Internet programiranje

PHP - Kolačići i sesije

Predavači:
Dr Dražen Drašković, docent
master inž. Jelica Cinović, asistent

HTTP ne čuva stanje

- Veb pregledač i veb server međusobno komuniciraju pomoću HTTP protokola koji ne čuva stanje između dve razmene podataka.
- Svaki HTTP zahtev koji veb pregledač šalje veb serveru nezavisan je od svih drugih zahteva.
- Ovakva komunikacija veb pregledača i veb servera pogodna je za aplikacije koje korisnicima omogućavaju da pretražuju ili pregledaju grupe povezanih dokumenata koristeći hiperlinkove.

Potreba za čuvanjem stanja

- U aplikacijama u kojima je potrebna složenija intervencija korisnika, mora se obezbediti čuvanje stanja pri prelasku sa jedne stranice aplikacije na drugu.
- *Primer*: Veb aplikacija prodavnice
  Korisnik dodaje stavke u korpu
  dok pretražuje ili pregleda neki katalog.
  Stanje korpe za kupovinu (stavke koje su sadržane u njoj) mora se negde čuvati.
  Kada korisnik zahteva stranicu za prikaz sadržaja korpe, mora se prikazati koje se stavke u njoj nalaze.

Čuvanje stanja bez upotrebe sesije

- U svakom HTTP zahtevu, među podacima koji se šalju metodom GET ili POST, prosleđivati i podatke koji predstavljaju trenutno stanje aplikacije.
- Nepotrebno povećanje saobraćaja na veb-u.
- Ako se podaci koji opisuju stanje prenose HTTP GET metodom (kao deo URL-a), korisnik može ručno da izmeni vrednosti koje se šalju zahtevom, a često nastaju i dugačke i nepregledne URL adrese.

6) Čuvanje stanja - kolačići i sesije

- Stanje aplikacije (vrednosti pojedinih promenljivih) mora se negde skladištiti između dva HTTP zahteva.
- Promenljive koje opisuju stanje mogu se čuvati na dva mesta:
  1. u klijentskom veb pregledaču
  2. na veb serveru

Pristup kolačićima u PHP skriptu

- Ako HTTP zahtev sadrži kolačiće, oni se smeštaju u superglobalnu promenljivu $_COOKIE (asocijativan niz, sa indeksom koji odgovara nazivu kolačića)

- Pristupanje kolačiću, čiji je naziv moj_kolacic, da bismo pročitali vrednost dobijamo pomoću $_COOKIE["moj_kolacic"]

Funkcija setcookie (1)

- Funkcija setcookie() generiše ispravno polje zaglavlja i ugrađuje ga u HTTP odgovor

- Funkcija setcookie() se poziva sa 6 argumenata, iako je samo prvi (ime kolačića) obavezan

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

Funkcija setcookie (2)

- **name**: naziv kolačića; obavezan parametar
- **value**: vrednost kolačića koja će se čuvati na računaru klijenta; ne čuvati poverljive informacije; vrednost kolačića čiji je naziv my_cookie u okviru skripta se dohvata sa $_COOKIE["my_cookie"]
- **expire**: vreme u timestamp formatu kada kolačić prestaje da važi; najčešće korišćeni oblici **time()+broj_sekundi** ili **mktime()**

Funkcija setcookie (3)

- **path**: putanja na serveru na kojoj će kolačić biti dostupan
  Primeri:
  - **' / '** - ceo domen
  - **'/www/admin/'** - u folderu www/admin i svim podfolderima,
  - **default path** - tekući direktorijum u kome se nalazi skript koji je kreirao kolačić

Funkcija setcookie (4)

- **domain**: domen u kome je kolačić dostupan

- Primeri:
  - **' '** - (prazan string) domen kome pripada server na kome se nalazi skript koji kreira kolačić
  - **'www.example.com'** - u navedenom domenu,
  - **'.bg.ac.rs'** - u svim poddomenima domena Univerziteta u Beogradu bg.ac.rs,

Funkcija setcookie (5)

- **secure** :
  Označava da kolačić treba da bude poslat samo putem sigurne HTTPS konekcije.

