SAMSUNG
SW
ACADEMY
FOR
YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

Spring Framework

SpringBoot

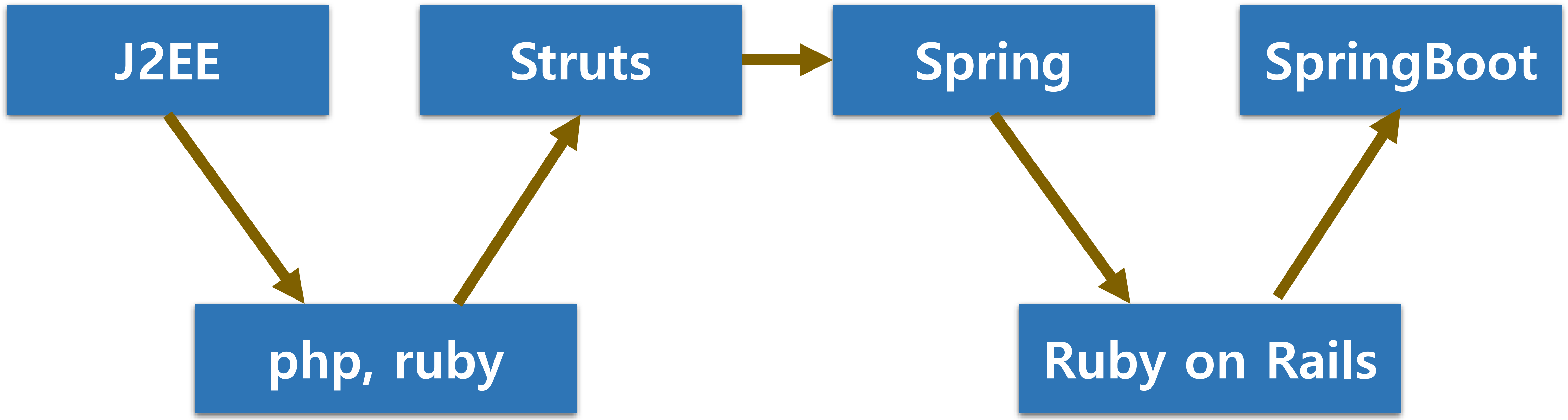
- SpringBoot
- SpringBoot 게시판

SAMSUNG SW ACADEMY FOR YOUTH

