

singletonejb-javaee6 - A simple JSF+Singleton+ejb injection application

Application overview

This sample demonstrates **@Singleton** and **@EJB** annotations described in EJB3.1 lite as well as the usage of JSF framework. **@Singleton** annotation defines a singleton session bean which can be shared across applications and with its states maintained during the application's lifecycle. **@EJB** annotation shows how to have another session bean injected to your backing beans.

Application content

singletonejb-javaee6 application consists of following list of packages and classes.

org.apache.geronimo.samples.javaee6.singletonejb.backbeans

- AddCalculatorBackBean.java
- SubtractCalculatorBackBean.java

org.apache.geronimo.samples.javaee6.singletonejb.sessionBeans

- SingletonCalculator.java

The list of web application files in the application is depicted in the following.

```
- WEB_INF
  +- web.xml
  +- geronimo-web.xml
- index.html
- index.xhtml
- SubtractCal.xhtml
```

Application implementation

The **web.xml** defines the **welcome-file** of the web application **index.html** and declares the involvement of JSF framework with a standard servlet **Faces Servlet** for request processing.

```
xml
```

geronimo-web.xml describes information about the project (e.g. module's unique identification, any dependencies) inside the **<sys:environment>** tag. It is a good idea to give this module some sort of unique identification, so that it can later be referenced by some other deployable application. The path specified in the **<context-root>** tag will be the first segment of the URL used to access this web application. So to access this web application the url will be **http://<hostname>:<port>/Singletonejb-javaee6**. Geronimo uses **Apache MyFaces** as JSF implementation and the component is started by default and available server-wide. Therefore you do not have to add the dependency for myFaces component.

```
xml
```

index.xhtml uses the standard HTML tag library to render a form for user input and bind the components on the form with the managed beans properties respectively.

```
html
```

AddCalculatorBackBean.java and **SubtractCalculatorBackBean.java** hold the form data and invoke a singleton session bean **SingletonCalculator** for the calculation. Also the JSF managed beans use **@EJB** annotation for bean injection. We can see two references of **SingletonCalculator** for *Add* and *Subtract* beans. Because both backing beans **AddCalculatorBackBean** and **SubtractSingletonCalculatorBackBean** refer to the same object identity, each calculation will be performed on the result from last calculation.

```
50% javaAddCalculatorBackBean.java 50% javaSubtractCalculatorBackBean.java
```

SingletonCalculator.java do the actual calculation. It's a singleton session bean and can only be initialized once for all requests. It will be destroyed only when the application is stopped.

```
java
```

Get the source code

Please refer to [Samples General Information](#) for informations on obtaining and building the source for this and other samples.

Build the web application

Once all the sources get checked out the next step is to build **singleton-javaee6** sample. It requires Maven 2 or above for building the binaries.

From the **<singletonejb-javaee6_home>** directory run the following command.

```
mvn clean install
```

This process will take a couple of minutes. The binaries will be generated in the corresponding **target** directory .

Deploy the web application

Deploying sample application is pretty straight forward as we are going to use the Geronimo Console.

1. Scroll down to **Deploy New** from the **Console->Navigation panel**.
2. Load **singletonejb-javaee6-war-*version*.war** from **singletonejb-javaee6-war/target** folder in to the **Archive** input box.
3. Press **Install** button to deploy application in the server.

Test the web application

The application is visible at <http://localhost:8080/singleton-javaee6>.

Input a value and click **Add** for calculation.

.

Click **Go to Subtract Calculator** and input a value again, then click **subtract**. You will see the output is based on the result from the previews step.

.