RESTful Web Services in Java

In this practice session, you should learn a method for programming data-centric Web services in Java using a "RESTful" style using the Jersey framework.
The tutorial relies on NetBeans and the Glassfish application server deployed on the ATS server. The tutorial also uses the MySQL database server for which you have been given a login, database name and a password. This database server is only accessible from within the university network. If you're working from a machine outside the university network, you may download the MySQL system and install it locally in your machine so that you can use your own database on localhost.

Warmup Exercise – Creating a Level 1 REST service

Design and implement a REST level 1 service that converts a temperature in Farenheit into an equivalent temperature in Celcius and vice-versa. The service must be implemented in JAX-RS and must be configured to return XML.

Hint: Netbeans provide automatic code generation, add a new "RESTful Web Services from Patterns" into your project.

Exercise 1 – Creating a Level 2 REST service from an existing database

- Download the sakila sample database for MySQL.
- In the practice handout of week 2 (MySQL and JPA) you saw how to connect to a MySQL server using NetBeans. Use those instructions to connect to MySQL. Once you do this, you will see in the "Databases" explorer (on the "Services" view), under the menu Drivers, a *connection node* (icon "jdbc://mysql://..."). Right-click on the connection node, and choose "Execute Command". This opens the SQL editor.
- Now, let's load the sakila sample data into your MySQL database. The sakila database comes in two files: one with all the commands for creating the schema (sakila-schema.sql) and another for populating the tables (sakila-data.sql).
- Copy/paste the commands to create the Sakila schema into the SQL editor. Delete the commands "drop schema", "create schema" and "use" because you don't have rights to create new schemas on the MySQL server. Instead, the tables will be created in your existing database. You should also delete the commands "create trigger", "create procedure" and "create function" because you do not have rights to create triggers nor stored procedures. To run the SQL commands, right-click anywhere in the SQL editor and choose "Run Statement".
- Repeat the operation to run the commands in the file "sakila-data.sql" in order to populate the tables. Note that you should remove the command "USE" at the beginning of this file. You should also remove the "create trigger" commands in this file.
- Complete the tutorial Exposing a MySQL Database with RESTful Web Services. You can also view the video tutorial. In this tutorial, it is assumed that you connect to a database called "sakila", but your database name is "esi_YourLogin". Also, in the step of the tutorial where you enter the properties for the persistence unit, you should enter your MySQL login details.
- Once you have created and compiled the project, you can upload the war file into the Glassfish server through the admin console as explained in previous practice sessions. Another option is to register the Glassfish server running in ATS with the NetBeans server manager. For this, go to the tab "services" on NetBeans, right-click on "servers", add a new server, and follow the screens, making sure that you select the option "register remote server". At some point you will have to enter the URL and login details of the Glassfish instance on the ATS server. Then go to the Project tab on NetBeans, right-click on the name of the Java Web application project you created above, select "Properties", then select the "Run" category, change the "Server" property to the remote Glassfish server. Once you have completed these steps, you can run the project from NetBeans. NetBeans will deploy the Web application into the remote Glassfish server.
- You can delete the tables/views created by this exercise using the NetBeans database browser.

Exercise 2 – Create your Level 2 REST service

- Read the following article RESTful Web Services: Concepts and Design. Create database tables corresponding to the example "building/room" in this article and populate these tables with a few sample tuples. Repeat the steps above to expose these tables as RESTful web services.