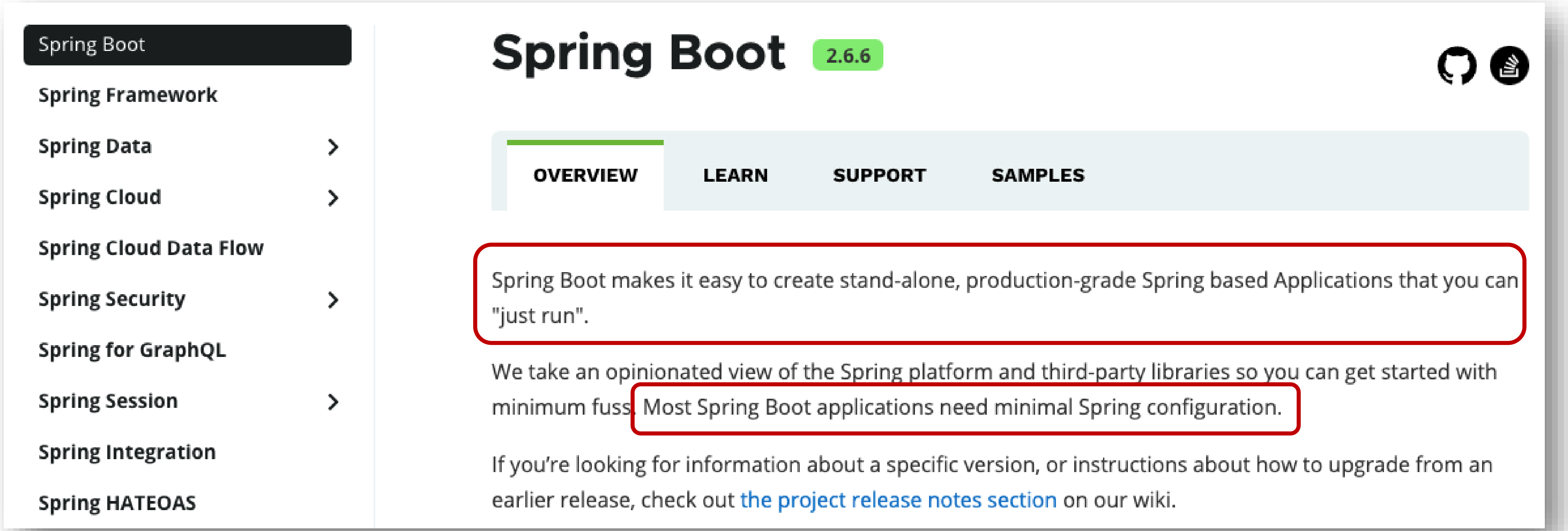
함께가요 미래로! Enabling People

SpringBoot

SpringBoot의 탄생



SpringBoot란



Spring Boot 2.6.6

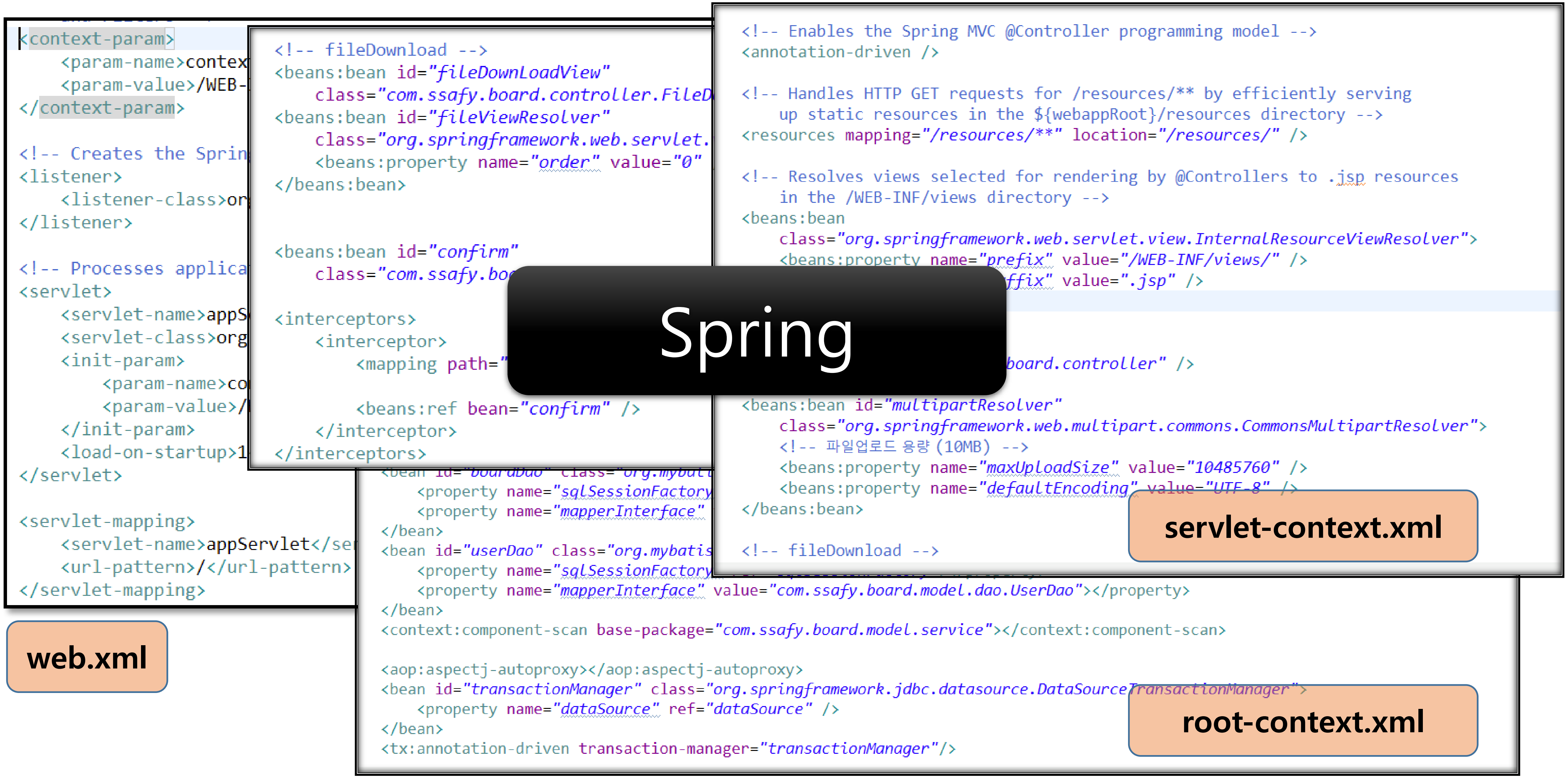
Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

