Rinktiniai Java skyriai

Pratybos #7

2007-12-06

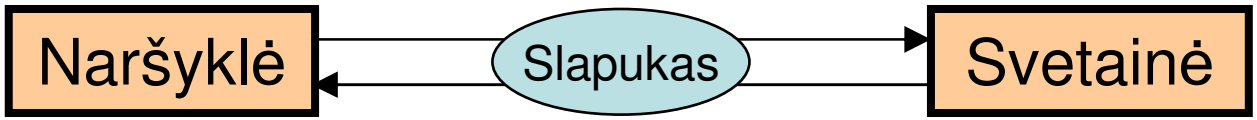
Slapukai, sesijos, vartotojai ir rolės

Slapukai

Slapukai

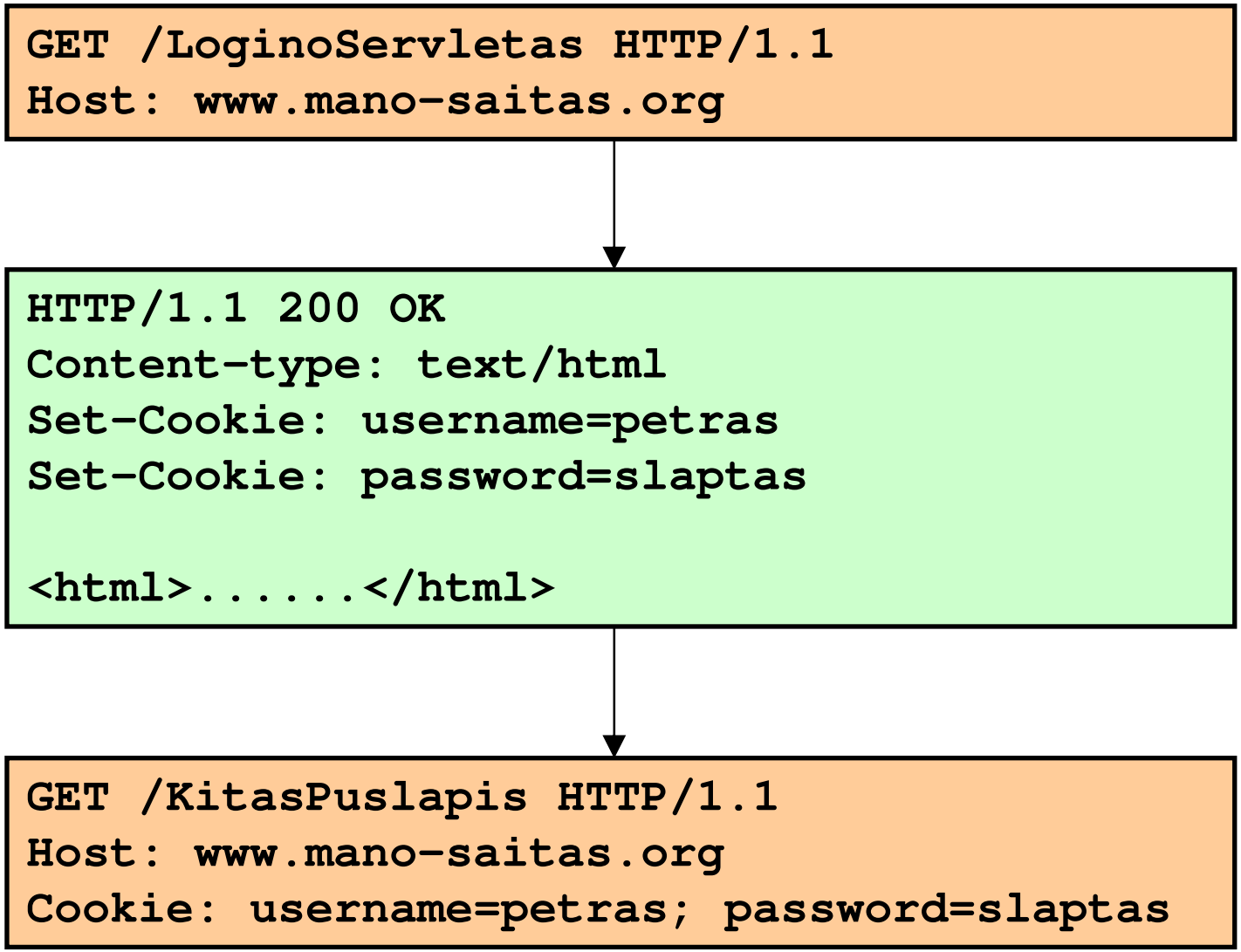
- Slapukas = HTTP cookie = Cookie
  - Pavadinimą “Cookie” sugalvojo Netscape Communications
  - Kodėl “Cookie”? Nes gerai skamba
- Informacinės visuomenės plėtros komitetas siūlo naudoti terminą “slapukas”
  - kuris skamba ne ką blogiau, todėl taip ir darysim

