes
MySQL development status	Active development	Active development as of PHP 5.3	Maintenance only
Recommended by MySQL for new projects	Yes - preferred option	Yes	No
API supports Charsets	Yes	Yes	No
API supports server-side Prepared Statements	Yes	Yes	No
API supports client-side Prepared Statements	No	Yes	No
API supports Stored Procedures	Yes	Yes	No
API supports Multiple Statements	Yes	Most	No
Supports all MySQL 4.1+ functionality	Yes	Most	No

\* Preuzeto sa adrese <http://php.net/manual/en/mysqli.overview.php>

# Konekcija sa bazom podataka

- **MySQL**

- Na slajdovima u nastavku će biti prikazano korišćenje MySQL metoda

# Konekcija sa bazom podataka

- `MYSQL_CONNECT()`
  - Funkcija za uspostavljanje konekcije sa serverom
  - Obavezni parametri su naziv (IP) servera i naziv korisnika

```
mysql_connect($host[:$port][$socket_path],$user[, $pass])
```

# Konekcija sa bazom podataka

- `MYSQL_CONNECT()`
  - Ukoliko je potrebno da se izvrši prekidanje izvršenja skripta, poziva se funkcija `DIE()`

# Konekcija sa bazom podataka

```
<?php
    $host = "localhost";
    $user = "ana";
    $pass = "pass";

    $x = mysql_connect($host, $user, $pass);
    // ili
    $x = mysql_connect($host, $user, $pass) or die("Veza nije mogla biti uspostavljena.");
    //ili
    $x = mysql_connect($host, $user, $pass);
    if (!$x)
    {
        die('Could not connect');
    }

?>
```

# Konekcija sa bazom podataka

- `MYSQL_ERRNO()`
  - Funkcija koja vraća numeričku vrednost greške poslednjeg MySQL iskaza
  - Ako \$veza nije definisana, koristi se trenutno aktivna konekcija

```
mysql_errno([$veza])
```



# Konekcija sa bazom podataka

- `MYSQL_ERROR()`
  - Funkcija koja vraća opisnu (tekstualnu) vrednost greške poslednjeg MySQL iskaza
  - Ako \$veza nije definisana, koristi se trenutno aktivna konekcija

```
mysql_error([$veza])
```

# Konekcija sa bazom podataka

```
<?php
    $host = "localhost";
    $user = "ana";
    $pass = "pass";

    $x = mysql_connect($host, $user, $pass);
    // ili
    $x = mysql_connect($host, $user, $pass) or die("Veza nije mogla biti uspostavljena.");
    //ili
    $x = mysql_connect($host, $user, $pass);
    if (!$x)
    {
        die('Greska (' . mysql_errno() . '): ' . mysql_error());
    }

?>
```

# Konekcija sa bazom podataka

- `MYSQL_SELECT_DB()`
  - Funkcija bira MySQL bazu nad kojom će se izvršavati upiti
  - Ako \$veza nije definisana, koristi se trenutno aktivna konekcija

```
mysql_select_db($baza[, $veza])
```

MySQL

# Konekcija sa bazom podataka

```
<?php
    ...
    $x = mysql_connect($host,$user,$pass);
    $y = mysql_select_db($baza);
    // Rezultat je: baza je odabrana
?>
```

# Obrada podataka

- MYSQL\_QUERY()
  - Funkcija prosleđuje SQL upit ka konektovanom serveru
  - Iskaz može da se proširi korišćenjem funkcije DIE()

```
mysql_query($iskaz[, $veza])
```

# Obrada podataka

```
<?php
    $naziv=$_POST['naziv'];
    $oblast=$_POST['oblast'];
    $query = " SELECT * FROM odgovori WHERE poeni <> 0 ";

    //izvrsavanje upita
    $x = mysql_query($query);
    //ili
    $query =mysql_query("SELECT * FROM odgovori WHERE poeni <> 0");
?>
```

# Obrada rezultata

- MYSQL\_NUM\_ROWS ()
  - Funkcija vraća broj redova u rezultatu
  - Koristi se uz naredbu SELECT

```
mysql_num_rows($rezultat[, $veza])
```

# Obrada rezultata

- `MYSQL_AFFECTED_ROWS()`
  - Funkcija vraća broj redova izmenjenih poslednjim upitom (radi uz UPDATE, INSERT, DELETE)
  - Za DELETE upit (bez WHERE uslova), briše se cela tabela, te funkcija vraća 0

```
mysql_affected_rows([$rezultat])
```



# Obrada rezultata

- `MYSQL_FETCH_ARRAY()`
  - Funkcija preuzima rezultate upita koje je vratila baza i smešta ih u niz
  - Nizu može da se pristupi i kao indeksnom i kao asocijativnom

```
mysql_fetch_array($rezultat[, $array_vrsta])
```

# Obrada rezultata

```
<?php
    $x = mysql_query("SELECT ime, prezime FROM tabela");
    while ($red = mysql_fetch_array($x))
    {
        echo $red['ime']." ".$red['prezime'];
        // ili echo $red[0]." ".$red[1];
    }
    mysql_free_result($x);
    // Rezultat je: sva imena i prezimena iz tabela tabela
?>
```

# Obrada rezultata

- `MYSQL_FETCH_ASSOC()`
  - Funkcija preuzima rezultate upita koje je vratila baza i smešta ih u niz
  - Nizu može da se pristupi samo kao asocijativnom nizu

```
mysql_fetch_assoc($rezultat)
```

# Obrada rezultata

```
<?php
    $x = mysql_query("SELECT ime, prezime FROM tabela");
    while ($red = mysql_fetch_assoc($x))
    {
        echo $red['ime']." ".$red['prezime'];
    }

    mysql_free_result($x);
    // Rezultat je: sva imena i prezimena iz tabela tabela
?>
```

# Obrada rezultata

- `MYSQL_CLOSE()`
  - Funkcija zatvara aktivnu konekciju sa bazom

```
mysql_close ()
```

# Zadatak 1

- Napraviti formular za unos podataka u tabelu *Knjiga*.
  - Formulan definisati na *html* stranici
  - Potvrdom formulara izvršiti *php* skript
  - Izvršiti validaciju podataka

# Zadatak 2

- Napraviti stranicu za pregled Izdavača
  - Omogućiti html stranicu sa formularom za opcioni unos naziva izdavača ili statusa
  - Potvrdom akcije, korisniku se prikazuju filtrirani podaci

# Building Web Applications – Advanced PHP Programming

MySQL

***ITAcademy***