

Abstract geometric lines in the top left corner.

웹개발 백엔드 포트폴리오 만들기 2차 과제

오혜진

과제 목표

Spring Framework 프로젝트에
데이터베이스 (DBMS) 연동하기

제약조건

1. Oracle 또는 MySQL 사용



사용한 DB, 테스트 프레임워크

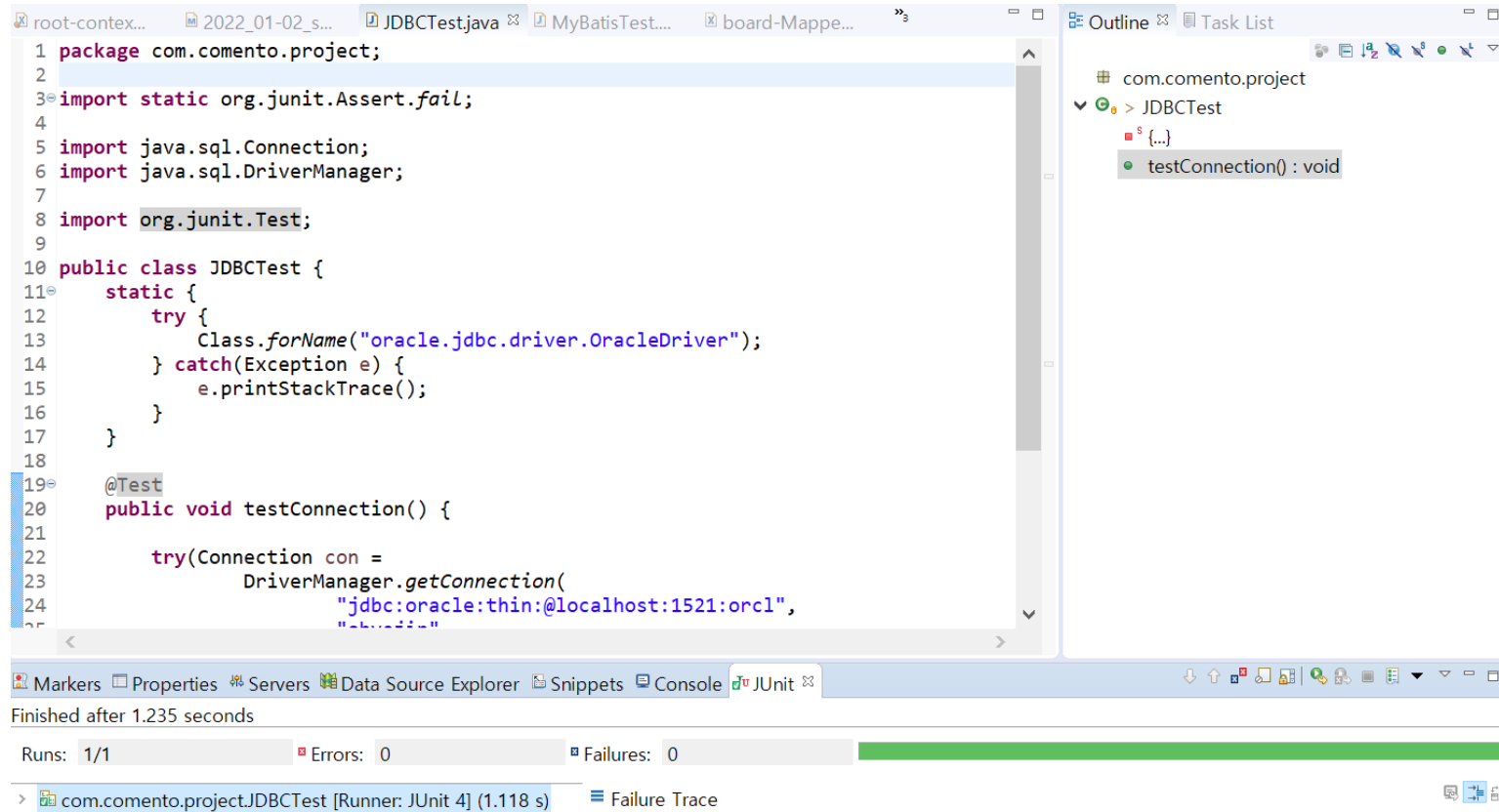
Oracle & SQL Developer

DB로는 오라클을 선택했고, SQL로 작업하기 위한 통합 개발 환경(IDE)로는 SQL Developer를 사용했다.