Slapukai



- Slapukas – tai mažas informacijos paketas, kurį interneto svetainė siunčia naršyklei HTTP atsakyme
  - naršyklė išsaugo slapuką kietajame diske arba RAM'e
  - naršyklė persiunčia slapuką atgal
    - kiekvienoje sekančioje HTTP užklausoje į tą pačią svetainę
  - svetainė vienoje užklausoje gali siųsti daugiau nei vieną slapuką
  - kiekviena svetainė mato tik savo slapukus
  - svetainė gali trinti slapukus, keisti jų reikšmes, pridėti naujų slapukų
    - naršyklė juos tik saugo ir persiunčia

Apsikeitimas slapukais



Apsikeitimas slapukais

- Response header'is "Set-Cookie" siunčiamas tik tuomet, kai:
  - kuriamas naujas slapukas
  - keičiama esamo slapuko reikšmė
  - keičiamas esamo slapuko galiojimo laikas
- Request header'is "Cookie" siunčiamas su kiekviena užklausa
  - tol, kol baigiasi slapuko galiojimo laikas

Sesijos slapukai

- Slapukai būna dviejų rūšių:
  - sesijos slapukai (Session cookies)
  - pastovūs slapukai (Persistent cookies)
- Sesijos slapukai paprastai laikomi naršyklės RAM'e
  - todėl jie išsitrina, vos tik vartotojas uždaro naršyklę

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value

<html>......</html>
```

Pastovūs slapukai

- Pastovūs slapukai paprastai saugomi vartotojo kietajame diske
  - tokie slapukai sukuriami header'yje Set-Cookie nurodant slapuko galiojimo laiką (atributas "expires")
  - kai galiojimo laikas baigiasi, naršyklė ištrina slapuką iš vartotojo kompiuterio

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value; expires=Fri, 31-Dec-2010 23:59:59 GMT;

<html>......</html>
```

Slapukų kontekstas

- Kuriant ar keičiant slapuką, paprastai be “expires” atributo dar siunčiami “domain” ir “path” atributai, kurie nurodo, kokiame kontekste galioja slapukas
  - “domain” nurodo domeno vardą
  - “path” nurodo virtualų kelią tame domene
- Tokiu atveju naršyklė persiunčia tik tuos slapukus, kurių domenas ir kelias sutampa su HTTP užklausos domenu ir keliu

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value; expires=...; path=/members; domain=.test.com

<html>......</html>
```

```
GET /members/InternalPage HTTP/1.1
Host: www.test.com
Cookie: name=value
```

Slapukai su Java

- Sesijos slapuko kūrimas

```
Cookie cookie = new Cookie("name", "value");
cookie.setPath("/members");  // pagal nutylėjimą "/"
response.addCookie(cookie);
```

- Pastovaus slapuko kūrimas

```
Cookie cookie = new Cookie("name", "value");
cookie.setPath("/members");  // pagal nutylėjimą "/"
cookie.setMaxAge(7*24*60*60);  // 604800 sekundžių = 1 savaitė
response.addCookie(cookie);
```

- Slapuko trynimas

```
Cookie cookie = new Cookie("name", "value");
cookie.setPath("/members");  // pagal nutylėjimą "/"
cookie.setMaxAge(0);  // galiojimas baigiasi dabar
response.addCookie(cookie);
```

Slapukai su Java

- Slapuko nuskaitymas iš HTTP Request

```
Cookie[] cookies = request.getCookies();
String username = null;

if (cookies != null) {
    foreach (Cookie c : cookies) {
        if (c.getName().equals("username")) {
            username = c.getValue();
            break;
        }
    }
}

if (username != null) {
    out.println("Hello user " + username);
}
else {
    response.sendRedirect("/LoginPage");
}
```

Slapukai su Java

- Slapuko varde ir reikšmėje neturėtų būti tarpų ir tokių simbolių:

  [ ] ( ) = , “ / ? @ : ; %

- Todėl jeigu slapuko reikšmė iš anksto nežinoma, geriau yra ją užkoduoti su URLEncoder klase:

```
new Cookie(“name”, URLEncoder.encode(value, "UTF-8"));
```

- Atitinkamai nuskaičius slapuko reikšmę, ją reikėtų iškoduoti su URLDecoder klase:

```
String value = URLDecoder.decode(cookie.getValue(), "UTF-8");
```

HTTP sesija

HTTP trūkumai

- Problemos su HTTP protokolu:
  - Jis nesaugo būsenos tarp užklausų
  - Nėra pastovios jungties tarp kliento ir serverio
  - Serveris “pamiršta” apie klientą vos baigęs apdoroti jo užklausą
- Tai iš tiesų yra viena ir ta pati problema:
  - HTTP protocol is stateless
- Kuo blogai, kad nesaugoma būsena?
  - Neįmanoma realizuoti vartotojų autorizavimo, “prekių krepšelio” ir pan.

HTTP sesija

- HTTP sesija yra būsenos tarp užklausų palaikymas
- HTTP sesijos implementacijų yra daug, ir nei viena iš jų nepriklauso HTTP standartui
  - bet visos naudoja tą patį principą

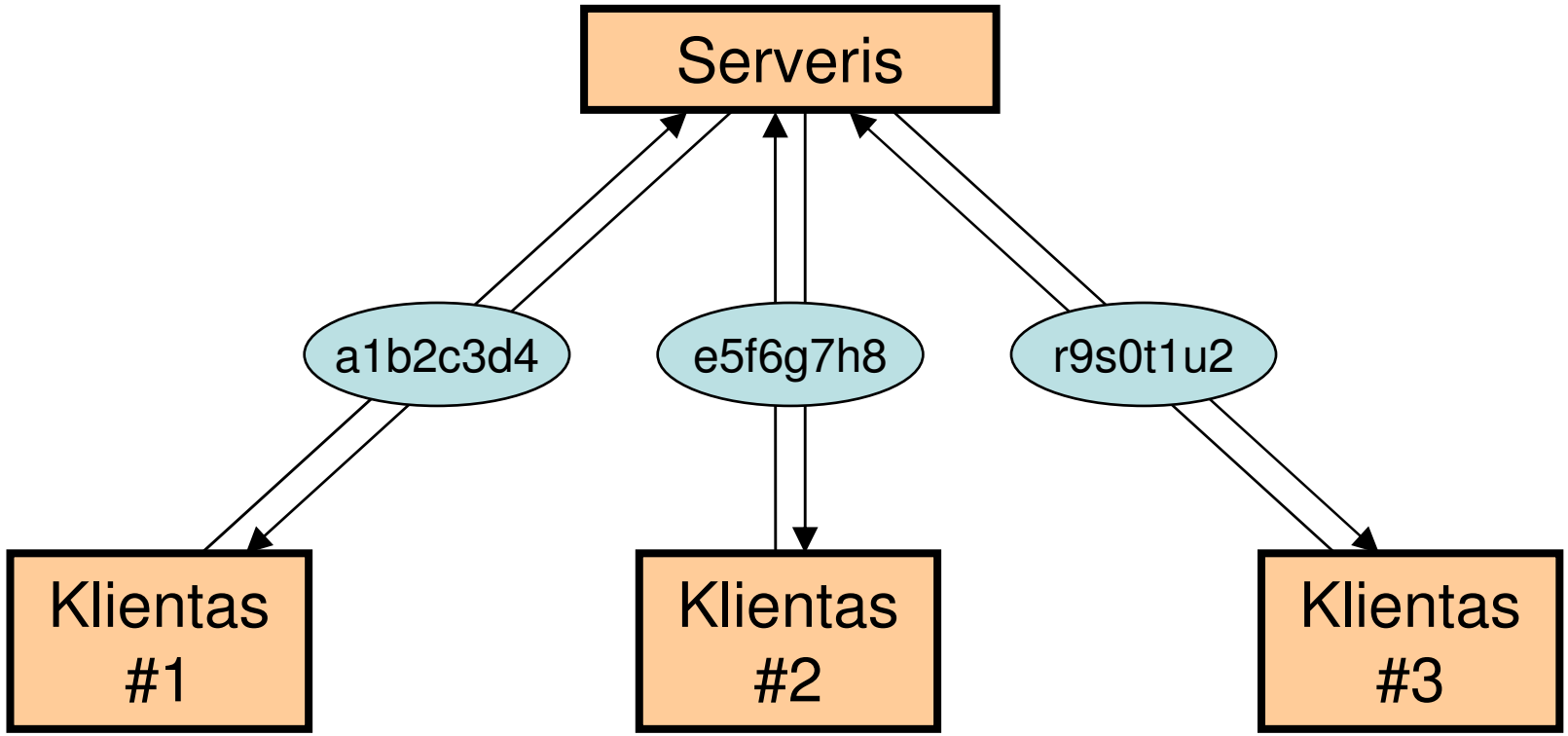
Sesijos identifikatorius

Idėjos esmė - sesijos identifikatorius (sessionID)

- Tai unikalus string'as, kuris sukuriamas tuo metu, kai prie serverio jungiasi naujas klientas (t.y. klientas, dar neturintis sessionID)
- Tuomet serveris perduoda sessionID klientui kiekviename HTTP atsakyme, o klientas perduoda sessionID serveriui kiekvienoje HTTP užklausoje
- Tokiu būdu serveris tiksliai žino, su kuriuo klientu bendrauja – taigi palaikomas pastovus ryšys

Sesijos identifikatorius

Kiekvienam klientui – po unikalų sesijos ID



Sesijos identifikatorius

sessionID perdavimo metodai:

- per URL'us

