Programiranje za Internet

HTTP *cookie* i sesije u PHP-u

Maja Štula

ak. god. 2011/2012

HTTP *cookie*

- Jedan od nedostatak HTTP protokola je to što je HTTP protokol *stateless* protokol tj. HTTP protokol ne pamti prethodna stanja ili prethodne HTTP zahtjeve.
- Svaki HTTP zahtjev za HTTP server je posebna naredba.
- Jedna od metoda praćenje pojedine sesije (komunikacije između klijenta i servera) je *cookie*.
- HTTP *cookie* ili Web *cookie* je skup informacija u tekstualnom obliku koje server pošalje klijentu (IE, Firefox,…) i koje klijent onda šalje nazad tom istom serveru u sklopu svakog budućeg zahtjeva.

HTTP *cookie*

- HTTP *cookieji* se obično koriste kod autentikacije korisnika, kod održavanja specifičnih podataka o korisniku i sl. tj. omogućavaju kreiranje sesije sa stanjem između klijenta i servera koji razmjenjuju HTTP poruke.
- Ideju o implementaciji *cookija* za praćenje sesije uveo je 1994. Lou Montulli jedan od ljudi iz Netscape Communications-a prilikom razvoja aplikacije za e-trgovinu za virtualna *shopping* kolica (kako bi aplikacija mogla pamtiti što je sve korisnik već "ubacio" u kolica).

HTTP *cookie*

- John Giannandrea i Montulli su napisali prvu specifikaciju Netscape *cookie-ja* koja je implementirana u 0.9beta verziji Netscapea.
- Prvo korištenje *cookija* bila je provjera da li je korisnik već pristupao Netscape Web site-u.
- Njihov patent je odobren 1998.
- Podrška za *cookije* je implementirana u IE verziji 2 iz 1995.

HTTP *cookie*

- Trenutni standard za *cookije* (HTTP State Management Mechanism) definiran je RFC-om 2965 iz 2000. (http://www.ietf.org/rfc/rfc2965.txt)
- Standard opisuje tri nova HTTP zaglavlja
  - Cookie,
  - Cookie2 i
  - Set-Cookie2
- koja prenose informacije o stanju konekcije između servera i korisničkog agenta. Korisnički agent (*user agent*) je klijentska aplikacija koja se koristi za pristup WWW.

HTTP *cookie*

- Praktične implementacije korisničkih agenata ograničavaju broj i veličinu *cookija* koji se mogu čuvati.
- Preporuka RFC 2965 standarda je bez ograničenja, ili barem da korisnički agent može pohraniti:
  - 300 *cookija*
  - 4096 bajtova po *cooki*ju
  - 20 *cookija* za jednu domenu ili server

HTTP *cookie*

- Server započinje sesiju na način da klijentu u odgovoru na zahtjev pošalje zaglavlje Set-Cookie (starija verzija) ili Set-Cookie2.
- Korisnički agent vraća serveru zahtjev sa Cookie zaglavljem u svrhu nastavka sesije.
- Server može ignorirati odgovor ili ga koristiti da odredi stanje sesije. Može klijentu poslati odgovor sa Set-Cookie ili Set-Cookie2 zaglavljem s istim ili s različitim podacima.
- Server završava sesiju šaljući klijentu Set-Cookie ili Set-Cookie2 zaglavlje s Max-Age=0.

Krađa HTTP *cookija*

- Zaštićeni podaci (korisničko ime, lozinka i sl.) ne bi se trebali slati *cookijem* jer se *cookie* šalje kao obična tekstualna datoteka.
- Stoga je te podatke potrebno zaštiti na neki način (kodiranjem, korištenjem *https*, …) .

PHP *cookie*

- setcookie() funkcija definira *cookie* koji se šalje s ostalim HTTP zaglavljima klijentu koji pristupa stranici.

bool setcookie (string name [, string value [, int expire [, string path [, string domain [, bool secure]]]]])

- Svi argumenti su opcionalni osim *name* argumenta.

