

## Blok 55

### Zadatak

Kreirati bazu podataka *blok55*. Unutar te baze kreirati tabelu transakcije koja sadrži sledeća polja: *id* (primarni ključ), *kolicina* i *naziv*. Kreirati nekoliko unosa u tabelu ručno ili automatski. Potrebno je svakom unosu dati jedinstveni broj isporuke, kratak naziv i količinu koja se isporučuje. Napraviti fajl *db.php* kojim se uspostavlja konekcija sa bazom. Zatim kreirati REST API fajl *api.php*, u kome se za dati ID isporuke vraćaju u JSON formatu sve potrebne informacije o isporuci iz tabele. Na primer, informacije o konkretnoj isporuci čiji je ID 12345 treba da budu dobijene pomoću zahteva oblika <http://localhost/api.php?id=12345>. Prepraviti *.htaccess* fajl tako da skraćena verzija adrese bude <http://localhost/12345>. Napraviti HTML formu kojom se za unesen ID isporuke u tekstualno polje, klikom na dugme Submit, ispisuju u svi relevantni podaci o datoj isporuci.

### Rešenje

Najpre je potrebno u *phpmyadmin*-u da kreiramo bazu podataka sa imenom *blok55*. To se postiže sledećom SQL naredbom:

```
CREATE DATABASE blok55;
```

Za kreiranje tabele *transakcije*, vrednosti *id* i *kolicina* će biti celi brojevi, a naziv string. Primarni ključ je polje *id*, a sva polja moraju imati unesene vrednosti. Sledeći SQL kod kreira traženu tabelu:

```
CREATE TABLE IF NOT EXISTS `transakcije` (  
  `id` int(10) NOT NULL,  
  `kolicina` int(10) NOT NULL,  
  `naziv` varchar(20) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

U tabelu ćemo zadati nekoliko početnih unosa. Ovo se može odraditi ručno, a SQL kod koji unosi tri proizvoda čiji su ID-jevi 1001, 1002 i 1003, sa odgovarajućim količinama i nazivima može biti oblika:

```
INSERT INTO `transakcije` (`id`, `kolicina`, `naziv`) VALUES  
(1001, 10, 'cokolada'),  
(1002, 50, 'bombona'),  
(1003, 20, 'keks');
```

U fajlu *db.php* pokušavamo pomoću funkcije *mysqli\_connect* da se konektujemo sa bazom. Ukoliko nije moguća konekcija, ispisujemo odgovarajuću poruku. Podrazumevano je postavljeno da se radi na lokalnom serveru, da je korisničko ime root i šifra prazna. Poslednji argument je ime baze podataka. Fajl *db.php* je sledećeg oblika:

```
<?php
```

```
$con = mysqli_connect("localhost", "root", "", "blok55");  
if (mysqli_connect_errno()) {  
    echo "Greska prilikom povezivanja sa bazom."  
    die();  
}
```

```
}
```

```
?>
```

Sada je potrebno da kreiramo *api.php* fajl, u kome se na osnovu zadatog URL-a dobijaju informacije o jednom unosu iz tabele u JSON formatu. Na početku je potrebno da uključimo *db.php*, zbog konekcije sa bazom. Nakon toga, funkcijom *mysqli\_query* pokušavamo odgovarajućom SQL naredbom da izvučemo isporuku koja ima traženi ID. Ukoliko je nađen unos, on je jedinstven, i u tom slučaju ispisujemo potrebne informacije i zatvaramo konekciju sa bazom. Inače, ukoliko unos nije nađen ili je došlo do greške prilikom zahteva, ispisuje se odgovarajuća poruka. Funkcija kojom se izvučene informacije iz tabele pretvaraju u JSON format je *json\_encode*. Fajl *api.php* izgleda ovako:

```
<?php
```

```
header("Content-Type:application/json");
```

```
if (isset($_GET['id']) && $_GET['id'] != "") {  
    include('db.php');  
    $id = $_GET['id'];  
    $result = mysqli_query($con, "SELECT * FROM `transakcije` WHERE id =  
$id");  
    if (mysqli_num_rows($result) > 0) {  
        $row = mysqli_fetch_array($result);  
        $kolicina = $row['kolicina'];  
        $naziv = $row['naziv'];  
        response($id, $kolicina, $naziv);  
        mysqli_close($con);  
    } else {  
        response(NULL, NULL, "Nije nadjen unos.");  
    }  
} else {  
    response(NULL, NULL, "Greska prilikom zahteva.");  
}
```

```
function response($id, $kolicina, $naziv) {  
    $response['id'] = $id;  
    $response['kolicina'] = $kolicina;  
    $response['naziv'] = $naziv;  
    $json_response = json_encode($response);  
    echo $json_response;  
}
```

```
?>
```

U ovom trenutku možemo izvući informaciju o jednoj isporuci zadavanjem odgovarajuće adrese. Na primer, za zahtev <http://localhost/api.php?id=1001>, ukoliko postoji unos čiji je ID jednak 1001, naziv čokolada i količina 10, ispis u JSON formatu je: `{"id":"1001","kolicina":"10","naziv":"cokolada"}`. Da bismo podesili da zahtev bude oblika <http://localhost/1001>, treba da kreiramo *.htaccess* fajl, gde se željeni efekat može postići na sledeći način:

RewriteEngine On

RewriteRule ^([0-9]+)\$ api.php?id=\$1

U drugoj liniji smo naznačili da se celina *api.php?id=broj*, gde broj predstavlja neki ID, menja samo sa broj. Opšti princip za broj je postignut regularnim izrazom *[0-9]+*, koji označava da se cifre od 0 do 9 mogu ponavljati jednom ili više puta (drugim rečima, taj regularan izraz predstavlja neki ceo broj).

Na kraju ćemo napraviti jednostavnu HTML formu *index.html* i odgovarajući PHP fajl koji prima zahteve. HTML forma sadrži labelu, tekstualno polje i dugme *Submit*, i sledećeg je oblika:

```
<html>

<body>

<h3>Informacije o isporukama</h3>

<form action="result.php" method="POST">
    <label>Unesite ID:</label><br/>
    <input type="text" name="id" required/>    <br>
    <button type="submit" name="submit">Submit</button>
</form>

</body>

</html>
```

PHP fajlom kojim se primaju zahtevi se generiše tabela, gde se ispisuju vrednosti o zahtevanoj isporuci. Naravno, ti podaci mogu biti ispisani u bilo kom drugom obliku. Fajl *result.php* izgleda ovako:

```
<?php

if (isset($_POST['id']) && $_POST['id']!="") {
    $id = $_POST['id'];
    $url = "http://localhost/".$id;

    $client = curl_init($url);
    curl_setopt($client, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($client);

    $result = json_decode($response);

    echo "<table>";
    echo "<tr><td><b>ID:</b></td><td>$result->id</td></tr>";
    echo "<tr><td><b>Kolicina:</b></td><td>$result->kolicina</td></tr>";
    echo "<tr><td><b>Naziv:</b></td><td>$result->naziv</td></tr>";
    echo "</table>";
}

?>
```