```
<a href="/SomePage;sessionID=a1b2c3d4">Link</a>
```

- per HTML formas

```
<form action="/SomePage" method="post">
  <input type="hidden" name="sessionID" value="a1b2c3d4"/>
</form>
```

- per slapukus
  - Set-Cookie: sessionID=a1b2c3d4 *(HTTP Response)*
  - Cookie: sessionID=a1b2c3d4 *(HTTP Request)*

Sesijos identifikatorius

Jei serverio pusėje turėtume po unikalų sessionID kiekvienam klientui, galėtume jį susieti su globalia (web aplikacijos lygio) HashMap kolekcija, kurioje laikytume tik tam klientui prieinamus objektus, - pavyzdžiui, kliento “prekių krepšelį”

Java sesija

- Aptarėme, kaip galima patiems rankiniu būdu suprogramuoti sesijos palaikymą
- Tačiau Java servletai jau turi standartinę sesijos implementaciją:
  - klasė HttpSession
  - sesijos ID perduodamas dviem būdais:
    - slapuke “JSESSIONID”, kuris išsitrina uždarius naršyklę
    - JSP puslapiuose visi <c:url> tag’ai į URL’ą automatiškai įterpia `;jsessionid=...` (tai padeda tuo atveju, kai kliento naršyklė blokuoja slapukus)

Java sesija

- Prekių krepšelio nuskaitymas/įrašymas Java sesijoje

```
HttpSession session = request.getSession();

ShoppingCart cart = (ShoppingCart) session.getAttribute("cart");
if (cart == null) {
    cart = new ShoppingCart();
    session.setAttribute("cart", cart);
}

String itemID = request.getParameter("itemID");
if (itemID != null) {
    cart.addItem(item);
}
```

- PASTABA. Visų klientų sesijų duomenys (prekių krepšeliai) saugomi serveryje. Tarp serverio ir kliento siuntinėjamas tik sesijos ID !

Java sesija

- Prekių krepšelio trynimas iš sesijos

```
session.removeAttribute("cart");
```

- Sesijos užbaigimas, visų jos objektų naikinimas (paprastai atliekamas tuo metu, kai vartotojas atsijungia)

```
session.invalidate();
```

- Taip pat sesija yra naikinama automatiškai, kai baigiasi jos galiojimo laikas, nurodytas web.xml

```
<session-config>
    <session-timeout>180</session-timeout>    <!-- 2 valandos -->
</session-config>
```

Vartotojai ir rolės

Terminologija

- Autentifikacija – vartotojo tapatybės nustatymas
- Autorizacija
  - teisių atlikti tam tikrus veiksmus suteikimas vartotojui
  - tikrinimas, ar vartotojas neviršija savo įgaliojimų
- Autorizacija visada seka po autentifikacijos
- Rolė – abstrakti galimų atlikti veiksmų aibė
  - suteikiama vienam arba keliems vartotojams
  - vienas vartotojas gali turėti daugiau nei vieną rolę
- Vartotojo teisės – vartotojo rolių sąjunga

Terminologija

- Apsauga (security) – rūpestis, kad tik autorizuoti vartotojai galėtų prieiti prie atitinkamų serverio resursų
- Deklaratyvi apsauga – kai apsauga konfigūruojama failuose (ppr. XML), ją realizuoja Tomcat’as ar kitas web aplikacijų serveris
- Programinė apsauga – kai apsauga rūpinasi programuotojas