Distributed Systems: Java EE session 2 & 3

Wouter De Borger, Stefan Walraven, Steven Op de beeck,
Fatih Gey, Bert Lagaisse and Wouter Joosen

November 3 and 10, 2015

Overview

There will be 3 exercise sessions on Java EE:

- 13/10/15 in PC lab:
  - Introduction to Java EE and session beans.
  - Debugging in the context of Java EE.
  - **Submit** your code on Toledo **before Friday, October 16, 19:00**.
- 03/11/15 in PC lab:
  - Introduction to Java EE persistence.
  - **No code to submit.**
- 10/11/15 in PC lab:
  - Java EE persistence and transactions.
  - **Submit** the final version of your code on Toledo **before Friday, November 13, 19:00**.

In total, **each** student must submit 2 assignments to Toledo.

1 Introduction

Goal: The goal of this session is to design and implement an application using persistence and transactions in Java EE.

Approach: This session must be carried out in groups of *two* people. Team up with the same partner you collaborated with in the previous sessions.

Submitting: *Each* student must submit a zip file to the `Assignment` section of Toledo, with his or her combined results of the 2nd and 3rd session *before* the deadline stated in the overview above.

To do so:

1. Clean your project through NetBeans
2. Create a zip file of your `CarRental` project using the following command in the terminal:

```
zip -r firstname.lastname.zip <your project directory>
```

Important:

- When leaving, make sure your server is no longer running.
- NetBeans projects have a `nbproject` directory. In this directory, there is a `private` subdirectory. When moving NetBeans projects to other machines, make sure to remove all `private` folders in all projects.
- Retain a copy of your work for yourself, so you can review it before the exam.

2 Preliminaries

1. Start NetBeans by executing the following command:

   ```
   /localhost/packages/ds/netbeans/bin/netbeans
   ```

2. Set up a domain, as described in the first assignment.
3. Download `ds_jee_2.zip` from Toledo. Extract the contents of the zip file into your home directory. The zip file contains the Java EE car rental application (a NetBeans project).
4. Open the `CarRental` project, as described in the first assignment.

3 Assignment

The goal of sessions 2 and 3 is to modify the given application such that the car rental companies, cars and reservations are stored in a database. More specifically:

- The classes `CarRentalCompany`, `Car`, `CarType`, and `Reservation` should be changed to `Entity` classes, and the appropriate relationships between the entities should be defined. After doing so, you should be able to delete the class `RentalStore`. However, it's best to move it aside as you might need its data loading code later.
- Add a way for managers to add car rental companies, car types and cars.
- Extend the client so that it uses the management interface to load the car rental companies and their cars (and their respective car type) into the database.
- Managers should be able to query information and to retrieve different statistics to support customer profiling and advertising. Because of efficiency reasons, it is recommended in Java EE to use persistence queries (JPQL) without additional application logic in order to retrieve the information of interest. To query a specific entity with a given key, `em.find()` is the recommended approach. Can you explain those efficiency reasons?

  Therefore, use JPQL as much as possible, especially for the following queries for the manager:

  1. look up all car rental companies
  2. look up all car types in a company
  3. retrieve the number of reservations for a particular car type in a car rental company
  4. retrieve the number of (final) reservations made by a particular car renter
  5. retrieve the cheapest car type available in a given time period
  6. retrieve the most popular car rental company (i.e. highest total of reservations)

- Car renters should be able to make reservations at multiple car rental companies. A list of `Quotes` (i.e. tentative reservations) should be kept on the server for the duration of the car renter's session. It is not necessary to make this list persistent. Once a car renter decides to confirm his quotes, all tentative reservations should be effectively stored in the database. If one of the reservations fails (e.g. because no free car of the specified car type is available anymore), all reservations corresponding to the session should fail (rollback). Use the transaction support of Java EE.
- Ensure that no overlapping reservations for the same car can be created!

Note:

1. Depending on your implementation strategy, the client application might have redundant parameters and/or it can be required to add extra classes, methods and other changes.
2. Throughout this assignment, always consider good software engineering and coding practices. More specifically: think about where to implement new behavior, respect encapsulation, use the proper arguments, don't scatter JPQL queries throughout the code, etc.

4 Important Remarks

- You have to create a new persistence unit before you can run an application using persistence entities. To create a new persistence unit, select the `CarRental-ejb` project, right-click the project name and choose `New` → `Other...`. In the "New File" window, choose `Persistence` → `Persistence Unit` and click "Next". Select `jdbc/sample` as your data source, keep the default persistence provider, and choose **"drop and create"** as your table generation strategy. Make sure the persistence unit uses the Java Transaction API. Click "Finish". A `persistence.xml` file is created in the folder "Configuration Files". Also make sure to explicitly include all entities in the list **"Include Entity Classes"**. Otherwise the application server may be incapable of detecting the entities in the `CarRental-lib` project.
- When running the application client within NetBeans, it will redeploy the application and reinitialize the database each time you click `Run`. However, do not continuously try to deploy your application when your (server) implementation is not finished. *"Trial-and-error" is bound to go wrong!*
- When defining relationships, do not forget to set the `CascadeType` and `FetchType` when necessary. Check the default settings.
- Fields of type `java.util.Date` in entity classes must be annotated with the `@Temporal` annotation. The correct `TemporalType` is `DATE`.
- Often it is useful to have Java EE generate primary keys for you. To do so, annotate the primary key field with `@GeneratedValue` with `AUTO` or `IDENTITY` as the strategy.
- Do not fully rely on NetBeans (e.g. warnings in yellow): you have to be able to explain everything!
- To inspect the database structure, use the services tab and under "Databases", right-click on `jdbc:derby://localhost:1527/sample [app on APP]` and select connect. In the `APP` database you will find the tables for your project.

Important: In case of problems, try the following things:

- Undeploy all applications from the application server in the services tab (especially the client application). Restart the application server.
- Check if you have enough space in your home directory (via `quota` and `du | sort -n`).
- Remove all your old NetBeans configuration and settings by deleting the folders `.netbeans`, `.netbeans-derby` and `.cache/netbeans` in your home directory.
- When you get a pop-up to unlock the login keyring, fill in your login password. When this fails, remove everything under `.gnome2/keyrings/`. Next time the pop-up shows up, fill in your login password.

5 Documentation

Several sources of documentation are useful, apart from the slides and the textbook:

- Java EE 5 tutorial: `http://docs.oracle.com/javaee/5/tutorial/doc/`
- Java EE 7 API doc: `http://docs.oracle.com/javaee/7/api/`
- Java EE example applications are available at `/localhost/packages/ds/javaeetutorial5/`. The most instructive example is the `Roster` application. To use the project, you need to copy the master project (shared by all examples) and the `Roster` project to a writable location (for example: your home dir ∼). To do this, run the following command: `/localhost/packages/ds/javaeetutorial5/copyExample.sh ∼ ejb/roster`. Then the project can be opened in NetBeans.

Good luck