

Spring Framework

1강_ 스프링의 개요

- 1-1. 스프링 프레임워크 모듈
- 1-2. 스프링 컨테이너
- 1-3. Maven 프로젝트 생성
- 1-4. pom.xml의 이해
- 1-5 스프링 프로젝트 맛보기

1-1: 스프링 프레임워크 개요

프레임워크? – 뼈대를 이루는 코드들의 묶음

- 프레임워크란 기본 뼈대 이다. 자동차를 만들 때 뼈대를 구현한 다음 만든다.
- 프레임워크는 방향성을 제시하고, 원하는 기능을 빠르게 만들 수 있게 함



스프링 프레임워크는 주요기능으로 **DI, AOP, MVC, JDBC** 등을 제공한다

모듈?

- 라이브러리
- 프레임워크안에 미리 만들어져 있는 기능

1-1: 스프링 프레임워크 특징

POJO(Plain Old Java Object) 기반의 프레임워크

자바 객체의 라이프사이클을 **스프링 컨테이너가 직접 관리**하며, 스프링 컨테이너로부터 필요한 객체를 얻어올 수 있습니다.

DI(Dependency Injection)을 지원

각 계층이나 서비스 들 사이 또는 **객체들 사이에 의존성**이 존재할 경우 스프링 프레임워크가 서로를 연결시켜줍니다. 이는 클래스들 사이에 약한 결합을 가능케 합니다.

AOP(Aspect Oriented Programming)를 지원

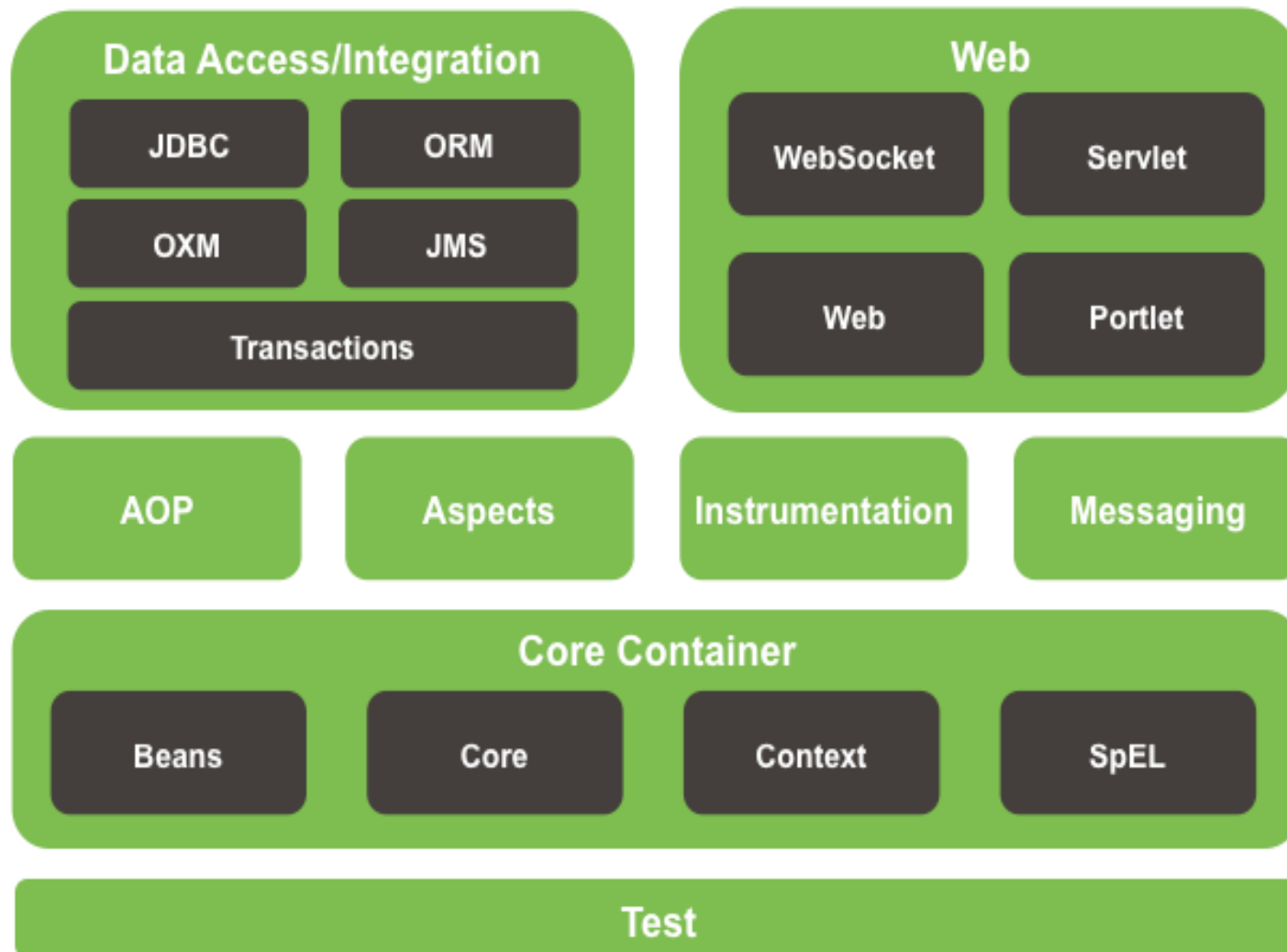
트랜잭션, 로깅, 보안 등 여러 모듈에서 **공통적으로 지원하는 기능을 분리**하여 사용할 수 있습니다.