  Vrednosti TRUE ili 1 znače da se kolačić šalje samo kroz sigurne HTTPS konekcije.

  Default vrednost je FALSE ili 0 što znači da se kolačić šalje po svim vezama (i sigurnim i nesigurnim).

Funkcija setcookie - ograničenja

- Kao i ostali header-i, kolačići moraju biti poslati pre bilo koji naredbe izlaza u okviru skripta (ograničenje protokola).

- To znači da se poziv funkcije setcookie() prethodi bilo kom izlazu (npr. pozivu funkcije echo), uključujući i <html> i <head> tagove kao i bilo koje beline.

Funkcija setcookie - povratne vred.

- Ako postoji naredba izlaza koja prethodi funkciji setcookie(), funkcija se neće uspešno završiti i vratiće FALSE.

- Ako se funkcije setcookie uspešno izvrši, vratiće vrednost TRUE, što ne znači da je korisnik prihvatio kolačić.

Ograničenja za kolačiće (1)

- Kolačići se mogu koristiti u jednostavnim aplikacijama u kojima nije neophodno da se složeni podaci čuvaju između dva zahteva serveru.

- Broj i veličina kolačića su ograničeni:
  veb pregledač može da čuva samo poslednjih 20 kolačića koji su mu bili poslati iz određenog domena, a veličina svakog kolačića je ograničena na 4KB (ovo se vremenom menja i zavisi od verzije pregledača).

Ograničenja za kolačiće (2)

- Pitanje privatnosti korisnika i zaštite aplikacije u kojima se koriste kolačići.
- Neki korisnici isključuju podršku za rad sa kolačićima.

Upravljanje sesijama na vebu

- Podaci o tekućem stanju aplikacije čuvaju se na veb serveru tj. u srednjem sloju aplikacije
- Rešava se problem čuvanja promenljivih stanja koje zauzimaju više prostora i/ili većeg broja promenljivih stanja.
- Rešava se problem zaštite podataka sadržanih u promenljivama stanja od nenamernih ili namernih izmena koje bi korisnik mogao načiniti.

Sesija

- Sesija (engl. *session*) je jedan od načina označavanja i upravljanja skupom podataka o stanju, pomoću sesijskih promenljivih datog korisnika.

- Kada korisnik pošalje HTTP zahtev, srednji sloj aplikacije mora da obradi tekući zahtev vodeći računa o kontekstu (stanju) sesije.

Početak sesije

- Kada korisnik započne sesiju, klijentskom pregledaču šalje se identifikator sesije, najčešće u obliku kolačića, čija se vrednost ugrađuje u sve naredne zahteve koje veb pregledač upućuje serveru.

- Pomoću identifikatora sesije server identifikuje odgovarajuću sesiju pre nego što nastavi dalju obradu prispelog zahteva.

Identifikator sesije

- Kod kolačića se lokalno (u klijentskom veb pregledaču) skladište vrednosti svih promenljivih koje su neophodne za održavanje stanja i one se ugrađuju u svaki HTTP zahtev.

- Ako se koriste sesije **veb pregledač čuva i ugrađuje u svaki HTTP zahtev samo identifikator sesije, na osnovu koga se jednoznačno identifikuju sesijske promenljive njegove sesije.**

Problem zamrznutih sesija (1)

- U srednjem sloju se čuvaju zasebni podaci za svaku sesiju.
- Pitanje: Koliko dugo?
- Kada se sve odvija kako bi trebalo korisnik se sam odjavljuje iz aplikacije (recimo klikom na dugme "Kraj rada"), a skript koji se onda pokreće završava sesiju.
- Međutim, u ovo se ne smete pouzdati. Korisnici se često ne odjavljuju iz aplikacije na adekvatan način.

Problem zamrznutih sesija (2)

- Ako se korisnik ne odjavi adekvatno iz aplikacije, njegova sesija se neće završiti tj. njegove sesijske promenljive će se i dalje čuvati na serveru.
- Server nikada ne može da bude siguran da li se na drugoj strani veze još uvek nalazi korisnik (svaki HTTP zahtev je nezavisan od drugih zahteva).
- Zato server treba redovno da čisti stare, nezavršene (zamrznute) sesije u kojima se tokom određenog vremena nije ništa događalo.

