



Distance Learning System

Advanced PHP Programming

MySQLi

Connect – Proceduralni pristup


- MySQLi nudi API za proceduralni ali i za OOP pristup.
- Proceduralnim pristupom se može uraditi sve što se može uraditi i API-em za OOP.
- Proceduralni pristup može biti nešto jednostavniji za upotrebu.
- Primer:

```
$conn = mysqli_connect($DBServer, $DBUser, $DBPass, $DBName);  
  
// check connection  
if (mysqli_connect_errno()) {  
die('Database connection failed: ' . mysqli_connect_error());  
}
```

Connect – OOP prisup

```
$DBServer = 'server name or IP address'; // e.g 'localhost' or '192.168.1.100'
$DBUser = 'DB_USER';
$DBPass = 'DB_PASSWORD';
$DBName = 'DB_NAME';
$conn = new mysqli($DBServer, $DBUser, $DBPass, $DBName);

// check connection
if ($conn->connect_error) {
    die('Database connection failed: ' . $conn->connect_error);
}
```



Zatvaranje konekcije

- Sve konekcije ka bazi biće automatski zatvorene onog trenutka kada se skripta završi.
- Konekcija sa bazom se može zatvoriti i pre toga.
- Primer (procedural):
 - **mysqli_close(\$conn);**
- Primer (OOP):
 - **\$conn->close();**

Kreiranje baze podataka

```
$DBServer = '127.0.0.1';
$DBUser = 'root';
$DBPass = '';
$conn = new mysqli($DBServer, $DBUser, $DBPass);

// check connection
if ($conn->connect_error) {
    die('Database connection failed: ' . $conn->connect_error);
}

$query = <<<SQL CREATE DATABASE IF NOT EXISTS test_db_oop;
USE test_db_oop;
CREATE TABLE IF NOT EXISTS moja_tabela_oop (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50)
);
SQL;

$conn->multi_query($query);

$conn->close();
```

Unos podataka u bazu

```
$DBServer = '127.0.0.1';
$DBUser = 'root';
$DBPass = '';
$conn = new mysqli($DBServer, $DBUser, $DBPass, 'test_db_oop');

// check connection
if ($conn->connect_error) {
    die('Database connection failed: ' . $conn->connect_error);
}

$query = "INSERT INTO moja_tabela_oop (name) VALUES ('Petar')";
$conn->query($query);
$query = "SELECT * FROM moja_tabela_oop";
$res = $conn->query($query);

while($row = $res->fetch_array()){
    var_dump($row);
}
```

fetch_... metode

- Postoji više načina za preuzimanje podataka koji stignu iz baze kao rezultat izvršavanja upita.
- Rezultat se može posmatrati kao asocijativni niz, numerički niz ili kao oba niza istovremeno.
 - fetch_all()
 - fetch_all(MYSQLI_ASSOC)
 - fetch_all(MYSQLI_NUM)
 - fetch_all(MYSQLI_BOTH)
 - fetch_assoc()
 - fetch_row()
 - fetch_array()
- Takođe, rezultat može biti vraćen i kao novi objekat:
 - fetch_object()

SQL injection zaštita

- SQL upit može biti potencijalno opasan po bezbednost podataka u bazi.
- Osnovni princip zaštite je obrada stringa pre izvršavanja upita.
- Primer:

```
$v1="'" . $conn->real_escape_string('coll_value') . "'";  
$sql="INSERT INTO tbl (coll_varchar, col2_number) VALUES ($v1,10)";  
if($conn->query($sql) === false) {  
    die($conn->error);  
} else {  
    $last_inserted_id = $conn->insert_id;  
    $affected_rows = $conn->affected_rows;  
}
```