Junit

빠르고 간단하게 DBMS 연결을 확인하는 방법을 모색했다. 그 도중 Junit이란 걸 알게 되었고, 자바 클래스 하나만 생성해서 단위 테스트를 진행할 수 있었다.

JDBC 연결 테스트 캡처 화면



The screenshot displays an IDE with the following components:

- Editor:** Contains the code for `JDBCTest.java`. The code imports `org.junit.Assert.fail`, `java.sql.Connection`, `java.sql.DriverManager`, and `org.junit.Test`. It defines a `JDBCTest` class with a static block that attempts to load the Oracle JDBC driver. The `@Test` annotated `testConnection()` method uses `DriverManager.getConnection` with the URL `"jdbc:oracle:thin:@localhost:1521:orcl"` and the username `"abudif"`.
- Outline:** Shows the project structure with `com.comento.project` and the `JDBCTest` class containing the `testConnection() : void` method.
- JUnit Console:** Indicates the test finished after 1.235 seconds.
- JUnit Results:** Shows 1/1 runs, 0 errors, and 0 failures. A green progress bar indicates a successful test.
- Failure Trace:** A section for viewing test failures, currently empty.

◀ Junit을 사용하여 DB연결을 테스트하는 모습

결과창에서 초록색 막대가 나온다면 테스트 메소드에 문제가 없다는 뜻이다.

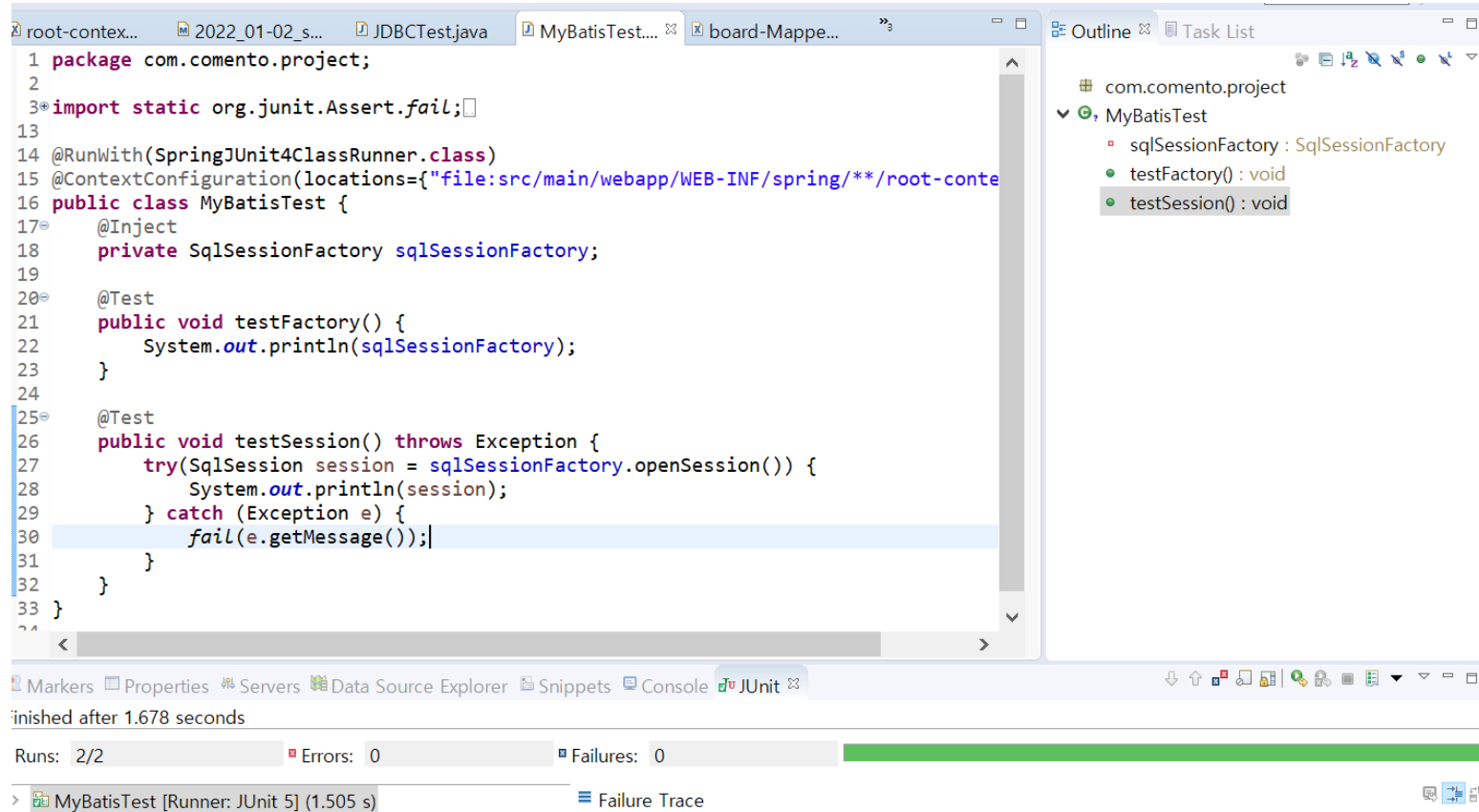
드라이버 네임, url에 이상이 없다는 걸 확인했다.