Problem zamrznutih sesija (3)

- Problemi koje izazivaju zamrznute sesije:
  - troše resurse servera
  - bezbedonosni rizik
- Dužina intervala čekanja, pre nego što se neka zamrznuta sesija očisti, nije univerzalni parametar i zavisi od potreba aplikacije.

Odlike mehanizma za upravljanje sesijama na vebu - rezime

- Podaci koji čuvaju stanje aplikacije moraju se skladištiti na serveru.
- Svaki HTTP zahtev mora da sadrži identifikator koji serveru omogućava da obradi prispeli zahtev u kontekstu tekućeg stanja.
- Za sesije mora biti zadato vreme čekanja.
  U suprotnom, ako korisnik napusti veb lokaciju, nema drugog načina da se završi sesija.

Upravljanje sesijama u PHP-u

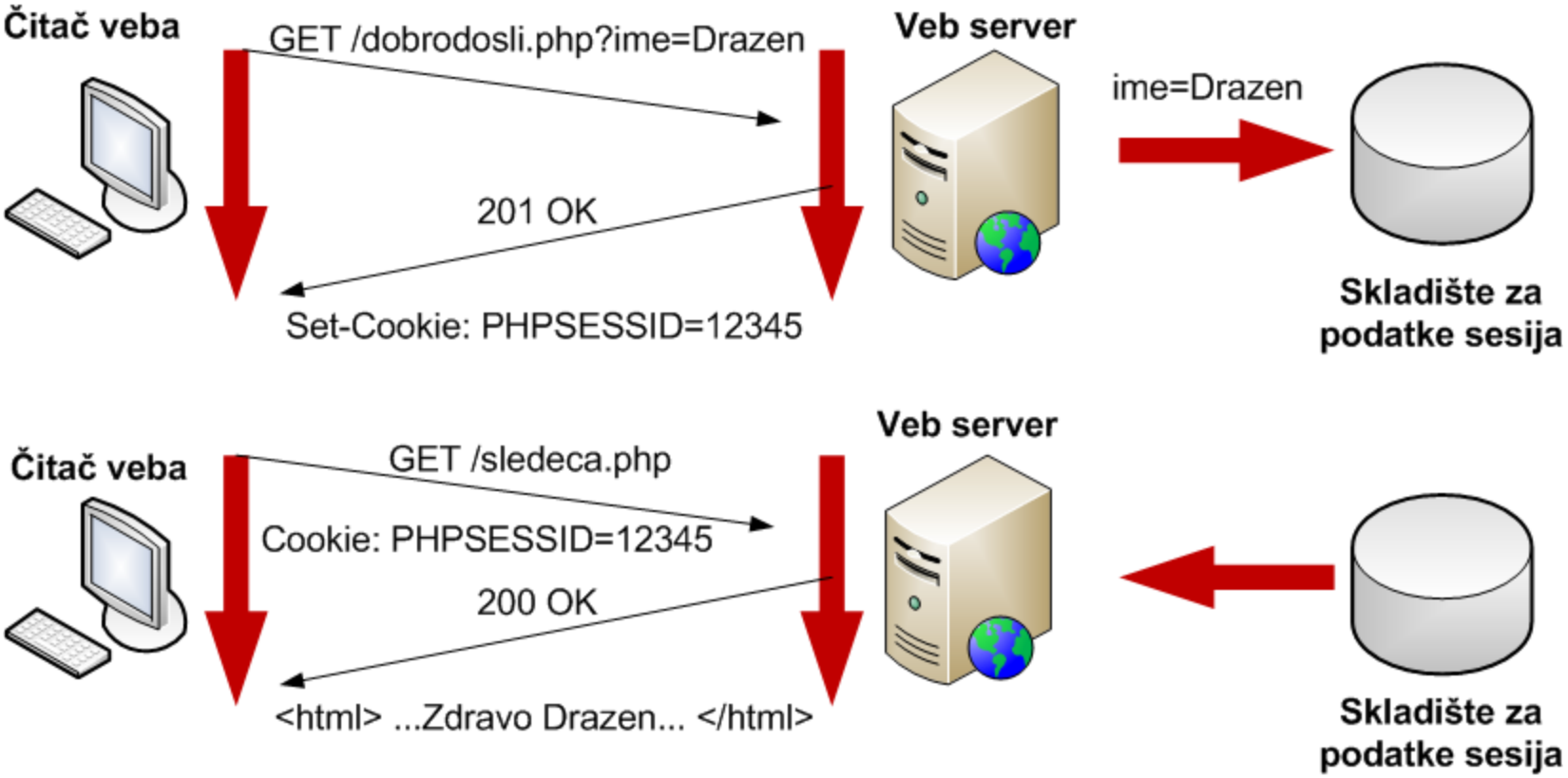
- PHP sadrži skup funkcija koje omogućavaju upravljanje sesijama, što olakšava razvoj aplikacija koje koriste sesije.
- Karakteristične situacije korišćenja sesija:
  - započinjanje nove sesije
  - korišćenje sesijskih promenljivih koje čuvaju stanje aplikacije
  - okončavanje sesije

