ejbtimer-javaee6 - A simple EJB timer service application

Application Overview

This sample demonstrates a calender based timer service that triggers certain callbacks to enterprise beans at a specified time or interval programmatically. The cron-style timer service is managed by the EJB container and can be created using **@Schedule** annotation.

Also this sample introduces the annotation way to define servlets, stateless session beans and Persistence Context.

Application Content

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

org.apache.geronimo.samples.javaee6.ejbtimer

- **ShowLog.java** displays the detailed activities of all created timer.
- **ScheduleTask.java** is a stateless session bean which creates timers automatically.
- **logFacade.java** is a stateless session bean which retrieves the detailed activities of all timers.
- **Log.java** is a persistent unit to preserve each timers' activity.

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

+-WEB-INF

    |_ web.xml

    |_ geronimo-web.xml

+- Resources

    |_ persistence.xml

|_ index.html

|_ header.html

|_ intro.html

Application Implementation

geronimo-web.xml specifies the module's information and the url for the web application

Information about the project sucha as module's unique identification, dependencies is described inside the **<sys:environment/>** tags. It is a good practise to give a module an unique identification, so that it can later be referenced by some other deployable application. This module is in the group **org.apache.geronimo.samples**. The path specified in the **<context-root>** tag will be the entry point of this web application. Therefore you can access this web application at **http://<hostname>:<port>/ejbtimer-javaee6**.

xmlgeronimo-web.xml

ShowLog.java is a servlet component which is annotated with **@WebServlet** and defines the url pattern using **urlPatterns** attribute. And a business interface of the session bean **logFacade** is injected using **@ejb** annotation so that the servlet could invoke `count()` and `findAll()` methods from **logFacade**.

- **@WebServlet**
  The WebServlet annotation is used to annotate a servlet to respond to certain requests on a particular web url in the application. A class annotated with @WebServlet extends the class javax.servlet.http.ServletContextListener and must have the `urlPatterns` or `value` attribute specified.

javaShowLog.java

scheduleTask is a stateless session bean with multiple methods annotated with **@Schedule** annotations. Each method will be invoked according to the attributes specified to **@Schedule**. Also, there are two specified methods `logIntoDB(String event)` and `clearLogs()` to perform actions on an entity `EJBTimerPU` injected with **@PersistenceContext** annotation.

xmlscheduleTask.java

logFacade is a stateless session bean to define another two actions on an entity `EJBTimerPU`, where both methods are invoked from the servlet component **ShowLog**.

javalogFacade.java

persistence.xml is the configuration file to define a persistence unit named `EJBTimerPU`, which represents a table named **OPENJPASEQ** in a derby database `ejbtimerdb`.

xmlpersistence.xml

Log.java is an entity bean representing a primary key class of the database table. A primary key class must implement the `hashCode()` and `equals (Object object)` methods.

javaLog.java

Get the source code

Please reference Samples General Information for information 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 **ejbtimer-javaee6**. It requires Maven 2 for building the binaries.

From the **<ejbtimer-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 **ejbtimer-javaee6-war-3.0-SNAPSHOT.war** from **ejbtimer-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 app is visible at http://localhost:8080/ejbtimer-javaee6/