MYBATIS SQLSESSIONFACTORY

MyBatis는 개발자가 지정한 SQL, 저장프로시저 그리고 몇가지 고급 매핑을 지원하는 퍼시스턴스 프레임워크이다.

이를 이용하기 위해서는 SqlSessionFactory 인스턴스를 사용해야 하는데, 이 인스턴스가 DataSource를 참조하여 MyBatis와 DB 서버를 연동시켜준다.

MYBATIS 연결 테스트 캡처 화면



```
1 package com.comento.project;
2
3 import static org.junit.Assert.fail;
4
13
14 @RunWith(SpringJUnit4ClassRunner.class)
15 @ContextConfiguration(locations={"file:src/main/webapp/WEB-INF/spring/**/root-conte
16 public class MyBatisTest {
17     @Inject
18     private SqlSessionFactory sqlSessionFactory;
19
20     @Test
21     public void testFactory() {
22         System.out.println(sqlSessionFactory);
23     }
24
25     @Test
26     public void testSession() throws Exception {
27         try(SqlSession session = sqlSessionFactory.openSession()) {
28             System.out.println(session);
29         } catch (Exception e) {
30             fail(e.getMessage());
31         }
32     }
33 }
```

finished after 1.678 seconds

Runs: 2/2 Errors: 0 Failures: 0

MyBatisTest [Runner: JUnit 5] (1.505 s)

◀ Junit을 사용하여 MyBatis와 Oracle 서버가 제대로 연결되는지 테스트하는 모습

초록색 막대가 결과창에 뜨므로 SqlSessionFactory 객체 또한 제대로 만들어졌음을 확인할 수 있다.

느낀 점

새로운 프레임워크

스프링 프레임워크뿐만 아니라 Junit, MyBatis라는 프레임워크를 접했다. JSP와 일부 라이브러리만 다뤄봤는데 잘만 익히면 기존보다 훨씬 쉽고 빠르게 코딩할 수 있을 것 같다. 직접 프로젝트를 진행해봤으니 얼마나 코드 라인이 크게 줄어든 수 있는지 실감 났다.

테스트 프레임워크인 Junit은 비교적 간단하게 연결 테스트를 진행할 수 있어서 편리했다.

부족한 점

이번 과제를 진행하면서 부족하다고 느낀 점은 다음과 같았다. 먼저, 스프링 환경 설정을 위해 xml 파일을 다루면서 내가 프레임워크 간 버전을 크게 신경 안 쓰고 지냈음을 깨달았다. 기존에 정해 놓은 환경에서만 작업하기도 했고, 스프링과 사용하는 프레임워크를 잘 모르다보니 버전별 호환 가능성을 소홀하게 봤다. 버전 차이때문에 테스트에 몇 번 실패하며 난항을 겪었다.

또한, 깃허브 리포지토리에서 커밋하면서 rebase를 더 익혀야 겠다고 생각했다. 커밋 메시지를 수정하느라 시도해보았지만 브랜치 모양이 꼬이며 뜻대로 되지 않았다.



감사합니다.

Name	오혜진
Phone	010-9107-4667
E-Mail	absence_0@naver.com
GitHub Link	https://github.com/O-HyeJin/2022_spring.git