Mehanizam sesija (1)

- Kada korisnik prvi put pokrene aplikaciju koja upravlja stanjem, tako što zahteva veb stranicu koja započinje sesiju, PHP radi sledeće stvari:
  - generiše identifikator (ID) sesije i
  - formira datoteku za smeštaj vrednosti promenljivih koje se odnose na novu sesiju

- U odgovor, koji skript generiše, PHP ugrađuje kolačić koji sadrži ID sesije. Klijentski veb pregledač potom skladišti kolačić i umeće njegovu vrednost u sve naredne zahteve koje šalje serveru.

Mehanizam sesije (2)



Razmena podataka između pregledača i servera,
kada se prvi zahtev pošalje aplikaciji koja čuva stanje

Gde čuvati sesijske promenljive?

- U standardnoj konfiguraciji, PHP upisuje vrednosti sesijskih promenljivih u datoteke na disku, što je pogodno za većinu aplikacija.

- Rešenje koje omogućava veći stepen prilagodljivosti, zasnovano je na upotrebu MySQL-ove baze podataka za čuvanje podataka o sesijama (samo za jako zahtevne aplikacije gde je broj i veličina sesijskih promenljivih izražena).

Rad sa sesijama

- Započinjanje sesije
- Rad sa sesijskim promenljivama
- Okončavanje sesije

Započinjanje sesije (1)

- Za započinjanje nove sesije ili učitavanje postojeće sesije koristi se funkcija session_start():

```
bool session_start ( )
```

- Ova funkcija kreira novu sesiju i time omogućava pristup superglobalnom nizu $_SESSION ili učitava postojeću sesiju koju identifikuje na osnovu ID sesije koji je prosleđen u HTTP zahtevu, putem GET, POST ili cookie.
- Ova funkcija uvek vraća vrednost TRUE.

Započinjanje sesije (2)

- Ako se koriste sesije zasnovane na kolačićima (gde se identifikator sesije prenosi kao kolačić), funkcija `session_start()` mora biti pozvana pre nego što se bilo koji izlaz uputi veb pregledaču.

Započinjanje sesije (3)

- Dva slučaja:

1. u HTTP zahtevu nema identifikatora sesije

2. u HTTP zahtevu ima identifikatora sesije
    1. Odgovarajuća datoteka sesije postoji
    2. Odgovarajuća datoteka sesije ne postoji

HTTP zahtev nema identifikator sesije

- Generiše se identifikator nove sesije.
- U standardnoj konfiguraciji, u HTTP odgovor se automatski umeće zaglavlje Set-Cookie koji na klijentskom računaru formira kolačić čije je ime PHPSESSID, a vrednost ID sesije (grupa od 32 nasumično generisane heksadecimalne cifre)
- PHP formira datoteku sesije.
  U standardnoj konfiguraciji, datoteke sesija se smeštaju u direktorijum /tmp, a ime datoteke se sastoji od ID sesije i prefiksa sess_

