

# Osnove izrade PHP aplikacija

## Sjednice

Izradio: Davor Cihlar

# Plan

---

- ▶ Što ćemo naučiti
  - ▶ Zaglavlja
  - ▶ Kolačiće
  - ▶ Sjednice
  - ▶ Prijava na sustav

# HTTP

---

- ▶ Kako bi izbjegli moguće probleme i imali dublje shvaćanje zaglavlja i sjednica, potrebno je znati kako radi HTTP:
  - ▶ Korisnik (preglednik) šalje zahtjev za stranicom
    - ▶ Zahtjev sadrži koju stranicu preglednik traži, ali i neke informacije o pregledniku, kao i sve kolačiće vezane uz domenu kojoj se pristupa
  - ▶ Poslužitelj šalje odgovor
    - ▶ Odgovor se sastoji od zaglavlja i podataka
      - Zaglavlje javlja o uspješnosti dohvaćanja zatražene stranice, ali i neke informacije o poslužitelju te zahtjeve za dodavanjem kolačića

# HTTP – primjer (1)

---

## ▶ Preglednik šalje:

### ▶ **GET /index.php HTTP/1.1**

Host: php.fer.hr

User-Agent: Mozilla/5.0 (X11; U; Linux x86\_64; en-US; rv:1.9.1.17) Gecko/20110302 Icedeasel/3.5.17

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7

Cookie: PHPSESSID=abcdefghijklmnopqrstuvwz12

## ▶ Poslužitelj odgovara:

### ▶ **HTTP/1.1 200 OK**

Date: Sun, 13 Mar 2011 17:07:50 GMT

Server: Apache/2.2.16 (Ubuntu)

Content-Encoding: gzip

Content-Length: 291

Content-Type: text/html

*<podaci, kompresirani metodom gzip>*

# HTTP – primjer (2)

---

- ▶ Što dobijemo kao odgovor ako zatražimo nepostojeću stranicu?

- ▶ **GET /bla.php HTTP/1.1**

Host: php.fer.hr

- ▶ **HTTP/1.1 404 Not Found**

Date: Sun, 13 Mar 2011 17:19:06 GMT

Server: Apache/2.2.16 (Ubuntu)

Content-Length: 281

Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
<html><head>
```

```
<title>404 Not Found</title>
```

```
</head><body>
```

```
<h1>Not Found</h1>
```

```
<p>The requested URL /bla.php was not found ...
```

# Zaglavlja

---

- ▶ Zaglavlja je ponekad potrebno mijenjati ili dodavati
  - ▶ Npr. želimo korisniku javiti grešku "403 Forbidden" kada pokuša pristupiti zabranjenoj stranici
- ▶ Zaglavlja nije moguće mijenjati nakon što se počne slati podatkovni dio odgovora!
  - ▶ Prije bilo kakvog generiranja HTML-a potrebno je obraditi sve što može generirati zaglavlja
    - ▶ `header`, `session_start`, obrada kolačića, ...

# Izmjena zaglavlja

---

- ▶ Zaglavlja možemo ručno mijenjati pomoću funkcije `header`
- ▶ Primjer:
  - ▶ 

