CPSC 476 - Java Enterprise Application Development

Programming Assignment 2 - Spring 2016

due March 29

In this assignment, you will refactor your code from Programming Assignment 1 to introduce the Spring Framework. You will then use Spring's JDBC support to persist event and user information in a relational database.

Complete the following steps:

1. Download and install Apache Ant.

2. Use Ant to install and run HSQLDB. (See below for details.)

3. Modify your project's Maven dependencies to include

    a. The Spring Framework, including Spring JDBC support
    b. Apache Commons DBCP
    c. The JDBC driver for HSQLDB

4. Create Data Access Objects (DAOs) and interfaces for each entity in the system.

5. Add a Spring `ApplicationContext` to your web application and configure it to create:

    a. An Apache Commons DBCP `DataSource`

    b. A Spring `JdbcTemplate` object

    c. Instances of each DAO

6. Modify your servlets to use DAOs to store and retrieve users and events. The rest of the code should remain unchanged from the previous assignment.

Installing and Using HSQLDB

1. Make sure that Ant is installed

2. Download `build.xml` and place it in a new, empty directory

3. Type `ant -projecthelp` to see the available commands

Once HSQLDB has been downloaded and installed, the database name, username, and password can be found in `etc/sqltool.rc`.

Tips

- There may not be a 1:1 correspondence between entities and tables; a DAO might need to access more than on table.
- Do not use Hibernate, JPA, or Spring Data for this assignment.
- Consider deriving your DAO implementation classes from `JdbcDaoSupport`.
- Use `ContextLoaderListener` to add an application context to your application, rather than `DispatcherServlet`. `DispatcherServlet` is used for Spring MVC.
- As alternative to creating the `DataSource` from Spring, you may wish to configure a container-managed `DataSource` for Tomcat. (See *Spring in Practice*, Recipe 2.2, or the end of Chapter 19 in the textbook)

Further Reading

- *Spring in Practice*, Recipes 2.1 and 2.4
  - (*NOTE*: this reference no longer available through library.fullerton.edu*)*
- *Spring in Action*, Chapter 10
- *Spring Recipes, Third Edition*, Chapter 10
- *Pro Spring, Fourth Edition*, Chapter 6
- *Learning Spring Application Development*, Chapter 3
- *Spring Essentials*, Chapter 3
- *Spring Cookbook*, Chapter 4

Working with Other Students

You may complete this project on your own, but you are encouraged to work together with one or two other students.

If you choose to work with others:

- Submit only a single assignment.
- Include the names of all members of a group on the submission.
- Each student in a group will receive the same grade.
- You may discuss the assignment with other groups, but each group must submit its own work.
- You may choose to work with a different group on future assignments.

Submission

E-mail the following in a single `.ZIP` or `.tar.gz` file to csuf.kenytt.net@gmail.com by 11:59p on the date indicated:

1. Your Maven `pom.xml` or other build / project files.

2. The `.java` and `.JSP` source code for your application

3. Any additional resources (e.g., graphics, deployment descriptors, XML files, SQL DDL scripts) required by your application.

4. Documentation in PDF format, including

    a. A brief description of your project's architecture
    b. Build instructions, if any are required beyond a simple `mvn package` command.
    c. Screenshots demonstrating your application's functionality

Include your name and the other members of your team (if you have one) in your e-mail.

Do not submit compiled code or `.WAR` files. Your submission should be in a condition to be re-built from source.

Set the `Subject:` line of your e-mail to

```
[CPSC 476] Assignment 2
```

You may submit multiple times before the deadline; I will only grade the most recent submission before the deadline, unless your e-mail indicates that I should do something else. Late work will not be accepted after the deadline.

Grading

This programming assignment will be graded on the following 10-point scale, based on the instructor's judgement of the amount of time required to submit A-level work:

10	Application builds and runs correctly as described above.
9	Minor issues with submission or format. Work required to correct the issues is likely to be less than 20 minutes.
8	Minor issues with the application. Work required to correct the issues is likely to be less than two hours.
7	Major issues. Work required to correct the issues is likely to be more than two hours.
6	Application substantially fails to meet requirements, but shows a good faith effort to attempt the project.
5	Application substantially fails to meet requirements. Minimal progress made.
4	Code submitted without documentation, or code fails to compile.

0	No assignment submitted.