HTTP zahtev ima identifikator sesije

- PHP će pokušati da nađe odgovarajuću datoteku sesije kojoj odgovara dati ID sesije.
- **Ako datoteka sesije postoji**,
  učitava se postojeća sesija, a promenljivama te sesije se može pristupati sa:
  `$_SESSION["naziv_sesijske_promenljive"]`
  (mogući su i čitanje i upis)
- **Ako datoteka sesije ne postoji**
  (usled sakupljanja smeća tj. brisanja zamrznutih sesija), PHP formira praznu datoteku.

Upotreba sesijskih promenljivih (1)

- Ispitivanje da li postoji određena sesijska promenljiva se vrši pomoću funkcije isset() na sledeći način:

```
isset($_SESSION["naziv_sesijske_prom"])
```

- Kreiranje nove sesijske promenljive ili dodela nove vrednosti sesijskoj promenljivoj se vrši pomoću naredbe:

```
$_SESSION["naziv_sesijske_prom"]=izraz
```

Upotreba sesijskih promenljivih (2)

- Brisanje postojeće sesijske promenljive se vrši pomoću funkcije unset():

```
unset($_SESSION["naziv_ses_prom"])
```

Okončavanje sesije (1)

- Sesija treba da se okonča kada se korisnik odjavi iz aplikacije (izborom opcije "Kraj rada").

  Okončavanje sesije vrši se pozivom funkcije `session_destroy()`

Okončavanje sesije (2)

`bool session_destroy ( )`

- Funkcija `session_destroy()` uklanja datoteku sesije iz sistema, ali ne uklanja kolačić $PHPSESSID.

- Funkcija vraća vrednost TRUE, ako je sesija uspešno uklonjena, inače vraća vrednost FALSE.

Upravljanje sesijama bez upotrebe kolačića (1)

- Jedna od izmena koja se može uneti u standardni PHP-ov način upravljanja sesijama jeste da se u zahtev koji se šalje metodom GET ili POST ugradi i vrednost ID sesije kao atribut da bi se izbegla potreba za formiranje kolačića.

- Razlog:
  korisnik može svoj veb pregledač da podesi tako da zabranjuje upotrebu kolačića.

Upravljanje sesijama bez upotrebe kolačića (2)

- Posledice zabranjivanja upotrebe kolačića:
  - aplikacije koje koriste sesije, kod kojih se ID sesije prenosi putem kolačića, neće raditi
  - svaki put kada se zahteva veb stranica, u odgovarajućem folderu se kreira nova sesija

- Kako napisati aplikacije koje će korektno raditi i kada korisnici u svojim veb pregledačima zabranjuju upotrebu kolačića
  => ID sesije se ne šalje u obliku kolačića, nego se prosleđuje unutar zahteva metodom GET ili POST!

Upravljanje sesijama bez upotrebe kolačića (3)

- Funkcija `session_start()` može da koristi promenljivu $PHPSESSID bez obzira na to da li je njena vrednost prosleđena u zahtevu metodom GET ili POST.

- Uglavnom se koristi metoda GET.

Isključivanje kolačića

- PHP-ov sistem za upravljanje sesijama se može podesiti tako da ne formira kolačić PHPSESSID.

- session.use_cookies = 0 (default 1) (php.ini)

Prosleđivanje ID sesije u obliku GET promenljive (1)

- Kada se pošalje prvi zahtev za izvršavanje skripta otvara se nova sesija i formira odgovarajuća datoteka čije bi ime bilo nešto poput: sess_be2202...AE80
- Svi naredni zahtevi trebalo bi da sadrže atribut PHPSESSID i izgledali bi:
  http://localhost/page.php?
  PHPSESSID=be2202...AE80

Prosleđivanje ID sesije u obliku GET promenljive (2)

- Kako navedeno postići u kodu?
- Skriptovi koji generišu hiperlinkove ka drugim veb stranicama koje koriste sesije moraju da ugrađuju u URL-ove GET atribut čiji je naziv PHPSESSID
  <a href="abc.php?PHPSESSID=<?=session_id() ?>"
  ili kraće
  <a href="abc.php?<?=SID ?>“
- PHP podešava konstantu SID u oblik pogodan za upotrebu u URL-u kao upitni deo, da bi se olakšalo pisanje URL-ova koji upućuju na skriptove u kojima se koriste sesije.