Postavljanje na IE prava za *cookije*: Meni→Tools→Internet Options→Privacy→Advanced

Postavljanje na FireFox-u prava za *cookije*: Meni→Tools→Options→Privacy→Cookies

PRIMJER: www.fesb.hr/~kiki/cookie_stranica.php

PHP *cookie*

Za 30 dana

```
<?php
$vracen=setcookie ('TestCookie', 'Nesto za
    cookie',time()+60*60*24*30 , '/~kiki/', 'www.fesb.hr', 0);
?>
<html>
<body>
Upravo ste primili moj cookie!
</body>
</html>
```

PHP *cookie*

- Osim para ime/vrijednost, *cookie* može imati krajnji rok (*expiration date*), put i ime domene te da li je *cookie* namijenjen samo za sigurne konekcije (HTTPS).

  Set-Cookie: TestCookie=Nesto za cookie; expires=Fri, 11-Jun-2006 23:59:59 GMT ; path=/~kiki/; domain=www.fesb.hr

- Domena i put određuju kojim URL zahtjevima korisnički agent šalje i podatke iz *cookija*. Ukoliko nisu navedeni u *cookiju* podrazumijevaju se domena i put odakle je dohvaćen *cookie*.
- Krajnji rok određuje kada će pretraživač obrisati *cookie*. Ukoliko nije naveden briše se nakon zatvaranja pretraživača. Navođenjem krajnjeg roka *cookie* može postojati i nakon zatvaranja pretraživača pa se *cookiji* s krajnjim rokom nazivaju trajni (*persistent*).

Sesija

- Osim *cookija* koji su integrirani u HTTP protokol za održavanje HTTP sesije koriste se još dva mehanizma:
  - parametri preko kojih se održava sesija prenose se kao dio URI-ja,
  - parametri preko kojih se održava sesija prenose se kao skrivena polja preko HTML forme

- Često se identifikatori sesije (*session ID*) koriste ne samo za identificiranje sesije nego i za identificiranje korisnika sesije tj. za autorizaciju. Uobičajeno se nakon logiranja, kada se korisnik autenticira sa svojim korisničkim podacima, generira statički identifikator sesije koji je privremena statička lozinka.

PHP sesija

- PHP pohranjuje ID sesije u cookie. Ukoliko su cookiji onemogućeni PHP će pokušati ID prenijeti preko URL-a.
- Za automatski prijenos identifikatora sesije preko URL-a opcija session.use_trans_sid u konfiguracijskoj datoteci php interpretera php.ini treba biti postavljena u 1 :

  session.use_trans_sid = 0

- Defaultno je vrijednost opcije session.use_trans_sid postavljena u 0. U tom slučaju (ako se neće koristiti cookie) identifikator sesije se može prenositi eksplicitno kao dodatak URI referenci.

PHP sesija

- Podrška za sesiju omogućava registriranje proizvoljnog broja varijabli na strani servera.
- Ukoliko je uključena automatska podrška za sesiju (session.auto_start PHP parametar postavljen na 1) ili na zahtjev skripte (session_start() ili session_register() funkcija) server provjerava da li je određeni identifikator sesije koji je primljen u HTTP zahtjevu već generiran i u tom slučaju server postavlja iste uvjete (obično vrijednosti određenih varijabli) kao što su bili pri kreiranju identifikatora.
- Sesija se neće pohraniti (na strani servera – obično /tmp/ direktorij) sve dok se varijable sesije ne registriraju funkcijom session_register() ili dodavanjem nove varijable sesije preko globalnog niza $_SESSION.

PHP sesija

- PHP uključuje podršku za sesiju na način da se svakom korisniku pri pristupu stranicama sa uključenom sesijom dodijeli jedinstveni identifikator tzv. *session id* koji se ili pohrani u *cookiju* na klijentskoj strani ili se propagira kao dio URI-a.

- Funkcija session_start() ili kreira novu sesiju ili nastavlja postojeću na osnovu identifikatora sesije koji je proslijeđen u zahtjevu kao dio URI-ja ili *cookija*.

bool session_start (void)

- Korištenjem session_start() funkcije automatski se inicira GLOBALNI niz $_SESSION koji služi za pohranjivanje tj. preuzimanje informacija o korisniku.

