

Lohtu-projekti

Ylläpitäjän dokumentti

Versiohistoria:

1.0	12.5.2003	Ensimmäinen versio	Andreas Asuja

Helsinki 12. toukokuuta 2003

Kimmo Airamaa, Andreas Asuja, Mari Muuronen, Seppo Pastila, Virve Taivaljärvi

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos
Laudatur-projektityö, kevät 2003

Sisältö

1	Yleistä.....	1
2	Asennusohje	1
2.1	<i>Tietokanta.....</i>	<i>1</i>
2.2	<i>Sovelluksen asentaminen.....</i>	<i>1</i>
2.2.1	Tietokantayhteys	1
2.2.2	Käyttäjän tunnistus	2
2.2.3	Sovelluksen asennus	2
3	Asetukset.....	2
	Liite 1. Tietokannan luontiin tarvittavat SQL –lauseet.	4

1 Yleistä

Tämä dokumentti kuvaa OTUS –projektin seuranta-järjestelmän asennuksen sekä järjestelmän asetusten tekemisen. Järjestelmä vaatii toimiakseen J2EE 1.3 –määrityksen mukaisen sovelluspalvelimen sekä Postgres –tietokannan sekä JDBC –tietokanta-ajurin.

2 Asennusohje

Tässä ohjeessa kuvataan OTUS –järjestelmän asennus. Sovelluspalvelimena käytetään tässä esimerkkinä Tomcat 4.1:tä.

2.1 Tietokanta

Tietokanta luodaan liitteen 1 tietokannan luontilauseilla.

2.2 Sovelluksen asentaminen

2.2.1 Tietokantayhteys

Sovelluspalvelimeen täytyy asettaa tietokantayhteys. Sovelluspalvelin siis tarjoaa JNDI:n kautta tietokantayhteyden OTUS –sovellukselle. Tomcatissa asetuksen tehdään `server.xml` –tiedostoon (`[asennushakemisto]/conf/server.xml`). Asetukset ovat kuten alla:

```
<Context ...>
...
  <Resource name="jdbc/lohtudb" auth="Container" scope="Shareable"
            type="javax.sql.DataSource"/>
  <ResourceParams name="jdbc/lohtudb">
    <parameter>
      <name>factory</name><value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
    </parameter>
    <parameter>
      <name>driverClassName</name><value>org.postgresql.Driver</value>
    </parameter>
    <parameter>
      <name>url</name>
      <value>jdbc:postgresql://dbhost.myhost.org:10103/databasename</value>
    </parameter>
    <parameter>
      <name>maxActive</name><value>10</value>
    </parameter>
    <parameter>
      <name>maxIdle</name><value>2</value>
    </parameter>
    <parameter>
      <name>maxWait</name><value>1000</value>
    </parameter>
    <parameter>
      <name>username</name><value>bogus</value>
    </parameter>
    <parameter>
      <name>password</name><value>SK7bGzL9lG</value>
    </parameter>
  </ResourceParams>
...
</Context>
```

2.2.2 Käyttäjän tunnistus

Sovelluspalvelimen täytyy suorittaa käyttäjän tunnistus tietokantaan luotujen tunnuksien ja salasanojen mukaan. Tomcatissa asetuksen tehdään `server.xml` -tiedostoon (`[asennushakemisto]/conf/server.xml`). Asetukset ovat kuten alla:

```
<Context ...>
...
  <Realm className="org.apache.catalina.realm.JDBCRealm" debug="99"
    driverName="org.postgresql.Driver"
    connectionURL="jdbc:postgresql://dbhost.myhost.org:10103/databasename "
    connectionName="bogus" connectionPassword="SK7bGzL9lG"
    userTable="Users" userNameCol="username" userCredCol="password"
    userRoleTable="Users" roleNameCol="role" />
...
</Context>
```

2.2.3 Sovelluksen asennus

Tietokanta-ajuri täytyy kopioida sovelluspalvelimen ajurihakemistoihin niin, että se on saatavilla sekä käyttäjätunnistusta että sovelluksen tarvitsemaan tietokantayhteyttä varten. Tomcatissa ajuri pitää kopioida kahteen hakemistoon: `[asennushakemisto]/common/lib` sekä `[asennushakemisto]/server/lib`.

Itse sovellus toimitetaan `war` -tiedostona (lohtu.war). Tiedosto laitetaan sovelluspalvelimen sovellushakemistoon. Tomcatissa tämä on `[asennushakemisto]/webapps`. Jos sovelluspalvelin on määritelty automaattisesti ottamaan käyttöön uudet sovellukset, pitäisi sen automaattisesti avata lohtu.war ja luoda hakemisto `[asennushakemisto]/webapps/lohtu`. Näin on Tomcatissa oletusasetuksena.

3 Asetukset

Kaikki itse sovelluksen asetukset tehdään sen `web.xml` -asetustiedostoon. Tomcat -ympäristössä se löytyy tiedostopolusta `[asennushakemisto]/webapps/lohtu/WEB-INF/web.xml`.

Se sisältää neljä asetusta, jota saattaa joutua muuttamaan.

