Pre-Class Assignment

More Spring Topics: AOP Using Spring; Spring Integration w/Hibernate; Spring Java Configuration

Here is a list of materials you will need to read to prepare for the next lab period. Read everything carefully to make sure you understand the concepts presented here. There will be a reading quiz at the beginning of class.

What to learn

- Java Configuration Annotations
- Compare and Contrast AspectJ and Spring AOP
- Benefits of using Spring to integrate Hibernate (Object Relational Mapping)

Readings

Spring in Action, 4th Edition by Craig Walls
https://learning.oreilly.com/library/view/spring-in-action/9781617291203/

Note: You may need to access Safari through the library website and sign in with your BYU email address to access the book.

Java Configuration: Section 2.3
Spring AOP: Sections 4.1.2, 4.4 - 4.4.1

Spring Documentation on Object Relational Mapping:
https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/data-access.html#orm-introduction
Read section 4.1 only

Pre-Class Assignment

This Spring assignment will be built upon the last one. Use the same Spring project that you have turned in for the last assignment.

Upgrade your database to MySQL

For integrating Hibernate with your current Spring project, we will need to use MySQL as our database. If MySQL is no longer installed on your computer, follow the instructions in the

Hibernate pre-class assignment to install the database. Complete the following steps when you have MySQL installed:

1. Open the MySQL Command Line Client. If you need a refresher on how to start mysql, refer to the pre-class assignment for the hibernate tutorial. Use the following commands to create a database and select it:

   CREATE DATABASE studentdb;

   USE studentdb;
2. Open the table_script.sql file. Copy and paste its contents into the Command Line Client. This was the same script used to create the tables in the data.sqlite database from the previous lab that is updated for MySQL.
3. Enter the command *SHOW TABLES;* You should see the following:

```
mysql> show tables;
+--------------------+
| Tables_in_studentdb |
+--------------------+
| cdh                |
| classes            |
| cp                 |
| csg                |
| students           |
+--------------------+
```

4. To make sure there is information in the database, you can do the command: SELECT * FROM *TableName.* For example, you should see the following by entering the command *SELECT * FROM students;* :

```
mysql> select * from students
    -> ;
+-----------+-------------+---------------+----------+
| studentID | name        | address       | phone    |
+-----------+-------------+---------------+----------+
|     12345 | C. Brown    | 12 Apple St.  | 555-1234 |
|     22222 | P. Patty    | 56 Grape Blvd | 555-9999 |
|     33333 | Snoopy      | 12 Apple St.  | 555-1234 |
|     67890 | L. Van Pelt | 34 Pear Ave.  | 555-5678 |
+-----------+-------------+---------------+----------+
4 rows in set (0.00 sec)
```

5. Your MySQL database is now ready to go.

Using IntelliJ Ultimate

Make sure you are using IntelliJ Ultimate for this lab to access all features IntelliJ has for Spring, AOP, and Hibernate. Instructions on how to get it are found in the AOP pre-class assignment.

Import Additional Dependencies

1. Open up your SpringTutorialProject from the previous assignment in IntelliJ Ultimate.
2. You will need to import additional dependencies for the latest production version of **spring-aop**, **aspectjweaver**, **spring-orm**, **hibernate-core**, and **mysql-connector-java** to the Maven **pom.xml** file. You will need spring-aop and aspectjweaver to implement AOP in Spring, while the last three dependencies are for Hibernate integration with Spring. Look at the websites here for the latest versions of these tools:

    Tools for Spring AOP
    org.springframework » spring-aop
    org.aspectj » aspectjweaver

    Tools for Hibernate Integration
    org.springframework » spring-orm
    org.hibernate » hibernate-core
    mysql » mysql-connector-java

Now you are ready to go on completing this Spring assignment. You will see how simple it is to enable logging on our DAO classes using Spring AOP. Next, you will see how JDBC can be replaced easily with an ORM framework like Hibernate. Lastly, you will be exposed to one of the most popular Spring configuration methods: JavaConfig.