The Learning House

427 S. 4th Street #300

Louisville KY 40202

Java and Databases Unit

Lesson 1 – Introduction to JDBC Templates

SOFTWARE GUILD

Objectives

- Create sample app database
- Create domain objects
- Create DAO using JDBCTemplate
- Configure data source and pooling
- Use DI to inject data source into JDBCTemplate
- Use DI to inject JDBCTemplate into DAO
- Create and execute test suite for DAO
- Create Spring config files for DAO and tests

Spring Data Access Philosophy

- Both Service Layer and DAO
- Program to interfaces
- Platform agnostic exceptions
- Exceptions are all unchecked

Data Access Templates

- Many flavors (JDBC, Hibernate, JPA, etc.)
- Templates take care of fixed pieces of the data access process for us:
  - Preparing resources
  - Starting transactions
  - Commits/rollbacks
  - Closing resources/error handling

Example: Student Roster

- We will use the Spring container
- We will create two DAO implementations:
  - In memory
  - Database
  - We'll use DI to switch between the two
- We'll use Spring JDBC Templates to access the database

Example: Student Roster (2)

- Look at In Memory data store first
- A few considerations:
  - o DAO will consist of a DAO Interface and (in our case) 2 implementations
  - o Only program to the Interface
  - o All code outside the DAO must be unaware of which DAO implementation is in use

DAO Layer (DB) Overview

- Start with database and tables
- Create domain objects
- Create DAO
- Configure data source and pooling
- Configure JDBC Template
- Configure DAO
  - For both tests and application

Create Database

- Create both StudentRoster and StudentRoster_test databases

Create Domain Objects

- Simple POJOs

Create DAO

- Simple POJO
- Create both interface and implementation
- Implementation has a JDBCTemplate member
- Use final string for SQL statements

Create Config Files

- applicationContext.xml (app)
- test-applicationContext.xml (test)

Add Configuration Elements

- Data Source
- Pooling
- JDBCTemplate
- DAO
- Add to both files

Create Test Suite

- Spring context will be loaded manually - just like we do in the application
- Test suite should work on either implementation
- As we implement, pay close attention to the new class of errors that can occur when dealing with an actual database