확장성이 높다.

스프링 프레임워크의 소스는 모두 라이브러리로 분리시켜 놓음으로써 **필요한 라이브러리만 가져다 쓸 수 있습니다**. 그리고 많은 외부 라이브러리들도 이미 스프링 프레임워크와 연동되고 있습니다.

Model2 방식의 MVC Framework를 지원

1-1: 스프링 프레임워크 런타임



1-1: 스프링 프레임워크 모듈

스프링에서 프레임워크에서 제공하고 있는 모듈

스프링 모듈	기능
spring-core	스프링의 핵심인 DI(Dependency Injection)와 IoC(Inversion of Control)를 제공
spring-aop	AOP구현 기능 제공
spring-jdbc	데이터베이스를 쉽게(적은 양의 코드) 다룰 수 있는 기능 제공
spring-tx	스프링에서 제공하는 트랜잭션 관련 기능 제공
spring-webmvc	스프링에서 제공하는 컨트롤러(Controller)와 뷰(View)를 이용한 스프링MVC 구현 기능제공

스프링 프레임워크에서 제공하고 있는 모듈(라이브러리)을 사용하려면, 모듈에 대한 의존설정을 개발 프로젝트에 XML 파일등을 이용해 서 개발자가 직접 하면 된다.

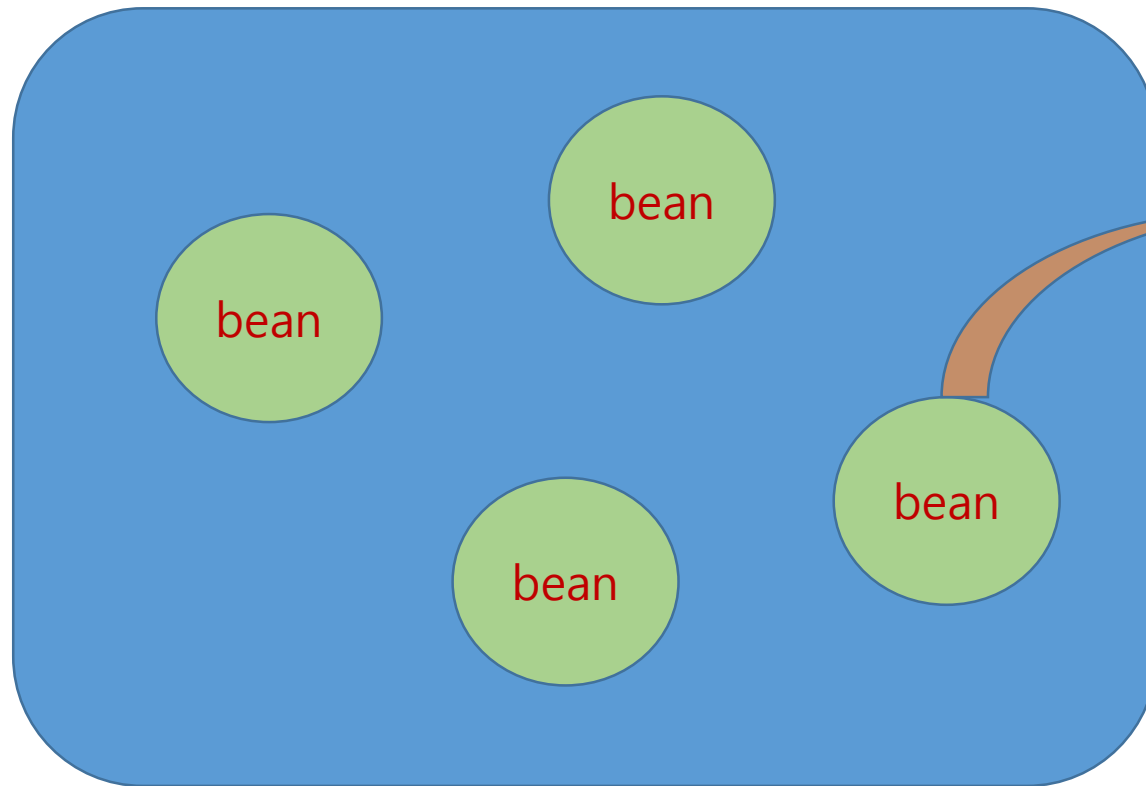
1-2 :스프링 컨테이너(IoC)

