Workshop 22: Persistence and ORM

DAT076/DIT126 Chalmers/Göteborgs Universitet
Joachim von Hacht

Objectives

Handling the persistence layer of a Web Application.

Resources

Code samples from l22, see lectures.
MySQL reference manual
Java Db Derby reference
JDBC reference
JEE Java Persistence API (JPA) reference.
Many code samples for JPA.
Express MySQL
Npm package mysql
Sequelize

Prerequisites

We need a database (server application). It should be possible to get an MySQL database from Chalmers IT service (not tested).

Other options

- Use the Derby database (AKA JavaDB) bundled with NetBeans. Also possible to download (with command line interface ij)

Instructions to create a MySQL database. Some steps may be different (or omitted) if using Chalmers IT Service/MySQL.

1. mysql > create database todo; (MySQL and case sensitive for names!)
2. mysql > create user, example 'hajo'@'localhost' identified by 'hajo';
3. mysql > grant all privileges on todo.* (the database) to 'hajo'@'localhost';
4. mysql > quit
5. mysql -u hajo -p (log in as newly created user with password is 'hajo')
6. mysql > use todo; (use new database)
7. mysql > source create_todo_db.sql (create table and populate, using script in same dir). The script is supplied in the zip-files below.

Also possible to use MySQl from inside NetBeans (create database in IDE).

If installing your own MySQL you need to do some preparations, here is a tutorial and here's another one. The steps are (on Linux):

1. Start mysqld (the server): sudo /etc/init.d/mysql start
2. Check server is up: ps aux | grep mysql ("mysqld" should show somewhere)
3. Start the mysql client (as root): mysql -u root -p
4. Continue as above to create database.

Data from prerequisites needed below to configure database connection, so better remember or make a note.

Inspecting Non-ORM usage

Download **l22.zip** from lectures. Try to run (and do some inspections, code, tables, …). Must have a working database!

1. /l22/node_mysql
2. /l22/jee_jdbc

Using ORM

We'll continue with the TODO-note application but with simplified functionality. We just list, add and delete notes, but … the notes will be stored in a database! Pages are very much the same as previous workshop, but simplified. Like this:

Choose one of the two approaches below

1. JEE (Java Persistence API (JPA), Tomcat)

This uses JPA and Querydsl.

1. Try to run l22/jee_orm
2. Download **jee_mysql_orm.zip**
3. Inspect dependencies.
4. It should be possible to run (but no functionality)
5. Inspect persistence.xml adapt to your needs (later, after successful build, inspect Generated Sources)
6. Look for //TODO comments what to do.
7. Try to run l22/jee_query
8. Add some queries using Querydsl.

2. NodeJS (JavaScript, Express, Sequelize)

This uses Sequelize. Nice Express/Sequelize samples

1. Try to run /l22/node_orm
2. Download **node_mysql_orm.zip**
3. Should be possible to run (but no functionality)
4. Modify connection data in db/todo_db_orm.js to adapt your needs
5. Look for //TODO comments what to do (no pages need to be modified).
6. Try to run some ad hoc queries (not raw SQL)

Optional

- For the ORM parts: Try to use associations between objects i.e. database tables have relationships. Possibly connect a note to an author or similar, use your imagination … Of course you must create a table for authors.
- Try the other approach.

Examination

Contact teaching assistant for a demo.