1. Lokitiedoston nimi. Tiedosto, johon järjestelmä kirjoittaa lokinsa.
2. Lokitustaso. Voi sisältää arvot `SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER` tai `FINEST`. Suositeltava lokitustaso tuotantokäytössä on `INFO`. `SEVERE` asetus tekee vähiten ja `FINEST` eniten lokitusta.
3. Tietokannan JNDI -nimi. Tämän tulee olla sama kuin kappaleessa 2.2.1 asetettu `server.xml` -tiedoston `Resource` -elementin `name` -attribuutin arvo (esimerkissä `jdbc/lohtudb`).
4. Sovelluksessa käytettävä päivämäärien esitysmuoto. Muoto on asetettavissa, kuten Java -luokan `java.text.SimpleDateFormat` Javadoc -dokumentissa kuvataan (<http://java.sun.com/j2se/1.4.1/docs/api/java/text/SimpleDateFormat.html>).

Kaikki em. asetukset asetetaan sovelluksen alkuasetus servletissä (Lohtu Application Initializer). Alla on esimerkki asetuksista.

```
<web-app>
...
  <servlet>
    <servlet-name>Lohtu Application Initializer</servlet-name>
    <servlet-class>fi.helsinki.cs.lohtu.util.LohtuInit</servlet-class>
    <init-param>
      <param-name>logFilename</param-name>
      <param-value>logs/lohtuLog</param-value>
    </init-param>
    <init-param>
      <param-name>logLevel</param-name>
      <param-value>INFO</param-value>
    </init-param>
    <init-param>
      <param-name>jndiRefToDatabase</param-name>
      <param-value>jdbc/lohtudb</param-value>
    </init-param>
    <init-param>
      <param-name>dateFormat</param-name>
      <param-value>d.M.yyyy</param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
  </servlet>
...
</web-app>
```

Liite 1. Tietokannan luontiin tarvittavat SQL –lauseet.

```
CREATE TABLE SoftwareTypes (  
swId Serial NOT NULL,  
softwareType varchar(50) NOT NULL,  
primary key (swId),  
unique (softwareType));  
  
CREATE TABLE ProcessModels (  
pmId Serial NOT NULL,  
processModel varchar(50) NOT NULL,  
primary key (pmId),  
unique (processModel));  
  
CREATE TABLE Datatypes (  
dtId Serial NOT NULL,  
datatype varchar(50) NOT NULL,  
primary key (dtId),  
unique (datatype));  
  
CREATE TABLE MandatoryQuantities (  
mqId Serial NOT NULL,  
pmId integer NOT NULL,  
dtId integer NOT NULL,  
name varchar(50) NOT NULL,  
primary key (mqId),  
unique (name),  
foreign key (pmId) references ProcessModels,  
foreign key (dtId) references DataTypes);  
  
CREATE TABLE Projects (  
projectId Serial NOT NULL,  
pmId integer,  
swId integer,  
cycles integer default 1,  
state boolean default true,  
projectName varchar(50),  
primary key (projectId),  
foreign key (pmId) references ProcessModels,  
foreign key (swId) references SoftwareTypes);  
  
CREATE TABLE ProjectQuantities (  
pqId Serial NOT NULL,  
projectId integer NOT NULL,  
dtId integer NOT NULL,  
name varchar(50) NOT NULL,  
primary key (pqId),  
foreign key (projectId) references Projects,  
foreign key (dtId) references DataTypes);  
  
CREATE TABLE QuantityStages (  
stageId Serial NOT NULL,  
stage varchar(15) NOT NULL,  
primary key (stageId),  
unique (stage));  
  
CREATE TABLE Roles (  
role varchar(10) NOT NULL,  
primary key (role));  
  
CREATE TABLE Users (  
userId Serial NOT NULL,  
projectId integer NOT NULL,  
username varchar(150) NOT NULL,  
password varchar(10) NOT NULL,  
role varchar(10) NOT NULL,
```

```

firstName varchar(20) NOT NULL,
lastName varchar(20) NOT NULL,
primary key (userId),
unique (username),
foreign key (projectId) references Projects,
foreign key (role) references Roles);

CREATE TABLE Components (
componentId Serial NOT NULL,
projectId integer NOT NULL,
componentName varchar(50) NOT NULL,
primary key (componentId),
foreign key (projectId) references Projects);

CREATE TABLE Phases (
phaseId Serial NOT NULL,
cycleId integer NOT NULL,
projectId integer NOT NULL,
plannedStart date,
plannedEnd date,
freezedStart date,
freezedEnd date,
actualStart date,
actualEnd date,
phaseName varchar(50) NOT NULL,
primary key (phaseId),
foreign key (projectId) references Projects);

CREATE TABLE Tasks (
taskId Serial NOT NULL,
phaseId integer NOT NULL,
componentId integer,
milestone boolean NOT NULL,
deadline date,
freezedDeadline date,
actualDeadline date,
taskName varchar(50) NOT NULL,
primary key (taskId),
foreign key (phaseId) references Phases,
foreign key (componentId) references Components);

CREATE TABLE UserTasks (
taskId integer NOT NULL,
userId integer NOT NULL,
foreign key (taskId) references Tasks,
foreign key (userId) references Users);

CREATE TABLE TaskQuantities (
quantityId integer NOT NULL,
stageId integer NOT NULL,
taskId integer NOT NULL,
dtId integer NOT NULL,
mqId integer,
pqId integer,
value varchar(100),
name varchar(50) NOT NULL,
primary key (quantityId, stageId, taskId),
foreign key (stageId) references QuantityStages,
foreign key (taskId) references Tasks,
foreign key (dtId) references DataTypes,
foreign key (mqId) references MandatoryQuantities,
foreign key (pqId) references ProjectQuantities);

CREATE TABLE Measures (
    taskId integer NOT NULL,
    quantityId integer NOT NULL,
    userId integer NOT NULL,
    value varchar(100) NOT NULL,

```

```
foreign key (userId) references Users);
```