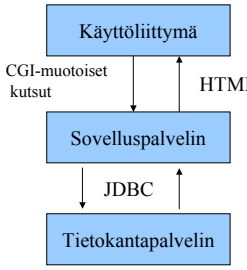
Servletit tietokantasovelluksen toteutuksessa



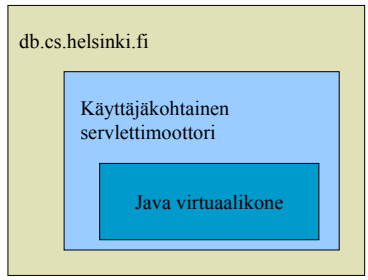
Selain (HTML + JavaScript)

Servletit = java ohjelmia

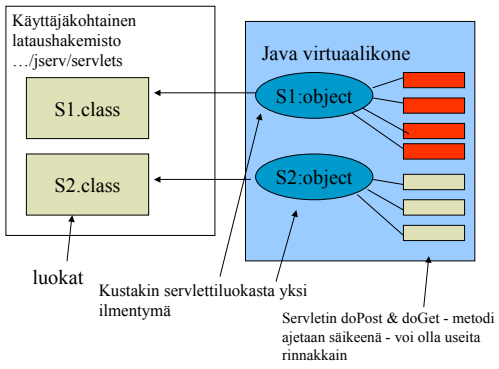
Oracle (bodbacka.cs.helsinki.fi)
PostgreSQL (db.cs.helsinki.fi)

Servletit

Sovelluspalvelimessa:



Servletit



Servletin kutsuminen

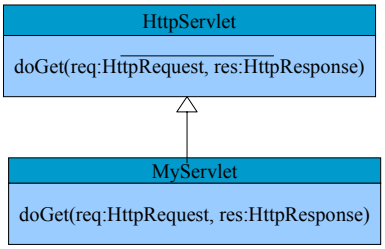
- HTML-sivulta servlettiä kutsutaan linkin avulla tai määrittelemällä servletti lomakkeen käsittelijäksi
- Linkin muoto:

<A href=
"http://db.cs.helsinki.fi/s/account/servlet_name?param1=jotain&param2=jotain2&…</A>

huom: Alkuosa 'http://db.cs.helsinki.fi/s/account/' on tarpeen vain ensimmäisen servletin käynnistykseen, muut kannattaa käynnistää suhteellisella viittauksella (= vain servlettiluokan nimi ja tarvittaessa parametrit)

Servletin yleisrakenne

Kaikki servletit perivät luokan HttpServlet



doGet ja doPost käsittelevät kutsun. Käsittelijä riippu kutsutavasta
Kummankin olisi syytä toimia samoin

Parametriarvot servletin käyttöön

- Kutsuparametrien arvo(t) saadaan ohjelman käyttöön luokan HttpRequest palvelulla
  - getParameter(kutsuparametrin_nimi) tai
    - palauttaa yhden merkkijonon
  - getParameterValues(kutsuparametrin_nimi)
    - palauttaa merkkijonotaulukon
  - Parametrin arvo haetaan samalla nimellä, joka parametrille on annettu joko lomakkeen input-kentässä tai linkin yhteydessä

Tulostus

- Tulostuksen ohjaamiseksi selaimelle pyydetään res-parametrilta tulostusjono ja alustetaan se

```
res.setContentType("text/html");
PrintWriter pw = res.getWriter();

Tämän jälkeen kirjoitetaan HTML:ää
  jonoon pw
```

Säieturvallisuus

- doGet ja doPost-metodia ajetaan säikeinä, tällöin oliomuuttujien arvon asetus saattaa sotkea toisen rinnakkaisen säikeen toiminnan. Oliomuuttujien käsittely pitäisikin synkronoida.
- Yksinkertaisin tapa säieturvallisuuden takaamiseksi on käyttää vain palvelumetodien sisäisiä muuttujia.