PHP otvaranje sesije

- http://www.fesb.hr/~kiki/primjeri_internet_2/session_otvaranje.php

```
<?php     session_start();  ?>
<html><body>
<?
echo "Sadrzaj globalnog niza SESSION: ";
print_r($_SESSION);
echo "<p>";
echo "Identifikator sesije dohvaćen preko poziva funkcije session_id: ";
echo session_id();
echo "<p>";
echo "Identifikator sesije dohvaćen preko varijable PHPSESSID: ";
echo $_REQUEST['PHPSESSID'];
?>
</body></html>
```

PHP identifikator sesije

- Identifikator sesije generira server (tj. php modul).
- Funkcija session_id dohvaća ili postavlja identifikator sesije

    string session_id ([string $id])

- Ukoliko se navede parametar $id, a sesija je već otvorena zamijeni će se postojeći identifikator sesije sa vrijednošću iz parametra $id. U tom slučaju se funkcija treba pozvati prije session_start() funkcije. Ukoliko se sesija održava preko cookija to znači i slanje novog Set-Cookie zaglavlja klijentu sa novim identifikatorom sesije.
- Povratna vrijednost je prazan string ukoliko sesija nije otvorena ili identifikator otvorene sesije.

PHP sesija

http://pzi.fesb.hr/session_stranica.php

```
<?php
    session_start();
    $_SESSION['userid']="maja";
?>
<html>
<body>
<a href="provjeri_session.php"> Provjeri ga </a>
</body>
</html>
```

PHP sesija

```
<?php
    session_start();
    if($_SESSION['userid']=="maja")
    {
            echo "<html><body>";
            echo "Vaš username = bas_sam_pametan, <br> a password =
    ja_sam_genije<br><br>";
            echo "Session je OK";
    }
    else
    {
            echo "<html><body>";
            echo "Otidjite na stranicu za pocetak sessiona";
            echo "<a href=\"session_stranica.php\"> Otvori ga </a>";
    }
?>
</body></html>
```

http://pzi.fesb.hr/provjeri_session.php

PHP sesija

- Funkcija session_unregister poništava registraciju globalne varijable iz trenutne sesije.

  bool **session_unregister** (string $name)

- Funkcija session_destroy uništava sve podatke registrirane za sesiju.

  bool **session_destroy** (void)

- Međutim funkcija session_destroy ne uništava globalne varijable povezane sa sesijom niti poništava cookie. Da bi se sesija do kraja uništila treba se poništiti id sesije.

PHP sesija primjer

```
<?php
                session_start();
                if(empty($_SESSION['count']))
                        $_SESSION['count']=1;
                else
                        $_SESSION['count']++;
?>
<html> <body>
<p> Pristupljeno stranici <?php echo $_SESSION['count'];?>
    puta. </p>
<p><a href="ponisti.php"> Ponistavanje
    sesije</p></body></html>
```

http://pzi.fesb.hr/sesija_url.php

PHP sesija primjer

```
<?php
        session_start();
        session_unregister("count");
        session_destroy();
?>
<html>
<body>
<p> Ponistena sesija </p>
</body></html>
```

http://www.fesb.hr/~kiki/primjeri_internet_2/ponisti.php

Krađa sesije (*session hijacking*)

- Krađa sesije je postupak preuzimanja sesije korisnika nakon što je napadač uspješno dohvatio ili generirao jedinstveni identifikator sesije.
- Napadač može dohvatiti identifikator sesije npr. *snifanjem* paketa ili može pokušati provalu sesije slučajnim generiranjem identifikatora sesije (zato su identifikatori sesije obično dugi brojevi kako ih ne bi bilo lako slučajno pogoditi) ili može probiti mehanizam generiranja identifikatora sesije i na taj način preuzeti sesiju legitimnog korisnika dok ta sesija još uvijek traje.