스프링에서 객체를 생성하고 조립하는 **컨테이너(container)**로,
컨테이너를 통해 생성된 객체를 **빈(Beans)**이라고 부른다.



1-2 :스프링 컨테이너(IoC)

스프링 IoC컨테이너



IoC : Inversion of Control

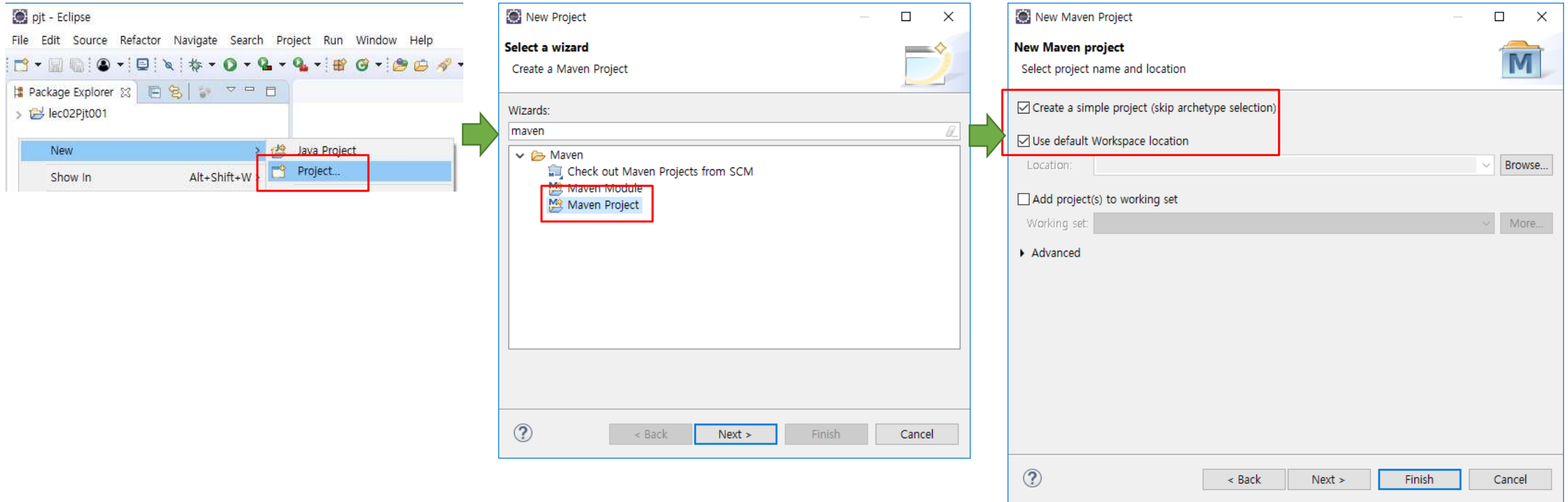
- 제어의 역전
- 객체를 필요할 때 생성해서 사용하던 방식을
- 미리 생성해 놓고 꺼내서 사용하는 형태로 변경

