



## | 학습주제

- ✓ Spring Framework
- ✓ XML 기반의 DI
- ✓ 단위테스트를 통한 동작 확인

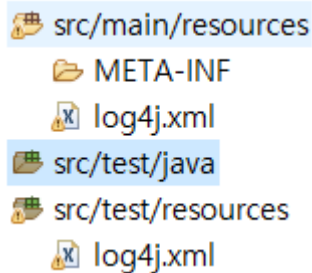
## | 학습목표

- ✓ spring legacy project 를 구성할 수 있다.
- ✓ pom.xml 을 편집하여 버전변경, 필요한 라이브러리 추가 등을 할 수 있다.
- ✓ XML 을 이용해서 스프링 환경을 구성할 수 있다.
- ✓ 단위테스트를 통해 코드의 동작을 테스트할 수 있다.

## | 지문

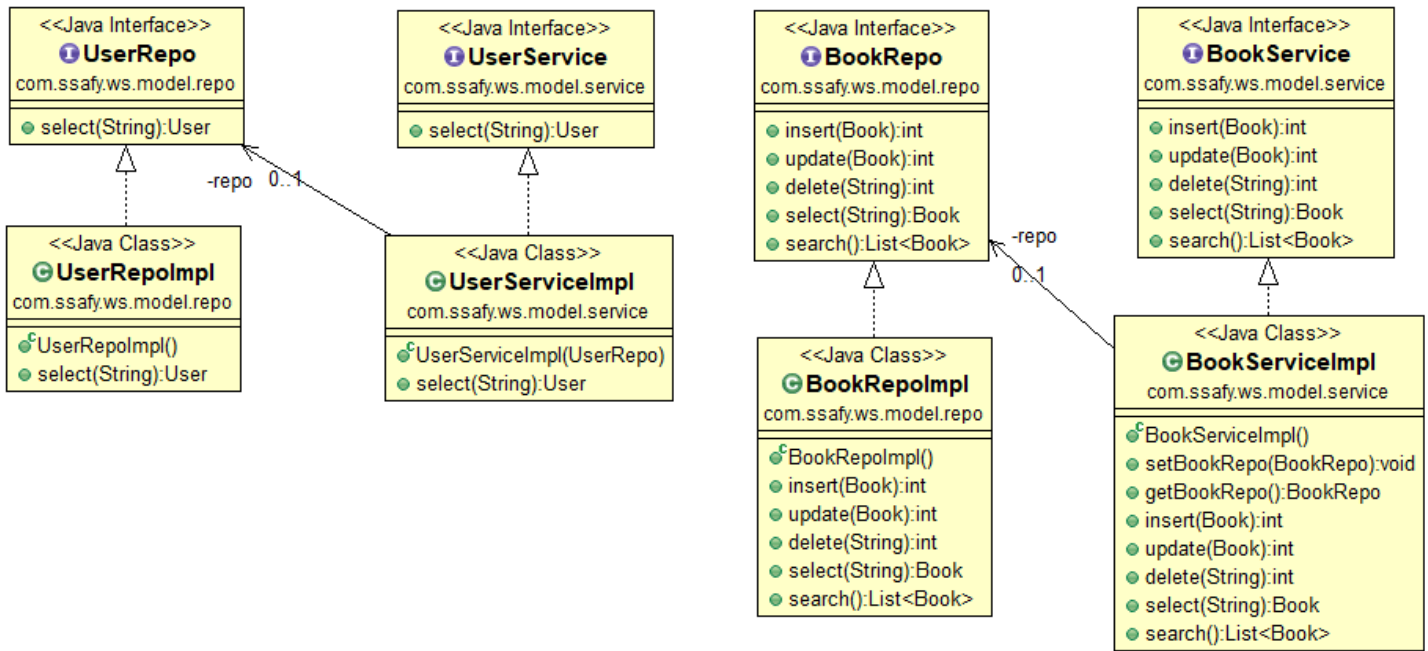
1. 도서 정보를 관리하기 위한 애플리케이션을 Spring Framework 기반으로 만들고자 한다. 아래 내용을 만족하도록 처리하시오. (기본적으로 환경 설정은 XML 기반으로 처리한다.)