Prosleđivanje ID sesije u obliku GET promenljive (3)

- Ako sesija nije kreirana pomoću funkcije `session_start()`, PHP podešava vrednost konstante SID na praznu znakovnu vrednost.

- Ako je sesija kreirana SID će biti oblika PHPSESSID=be2202...AE80

Prosleđivanje ID sesije u obliku GET promenljive (4)

- Umesto da se piše kod koji ugrađuje ID sesije u URL, u PHP-u postoji opcija URL *rewrite,* koja automatski podešava referentne URL-ove tako da sadrže ID sesije kao GET atribut.
- Da bi se ova mogućnost aktivirala (php.ini)
  session.use_trans_sid = 1 (default 0)
- Pošto se uključi opcija URL *rewrite,* PHP analizira HTML kod koji skriptovi generišu i automatski dopunjava URL-ove u njemu tako da sadrže atribut PHPSESSID u upitnom delu.

Prosleđivanje ID sesije u obliku GET promenljive (5)

- Mane opcije URL rewrite:
  - Potrebna je dodatna obrada zbog analiziranja svake generisane stranice
  - Bezbedonosni problemi
    (URL sa identifikatorom sesije se vidi u URL-u, može biti sačuvan u History ili Bookmarks na nekom public računaru,...) => LOŠE!!!

Sakupljanje smeća (1)

- Treba praviti aplikacije koje korisniku pružaju mogućnost da sam zatvori sesiju, tako što će u skriptu pozvati funkciju session_destroy()
- Ne može se garantovati da će se korisnik uvek odjaviti iz aplikacije tako što će zahtevati odgovarajući skript
- U PHP-ov mehanizam upravljanja sesijama ugrađen je mehanizam za sakupljanje smeća, koji obezbeđuje da se datoteke neaktivnih (zamrznutih) sesija posle izvesnog vremena brišu

Sakupljanje smeća (2)

- Neophodnost sakupljanja smeća:
  - sprečava se da se folder prepuni datotekama sesija, što slabi performanse sistema
  - smanjuje se rizik da neko ko nasumično pogađa ID sesija ukrade neku staru sesiju koja se više ne koristi

- Postoje dva parametra (definisana u datoteci php.ini) koja upravljaju sakupljanjem smeća:
  - session.gc_maxlifetime i
  - session.gc_probability

Sakupljanje smeća (3)

- Tokom postupka sakupljanja smeća, ispituju se sve sesije i briše se svaka sesija kojoj niko nije pristupio tokom vremena određenog parametrom gc_maxlifetime (default vrednost 1440 sec)

- Parametar gc_probability određuje procenat verovatnoće sakupljanja smeća (100% - svaki put kada se pozove funkcija session_start(), 1% - sa verovatnoćom 0.01 svaki put kada se pozove funkcija session_start())

Sakupljanje smeća (4)

- Postupak sakupljanja smeća može poprilično da optereti server, naročito kod veb aplikacija sa velikim brojem korisnika (mora se ispitati datum poslednjeg ažuriranja svake sesije)

- Ako se parametar gc_probability podesi suviše visoko nepotrebno se opterećuje server, a ako se podesi suviše nisko javlja se problem zamrznutih sesija i problema koje one uzrokuju

Sakupljanje smeća (5)

- Generalno, ne postoji pravilo za podešavanje parametara gc_maxlifetime i gc_probability
- Vrednosti ovih parametara trebalo bi da budu odabrane tako da obezbeđuju ravnotežu između potreba aplikacije i performansi sistema

Podešavanje sistema za upravljanje sesijama (1)

- Podešavanje sistema za upravljanje sesijama vrši se postavljanjem vrednosti odgovarajućim parametrima koje PHP koristi za upravljanje sesijama u datoteci **php.ini**