사용

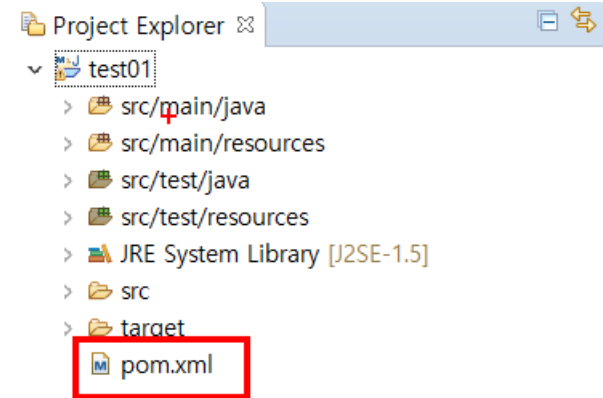
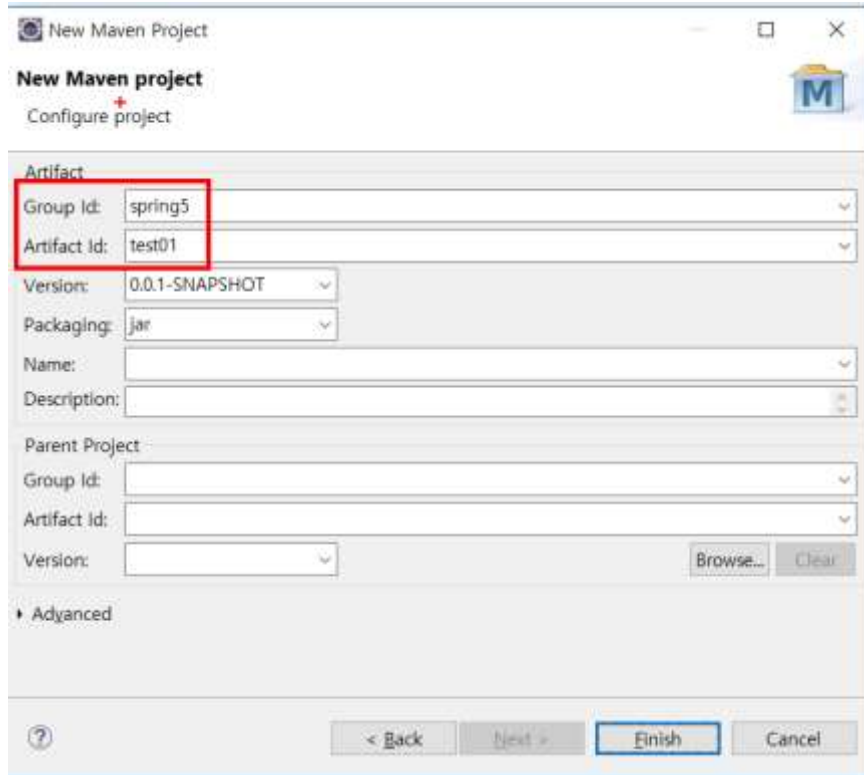
1-3: Maven프로젝트 생성

Maven

필요한 라이브러리를 특정 문서(**pom.xml**)에 정의해 놓으면 네트워크를 통해서 라이브러리들을 자동으로 다운받아줍니다.