- A. 도서 관리의 데이터베이스를 구축하기 위해 제공되는 ssafy\_spring\_ws.sql 을 실행한다.
- B. spring legacy 기반으로 프로젝트를 생성한다. Template 는 Spring MVC Project, 프로젝트 이름은 ws\_framework\_1 로 하고 기본 패키지는 com.ssafy.ws 로 한다.
- C. pom.xml 에서
  - i. spring framework 의 버전을 5.X, 사용하는 JRE 의 버전을 1.8 로 설정한다.
  - ii. junit 의 버전을 4.13.1 로 변경한다.
  - iii. spring-test, commons-dbc2, mysql-connector-java 를 추가한다.
- D. 디버그 레벨의 로그 출력을 위해 log4j.xml 에서 프로젝트의 기본 logger 레벨을 debug 로 변경한다.
  - i. Log4j.xml 은 두 군데에 있음을 주의한다.



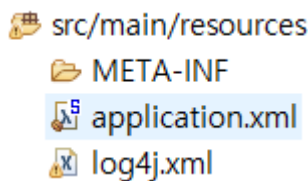
- E. ssafy\_spring\_ws schema 를 참고하여 DTO 인 User.java, Book.java 를 생성한다.
- F. 아래의 UML 을 참조하여 클래스들을 작성한다.
  - i. UserServiceImpl 은 UserRepo 를 has a 관계로 포함하는데 생성자를 통해서 주입 받는다.
  - ii. BookServiceImpl 은 BookRepo 를 has a 관계로 포함하는데 setter 메서드를 통해 주입 받는다.

# Framework : Spring 01. Spring DI 1



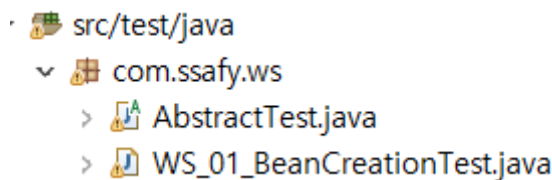
- iii. interface 에 선언된 메서드의 구현은 단지 자동 생성되는 코드면 충분하다. [자료-샘플코드]로 제공되는 UserRepo 와 UserRepoImpl 을 살펴보자.

G. 빈을 등록하기 위한 설정 파일로 application.xml 파일을 src/main/resources 에 작성한다.



H. application.xml 에서

- i. 위에서 작성한 UserRepoImpl, UserServiceImpl, BookRepoImpl, BookServiceImpl 을 빈으로 등록한다.
  - ii. 추가로 org.apache.commons.dbcp2.BasicDataSource 타입의 빈도 등록한다.
- I. src/test/java 에 WS\_01\_BeanCreationTest.java 파일을 만들고 아래 내용을 테스트한다.
- i. 테스트 환경의 재사용을 위해 AbstractTest 를 만들고 다른 테스트 클래스에서 상속받아서 사용하자.



- ii. BookRepo, UserRepo, BookService, UserService 가 잘 생성되었는가?
- iii. 컨텍스트에서 직접 얻어온 BookRepo 와 BookService 가 가지는 repo 가 동일한가?
- iv. DataSource 객체가 잘 생성되었는지 테스트하고 출력한다.



## | 자료 1: 샘플 코드(UserRepo.java)

```
1. package com.ssafy.ws.model.repo;
2.
3. import com.ssafy.ws.dto.User;
4.
5. public interface UserRepo {
6.     User select(String id);
```

## | 자료 2: 샘플 코드(UserRepoImpl.java)

```
1. package com.ssafy.ws.model.repo;
2.
3. import org.springframework.stereotype.Repository;
4.
5. import com.ssafy.ws.dto.User;
6.
7. @Repository
8. public class UserRepoImpl implements UserRepo {
9.
10.     @Override
11.     public User select(String id) {
12.         // TODO Auto-generated method stub
13.         return null;
14.     }
15. }
```



## | 자료 3: 프로젝트 전체 구조

- ws\_framework\_1
    - src/main/java
      - com.ssafy.ws
        - dto
          - Book.java
          - User.java
        - model
          - repo
            - BookRepo.java
            - BookRepoImpl.java
            - UserRepo.java
            - UserRepoImpl.java
          - service
            - BookService.java
            - BookServiceImpl.java
            - UserService.java
            - UserServiceImpl.java
      - src/main/resources
        - META-INF
        - application.xml
        - log4j.xml
      - src/test/java
        - com.ssafy.ws
          - AbstractTest.java
          - WS\_01\_BeanCreationTest.java
        - src/test/resources
        - Maven Dependencies
        - JRE System Library [JavaSE-1.8]
        - etc
        - src
        - target
        - pom.xml