Podešavanje sistema za upravljanje sesijama (2)

- `session.auto_start`
  Određuje da li se sesija započinje automatski, na svakoj stranici. Default: 0 (isključeno)
- `session.cookie_domain`
  Ime domena koje se zadaje u sesijskom kolačiću. Default: ništa
- `session.cookie_lifetime`
  Životni vek kolačića na klijentskom računaru na kome se nalazi identifikator sesije. Default: 0 (dok se ne zatvori veb čitač)

Podešavanje sistema za upravljanje sesijama (3)

- `session.cookie_path`
  Putanja koja se zadaje u kolačiću sesije.
  Default: /
- `session.name`
  Ime sesije koje se koristi i kao ime kolačića na klijentskom računaru.
- `session.use_cookies`
  Određuje da li sesije koriste kolačiće na klijentskoj strani (0-ne koriste, 1-koriste).
  Default: 1 (sesije koriste kolačiće)

Podešavanje sistema za upravljanje sesijama (4)

- `session.save_handler`
  Definiše mesto gde se upisuju podaci sesije.
  To može biti i BP, ali u tom slučaju morate napisati vlastite funkcije.
  Default: files (podaci sesije se upisuju u fajlovima)

- `session.save_path`
  Putanja datoteke u koju se upisuju podaci sesije (ako je session.save_handler=files).
  Default: /tmp

Primer sesije (1)

Stranica page1.php:

```
<?php
  // pocetak sesije
  session_start();
  // postavljanje 3 vrednosti u sesiju
  $_SESSION['color']='red';
  $_SESSION['size']='small';
  $_SESSION['shape']='round';
  print “Uneti podaci u sesiju";
?>
```

Gde staviti session_start() ?

- `session_start()` mora biti u zaglavlju i ne treba ništa da šaljete pregledaču pre toga.
- Najbolje bi bilo da sesiju otpočnete odmah posle `<?php` da ne bi nastali potencijalni problemi.

Primer sesije (2)

Stranica page2.php:

```
<?php
  // pocetak sesije
  session_start();
  // prikazujemo sta je sacuvano u sesiji
  echo “Boja je ".$_SESSION['color'];
  echo “Velicina je ".$_SESSION['size'];
  echo “Oblik je ".$_SESSION['shape'];
?>
```

Primer sesije (3)

Sve vrednosti sesije se čuvaju u superglobalnoj promenljivoj - nizu $ _SESSION, kome smo ovde pristupili.

Drugi način da se ovo prikaže je:

```
<?php
   session_start();
   Print_r($_SESSION);
?>
```

Funkcija Print_r()

- Vraća elemente niza koji prosleđujemo kao argument funkcije: **Print_r ($your_array)**

```
<?php
   $imena = array ('a' => 'Aleksandra', 'b' =>
   'Branislav', 'd' => array ('Drasko', 'Dusan'));
   print_r ($imena);  ?>
```

Rezultat je:

```
Array
   (
   [a] => Aleksandra
   [b] => Branislav
   [c] => Array
         (
         [0] => Drasko
         [1] => Dusan
         )
   )
```

Primer sesije (4)

- U okviru sesije je moguće sačuvati i niz podataka:

```
<?php
   session_start();
   // pravimo niz
   $boje=array('crvena', 'zuta', 'plava');
   // dodavanje u sesiju
   $_SESSION['boja']=$boje;
   $_SESSION['size']='small';
   $_SESSION['shape']='round';
   print "Uneti podaci u sesiju";
?>
```

Primer sesije (5)

- U okviru sesije je moguće sačuvati i niz podataka:

```
<?php
   // otvaramo sesiju radi modifikovanja sesije
   session_start();
   // promena varijable u sesiji
   $_SESSION['size']='large';

   //uklanjanje varijable shape iz sesije
   unset($_SESSION['shape']);

   // uklanjanje svih varijabli iz sesije, ali ne
   unistavamo sesiju
   session_unset();

   session_destroy(); // unistavanje sesije
?>
```

PHP - Kolačići i sesije

Hvala!