1-3: Maven프로젝트 생성

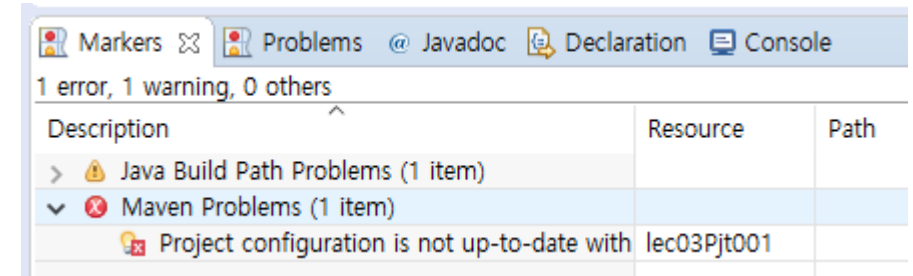


1-3: pom.xml에 다음과 같이 기술

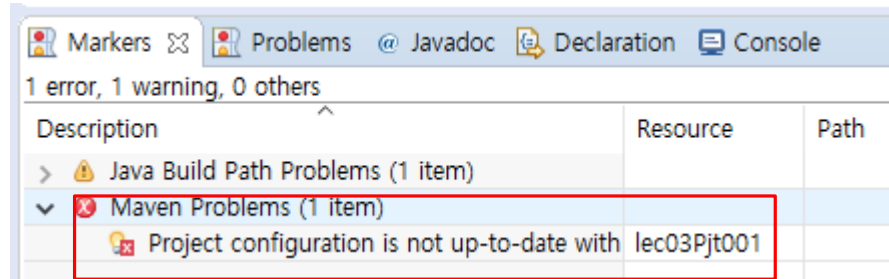
외부에 존재하는 Spring라이브러리를 메이븐으로 다운로드 하는 코드

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.0.7.RELEASE</version>
  </dependency>
</dependencies>
```

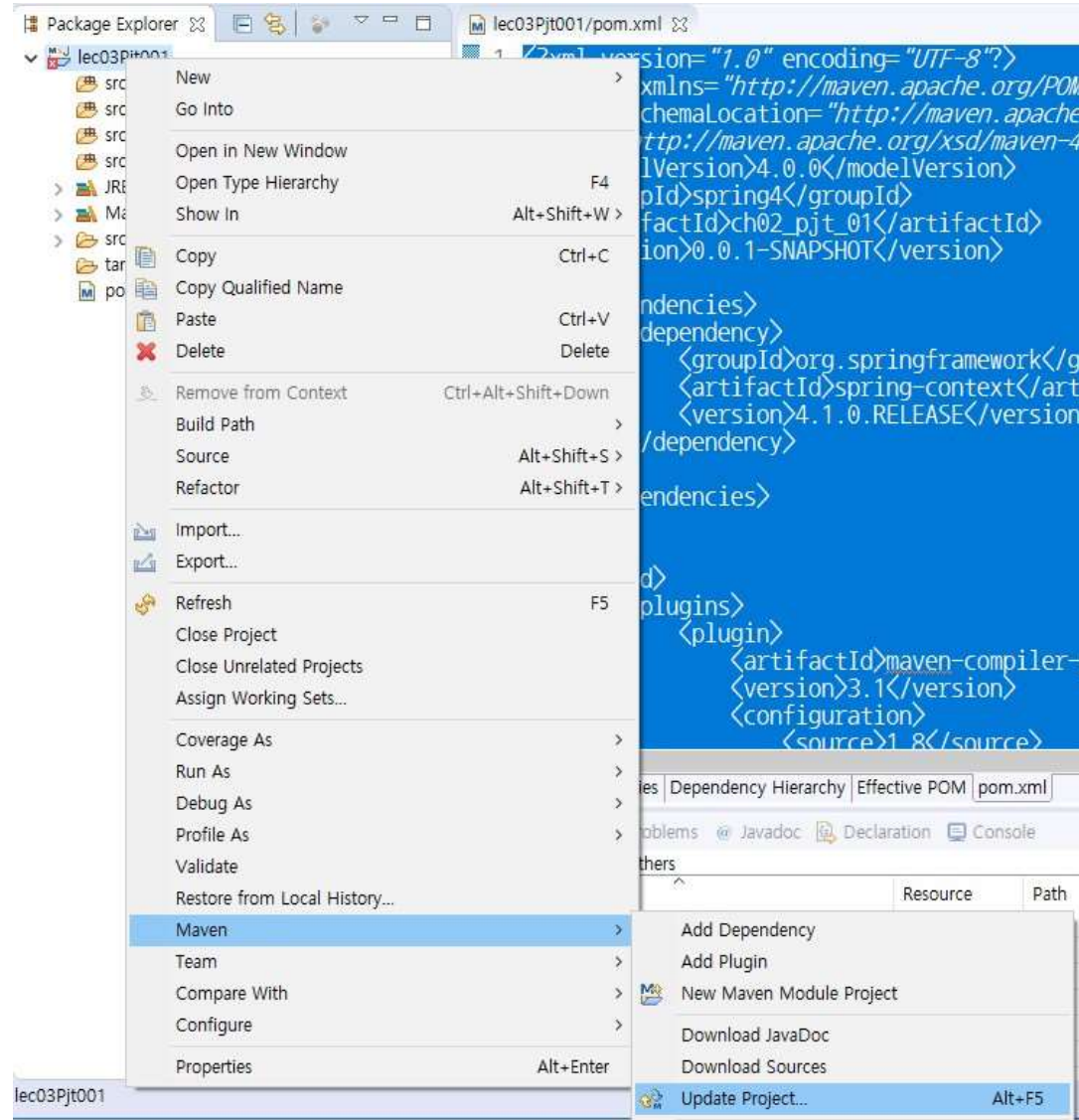
```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.5.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>utf-8</encoding>
      </configuration>
    </plugin>
  </plugins>
</build>
```



1-3 : pom.xml 작성



프로젝트의 JRE라이브러리 버전이 메이븐 설정파일에 명시되어 있는 버전과 일치하지 않아서 발생하는 것으로 프로젝트를 업데이트하라고 나온다.

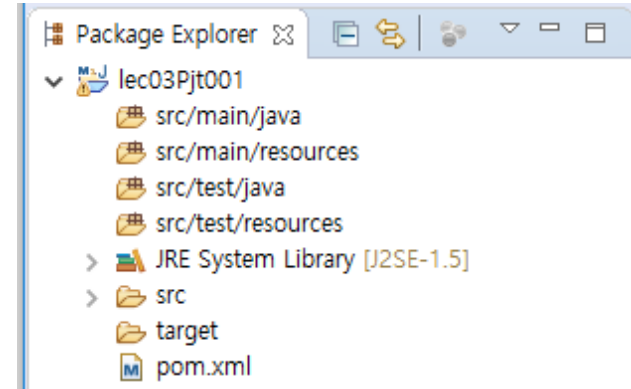


1-4 : 폴더 및 pom.xml 파일의 이해

프로젝트 : 스프링 프로젝트 Root.

/src/main/java 폴더 : .java파일 관리.