This page is about what kind of timers defined in the sample application and for each task, an log entry is created.
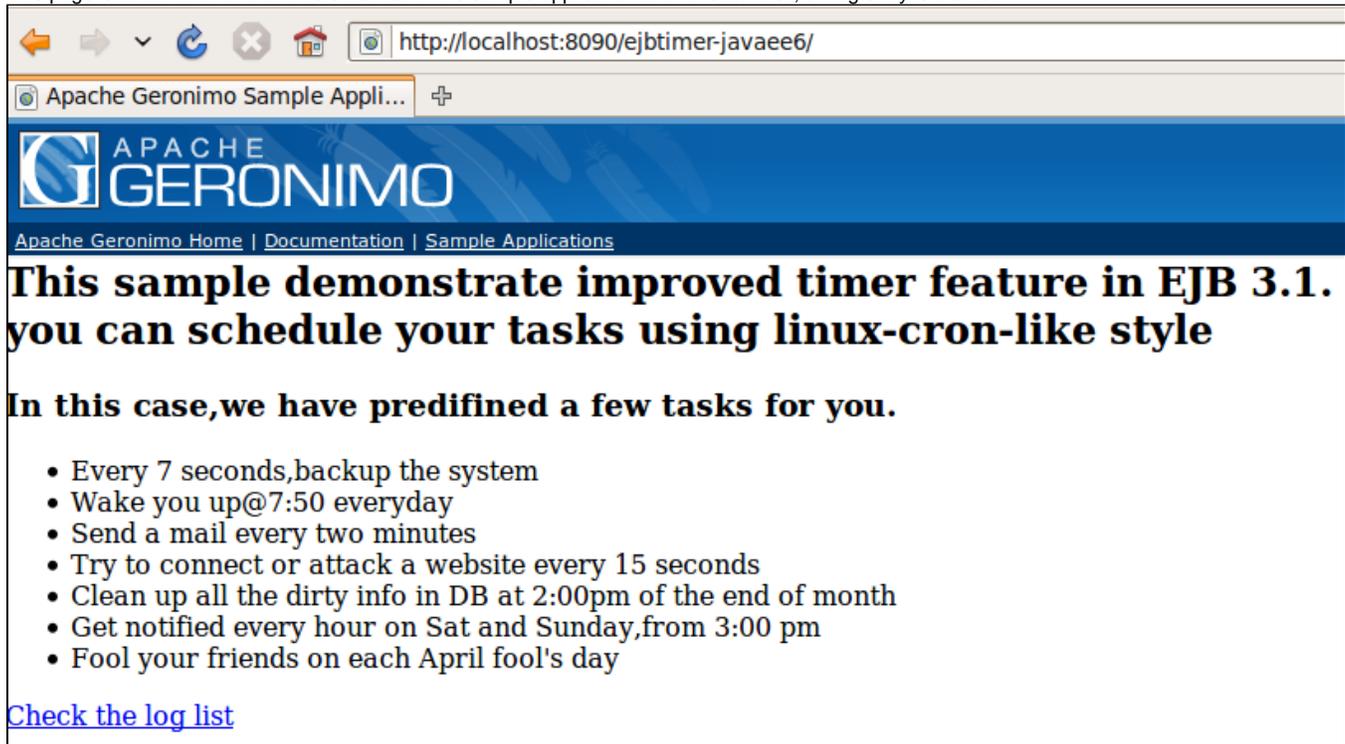
Click **Check the log list**, you will see all log info with specific time and event details.

Currently,there are 24 log records.

Log ID	Event Time	Event
1	2010/10/20 10:35:21	Every 7 seconds,backup the system
3	2010/10/20 10:35:22	Every 7 seconds,backup the system
2	2010/10/20 10:35:22	Every 7 seconds,backup the system
4	2010/10/20 10:35:22	Try to connect or attack a website every 15 seconds
5	2010/10/20 10:35:22	Every 7 seconds,backup the system
6	2010/10/20 10:35:22	Every 7 seconds,backup the system
7	2010/10/20 10:35:22	Every 7 seconds,backup the system
8	2010/10/20 10:35:22	Every 7 seconds,backup the system
9	2010/10/20 10:35:22	Try to connect or attack a website every 15 seconds
10	2010/10/20 10:35:22	Try to connect or attack a website every 15 seconds
11	2010/10/20 10:35:28	Every 7 seconds,backup the system
12	2010/10/20 10:35:30	Try to connect or attack a website every 15 seconds
13	2010/10/20 10:35:35	Every 7 seconds,backup the system
14	2010/10/20 10:35:42	Every 7 seconds,backup the system
15	2010/10/20 10:48:42	Every 7 seconds,backup the system
16	2010/10/20 10:48:35	Every 7 seconds,backup the system
17	2010/10/20 10:48:45	Try to connect or attack a website every 15 seconds