Sovelluksen jako servleteiksi

- Yksinkertaisinta ja selkeintä on toteuttaa jokaista palvelupyyntöä varten oma servletti.
- Jos useampi servletti tuottaa samanrakenteisen sivun, voi sivun tuottamisen määritellä jonkin apuluokan tehtäväksi, jota usea servletti käyttää hyväkseen – tai määritellä perintäpolulle väliluokka oman servletin ja HttpServlet:n väliin
- Sama servletti voi toki käsitellä useitakin parametrein eroteltuja pyyntöjä, mutta jokaisella pitäisi olla vain yksi selkeä tehtävä.

Istunnot

- Servletit mahdollistavat istuntokohtaisen tiedon säilytyksen Session olioissa. Näiden käyttö harjoitustyössä ei ole välttämätöntä tai edes tarpeen vaan istuntokohtainen tieto voidaan kierrättää piilokenttinä selaimen kautta tai tallentaa tietokantaan ja kierrättää istuntotunnusta selaimen kautta.
- Jos Session olioita käytetään, niitä pitää käyttää oikein - eli uusi saman koneen käyttäjä ei saa saada edellisen istuntotietoja

Tietokanta servleteissä

- Tietokanta servleteissä, yksinkertainen harjoitustöihin sopiva malli:
  - Palvelumetodin alussa avataan tietokantayhteys
  - Tehdään tietokantakäsittely
  - Suljetaan tietokantayhteys ennen palvelu-metodista poistumista
- Tämä malli takaa, ettei tietokantayhteyksiä jää palvelimessa auki (jatkuvasti avoinna pidettävien tietokantayhteyksien käyttö on harjoitustöissä kielletty)

Tietokantayhteyden käynnistys

```
public Connection createConnection(
         String driverName,
         String dataBase,
         String  user, String password) {
   try {
       Class.forName(driverName);
   } catch (ClassNotFoundException ex)
       return null;
   }
   Connection conn= null;
   try {conn=DriverManager.getConnection(dataBase,
        user, password);
   } catch (SqlException sex) {
        conn=null;
   }
   return conn;
 }
```

- Oracle bodbacka

```
conn = createConnection(
"oracle.jdbc.OracleDriver",
"jdbc:oracle:thin:@bodbacka.cs.helsinki.fi:1521:test",
user,password);
```

- PostgreSQL

```
conn = createConnection("org.postgresql.Driver",
"jdbc:postgresql://localhost:portnumber/username",username,pas
   sword);
```

Tietokannan käsittely

- Oracle-kannan tunnusta pyydetään ryhmän ohjaajalta. Tietokantatunnus tulee olemaan sama kuin käyttäjän unix-tunnus.
- Postgresql ympäristön voi luoda itse komennolla *wanna-postgres* - komento antaa lisäohjeita

Tietokannan käsittely

- Create table -lauseet kannattaa koota yhteen tiedostoon.
- Viitattavat taulut perustettava ennen viittaavia.
- Insert lauseet testidataa varten myös omiin tiedostoihin
- Taulu hävitetään drop table -lauseella
- Viiteavain voi estää hävityksen
-

Servlettiympäristön pystytys ja alasajo (jserv)

- Servlettiympäristön pystytys (jserv):
  - wanna-servlet -komento pystyttää ympäristön
  - lataushakemisto ../jserv/servlets/
  - ajoaikainen CLASSPATH asetetaan automaattisesti tiedoston ../jserv/etc/environment perusteella
  - käännösaikaisesti CLASSPATHin saa asetettua komennolla setup servlet
  - servlettiympäristö pystyyn komennolla start-servlet
    - oletusarvoisesti pystyssä 10h, aikaa voi kasvataa
  - alasajo komennolla stop-servlet

Servlettiympäristön pystytys ja alasajo (tomcat)

- Servlettiympäristön pystytys (tomcat):
  - wanna-tomcat -komento pystyttää ympäristön
  - lataushakemisto tomcat/webapps/WEB-INF/classes/
  - konfigurointitiedosto tomcat/webapps/WEB-INF/web.xml
  - ajoaikainen CLASSPATH asetetaan automaattisesti perusteella
  - käännösaikaisesti CLASSPATHin saa asetettua komennolla setup tomcat
  - servlettiympäristö pystyyn komennolla start-tomcat
    - oletusarvoisesti pystyssä 10h, aikaa voi kasvataa
  - alasajo komennolla stop-tomcat