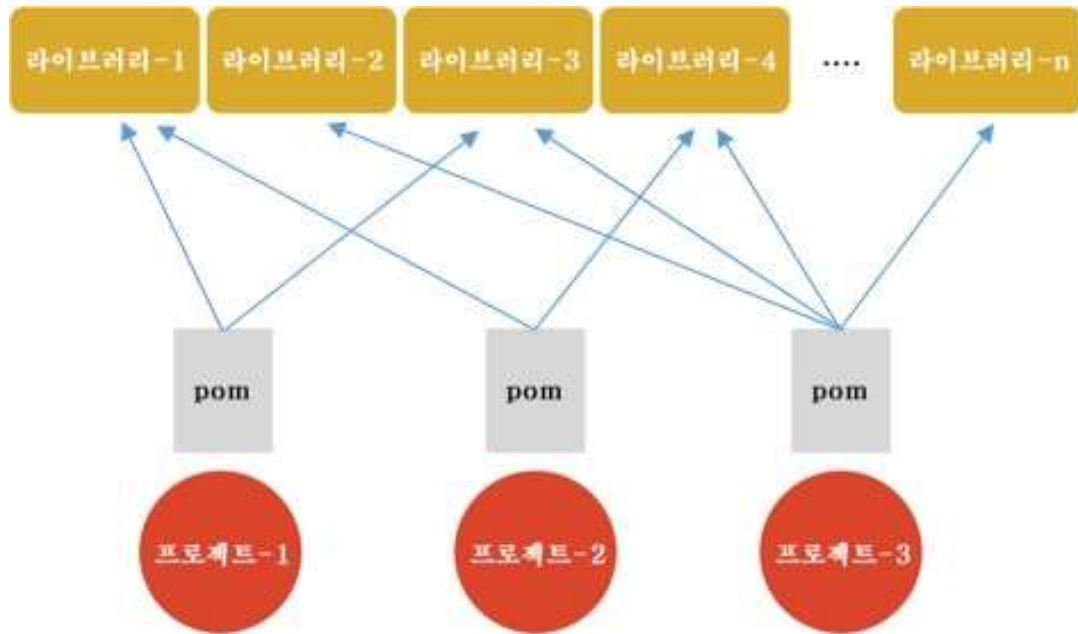
/src/main/resources 폴더 : 자원파일 관리.



1. **java** 폴더(lec03Pjt001/src/main/java)의 경우 특별한 것은 없고, 앞으로 만들어지는 **자바 파일들이** 관리되는 폴더이다.
2. **resources** 폴더(lec03Pjt001/src/main/resources)의 경우 자원을 관리하는 폴더로 **스프링 설정 파일(XML)** 또는 **프로퍼티 파일** 등이 관리된다.
3. **java, resources** 폴더는 스프링 프레임워크의 기본 구조를 이루는 폴더로 개발자는 이대로 폴더를 구성해야 한다.

1-4 : 폴더 및 pom.xml 파일의 이해

pom.xml 파일은 메이븐 설정파일로 메이븐은 라이브러리를 연결해주고, 빌드를 위한 플랫폼이다.
-> 한마디로 메이븐을 이용하면 라이브러리를 설정으로 바로 사용 할 수 있다



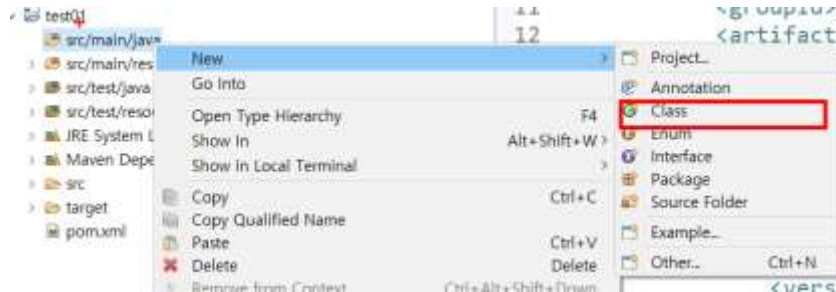
라이브러리는 아래의 경로에서 확인 가능하다.
C:\Users\사용자\.m2\repository\org\springframework

pom.xml에 의해서 필요한 라이브러리만 다운로드 해서 사용한다

※ 참고로 모듈의 라이브러리 파일명은 artifactId + “-” + 버전명 + “.jar” 로 표시된다.

1-5 : 스프링 프로젝트 맛보기

-Java 파일을 이용한 프로젝트 실행