```
// izmijeni zaglavlje
header("HTTP/1.1 403 Forbidden");
// prikaze grešku
include "greska403.html";
// zaustavi skriptu
exit;
```
- ▶ Opisi mogućih tipova odgovora:
  - ▶ <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

# Ostala korisna zaglavlja

---

- ▶ `header("Content-Type: text/plain");`
  - ▶ Javlja da je rezultat skripte običan tekst (ne HTML)
  - ▶ Moguće je javiti i da je rezultat nešto drugo:
    - ▶ image/png, image/jpeg, application/zip, text/xml, ...
- ▶ `header("Content-Length: 4444");`
  - ▶ PHP ne može znati koliko dugi ispis će biti
  - ▶ Ovo pomaže preglednicima kako bi znali koliki postotak podataka je stigao s poslužitelja
- ▶ `header("Location: index.php");`
  - ▶ Preusmjerenje preglednika na index.php

# Zadatak (1)

---



- ▶ Omogućite dohvaćanje slika preko PHP skripte
  - ▶ Naziv slike se odabire preko GET parametara
- ▶ Stvorite mapu "slike" i u nju postavite nekoliko slika tipa JPEG
  - ▶ Slikama postavite privilegije na 600
    - ▶ Za napredne: moguće je koristiti i .htaccess
    - ▶ Sada slikama više nije moguće pristupiti direktno
- ▶ Kako bi PHP skripta vratila sliku, potrebno je izmijeniti zaglavlje Content-Type; postavite i Content-Length
- ▶ Hintovi: `file_get_contents+strlen` ili `readfile+filesize`

# Kolačići

---

- ▶ Kolačići omogućuju pohranu podataka u pregledniku
  - ▶ Omogućuje automatski login, pamćenje lokalnih postavaka, ...
- ▶ Preglednik šalje kolačiće prilikom zahtjeva
- ▶ Kolačići su vezani uz domenu
  - ▶ Ne šalju se kolačići s npr. [www.google.com](http://www.google.com) na [php.fer.hr](http://php.fer.hr)
- ▶ Kolačići imaju rok trajanja!

# Dohvaćanje kolačića

---

- ▶ Svi kolačići nalaze se unutar superglobalne varijable `$_COOKIE`
- ▶ Kolačiće je moguće dohvatiti u bilo kojem trenutku jer ih preglednik šalje prilikom svakog zahtjeva za nekom stranicom
- ▶ Primjer:
  - ▶ `var_dump($_COOKIE);`

# Slanje kolačića

---

- ▶ Kolačiće je moguće slati pregledniku pomoću funkcije `setcookie`
  - ▶ Funkcija ima mnogo pretpostavljenih parametara koje nećemo objašnjavati
  - ▶ Prvi parametar je ime, a drugi sadržaj kolačića
  - ▶ Treći parametar je rok valjanosti kolačića
    - ▶ Ako nije zadan, pretpostavljena vrijednost je 0 koja označava da traje dok se preglednik ne zatvori
- ▶ Primjer:
  - ▶ `setcookie('tortica', '42');`
  - ▶ <http://php.net/manual/en/function.setcookie.php>

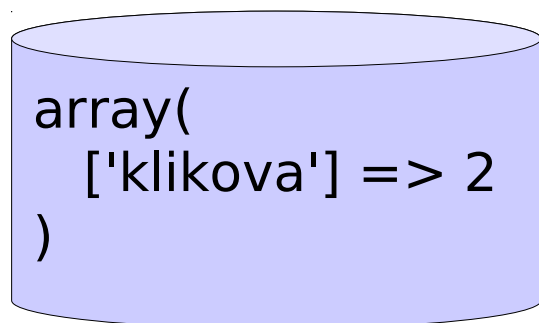
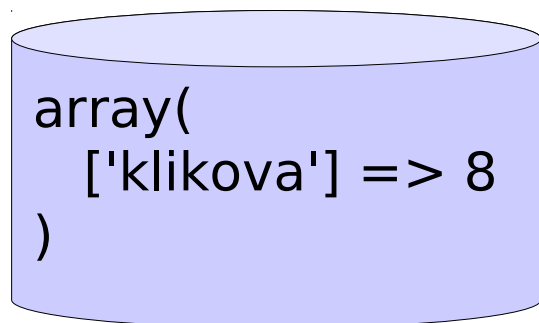
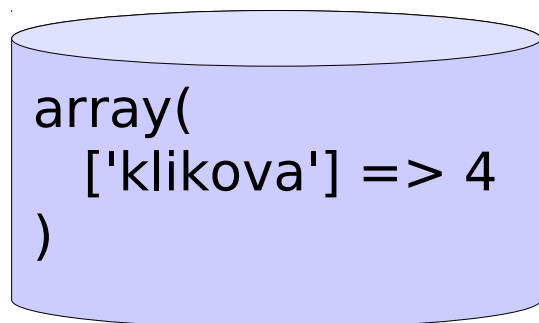
# Sjednice (1)

---

- ▶ Omogućuju privremeno pamćenje podataka **na poslužitelju** za pojedinog korisnika web aplikacije
- ▶ Zasebna sjednica za svakog korisnika
- ▶ Sjednica traje od kad korisnik počne s korištenjem web aplikacije, pa sve do isteka ili brisanja sjednice
  - ▶ Sjednica ističe kad korisnik zatvori preglednik ili nakon nekog određenog vremena
- ▶ PHP raspoznaje sjednice (korisnike) po kolačiću **PHPSESSID**

# Sjednice (2)

## Podaci sjednice:



## Korisnici i kolačići:



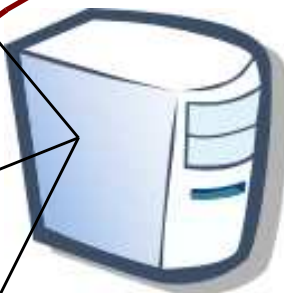
PHPSESSID =  
a1a11f05d...



PHPSESSID =  
b0b50c4...



PHPSESSID =  
1e5e311a...



**Poslužitelj**

# Započinjanje sjednica

---

- ▶ Sjednice započinju pozivom na funkciju `session_start`
  - ▶ Funkcija stvara novu sjednicu ako nije poslan kolačić `PHPSESSID` te šalje taj kolačić
    - ▶ Potrebno pozvati prije bilo kakvog ispisa!
  - ▶ Prilikom svakog idućeg posjeta stranici, preglednik šalje kolačić po kojem `session_start` nalazi postojeću sjednicu

# Pristup varijablama sjednice

---

- ▶ Varijable sjednice nalaze se unutar superglobalnog asocijativnog polja `$_SESSION`
- ▶ `$_SESSION` se prilikom završetka skripte sprema u datoteku, a učitava iz datoteke prilikom `session_start`
  - ▶ Nije moguće spremiti varijable tipa `resource`!
  - ▶ Za napredne: Mogući problemi prilikom spremanja instance, proučiti serijalizaciju
- ▶ `$_SESSION` je moguće koristiti u bilo kojem trenutku, ali nakon poziva na funkciju `session_start`

# Primjer (1)

---



- ▶ Isprobajte sljedeći kod:

```
session_start();  
if (!empty($_GET))  
    $_SESSION = array_merge($_SESSION,  
                             $_GET);  
  
var_dump($_SESSION);
```

- ▶ Tako da posjetite stranicu s ovim parametrima:

- ▶ ?x=42
- ▶ ?a=4
- ▶ ? (*bez parametara*)
  - ▶ Što možete zaključiti da se događa?
  - ▶ Ponovno pokrenite preglednik i posjetite istu stranicu



- ▶ Sjednicu je moguće uništiti funkcijom `session_destroy`
- ▶ Kolačić i `$_SESSION` ostaju nepromijenjeni
  - ▶ Kolačić ostaje dok ga preglednik ne izbriše
  - ▶ `$_SESSION` ostaje samo dok se skripta ne izvrši
  - ▶ Preporučeno izbrisati `$_SESSION`:
    - ▶ `$_SESSION = array();`

# Prijava na sustav

---

- ▶ Sjednice omogućuju pamćenje da li je neki korisnik prijavljen
- ▶ U sjednicu je moguće zapisati npr. korisnik=*ID*, pri čemu je *ID* identifikator prijavljenog korisnika
  - ▶ Ako u sjednici postoji varijabla "korisnik", znamo da je korisnik prijavljen i nije potrebno tražiti prijavu
  - ▶ Moguće je spremiti i druge podatke o korisniku kako ne bi trebali pristupati bazi

# Spremanje lozinki u bazu (1)

---

- ▶ Iz sigurnosnih razloga nije preporučeno direktno spremanje lozinke!
- ▶ Lozinke se spremaju tako da se izračuna sažetak od lozinke kojoj je pridodan neki konstantni podatak
  - ▶ Sažetak je moguće izračunati funkcijom md5 (128 bita) ili sha1 (160 bita, sigurnije!)
  - ▶ Primjerice:
    - ▶ 

```
function sazmi($pass) {  
    return sha1("slanasol$pass");  
}
```

## Spremanje lozinki u bazu (2)

---

- ▶ Rezultat funkcije sha1 je 40-znakovni string
  - ▶ `echo sha1("sazmi me");`
  - ▶ 8c1d5a7d7794b00788e45f352daa264170a24983
- ▶ Preporučeno ga je spremati u CHAR(40) tip podatka, radije nego VARCHAR(40) jer je uvijek iste duljine

# Provjera uspješnosti prijave

---

- ▶ Lozinku dobivenu prilikom prijave nije moguće direktno usporediti s onom spremljenom u bazi
- ▶ Potrebno je napraviti istu operaciju sažimanja i tek onda je moguće usporediti
- ▶ Primjer:
  - ▶ 

```
$q = $db->prepare("SELECT * FROM korisnici  
WHERE username=? AND password=?");  
$q->execute(array(  
    $_POST['user'], sazmi($_POST['pass'])  
));
```
  - ▶ Ako \$q nema rezultata, prijava nije uspješna

## Zadatak (2)

---



- ▶ Napraviti (pod)sustav za prijavu i registraciju korisnika
- ▶ `index.php` mora preusmjeravati na `login.php` ukoliko korisnik nije prijavljen
  - ▶ `login.php` ima obrazac s ovim `input` elementima:
    - ▶ `text` - za korisničko ime
    - ▶ `password` - za lozinku
    - ▶ `submit` - posebno za registraciju i za prijavu
- ▶ U protivnom samo prikaže da je korisnik prijavljen i omogući odjavu korisnika
  - ▶ Paziti da se ne prikazuje i prilikom preusmjeravanja! (`exit` nakon `header`)