```
package test01;

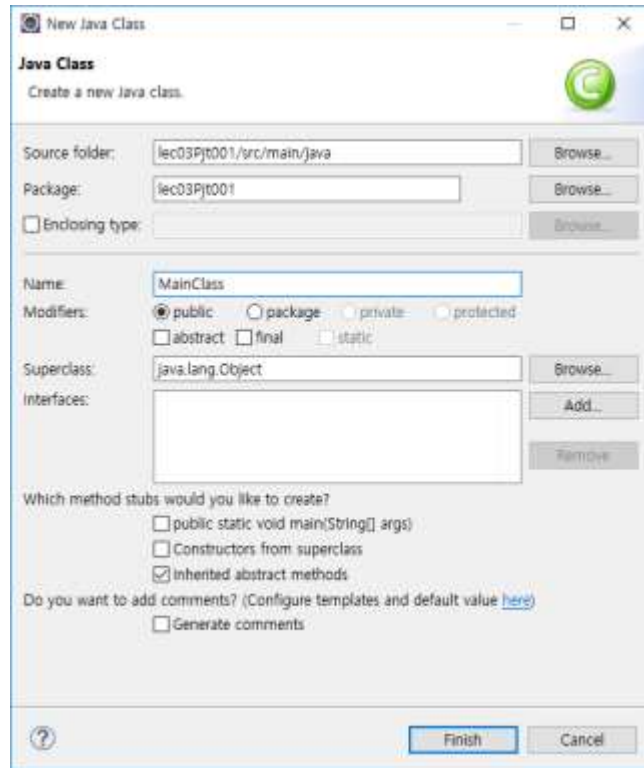
public class SpringTest {

    public void method() {
        System.out.println("스프링 프로젝트 맛보기");
    }

}
```

1-5 : 스프링 프로젝트 맛보기

-Java 파일을 이용한 프로젝트 실행



```
1 package test01;
2
3 public class MainClass {
4
5     public static void main(String[] args) {
6
7         SpringTest st = new SpringTest();
8         st.method();
9     }
10 }
11
```



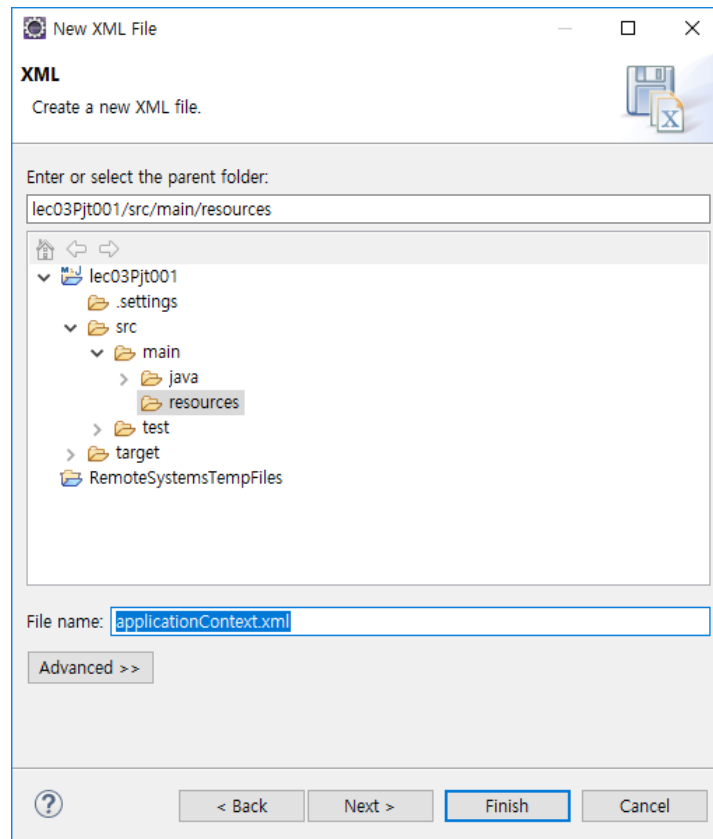
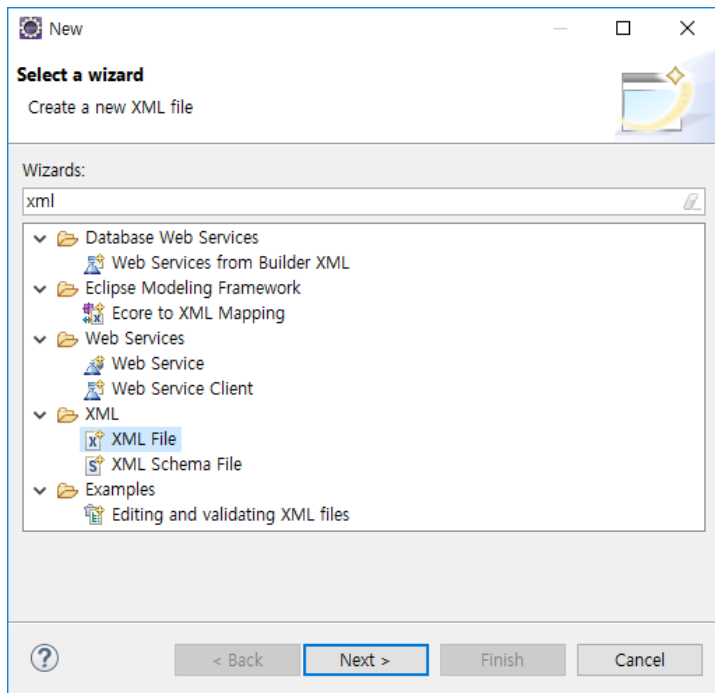
Markers Properties Servers Data Source Explorer Snippets Console

<terminated> MainClass [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe
스프링 프로젝트 맛보기

1-5 : 스프링 프로젝트 맛보기

-Java 파일을 이용한 프로젝트 실행

스프링 방식의 '의존'을 이용하기 위해서는 Main에서 SpringTest 객체를 직접 생성하지 않고, **스프링 설정 파일(XML)**을 이용해보기로 한다. 가장 큰 차이점은 **Java 파일에서 이용한 new 연산자를 이용하지 않고 스프링 설정파일(XML)을 이용하는 것이다.**



1-5 : 스프링 프로젝트 맛보기

-스프링 설정파일(xml)에 객체를 생성해 두고, 스프링 설정파일을 호출해서 객체를 호출

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="good" class="test01.SpringTest" />

</beans>
```

MainClass.java

```
1 package test01;
2
3 import org.springframework.context.support.GenericXmlApplicationContext;
4
5 public class MainClass {
6
7     public static void main(String[] args) {
8
9         // SpringTest st = new SpringTest();
10        // st.method();
11
12        GenericXmlApplicationContext ct =
13            new GenericXmlApplicationContext("classpath:applicationContext.xml");
14
15        SpringTest test = ct.getBean("good", SpringTest.class);
16        test.method();
17
18        ct.close();
19    }
20 }
```

Markers Properties Servers Data Source Explorer Snippets

<terminated> MainClass [Java Application] C:\Program Files\Java\jre1.8.0_...

3월 12, 2019 1:55:34 오전 org.springframework.c
정보: Loading XML bean definitions from c
3월 12, 2019 1:55:34 오전 org.springframework.c
정보: Refreshing org.springframework.cont
스프링 프로젝트 맛보기