If you're looking for information about a specific version, or instructions about how to upgrade from an earlier release, check out the project release notes section on our wiki.

https://spring.io/projects/spring-boot

SpringBoot



```
<context-param>
    <param-name>contex
    <param-value>/WEB-
</context-param>

<!-- Creates the Sprin
<listener>
    <listener-class>or
</listener>

<!-- Processes applica
<servlet>
    <servlet-name>appS
    <servlet-class>org
    <init-param>
        <param-name>co
        <param-value>/
    </init-param>
    <load-on-startup>1
</servlet>

<servlet-mapping>
    <servlet-name>appServlet</se
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

web.xml

```
<!-- fileDownload -->
<beans:bean id="fileDownLoadView"
    class="com.ssafy.board.controller.FileD
<beans:bean id="fileViewResolver"
    class="org.springframework.web.servlet.
    <beans:property name="order" value="0"
</beans:bean>


<beans:bean id="confirm"
    class="com.ssafy.boa

<interceptors>
    <interceptor>
        <mapping path="

        <beans:ref bean="confirm" />
    </interceptor>
</interceptors>
```

Spring

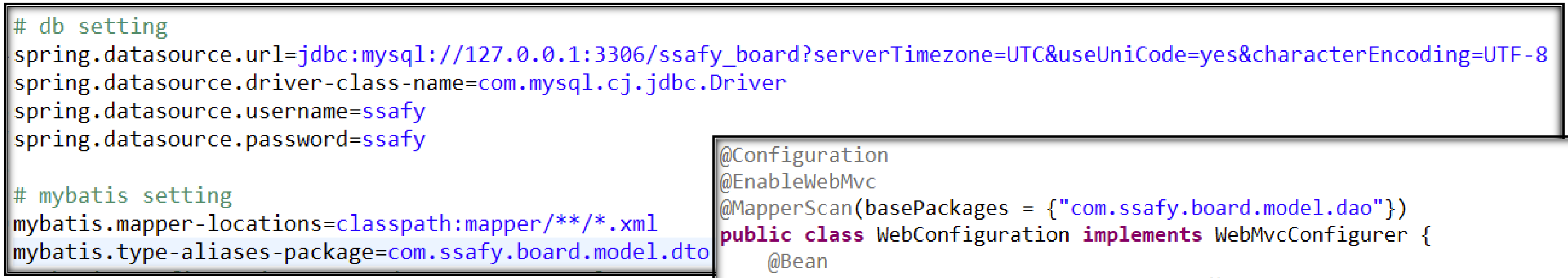
```
<!-- Enables the Spring MVC @Controller programming model -->
<annotation-driven />

<!-- Handles HTTP GET requests for /resources/** by efficiently serving
    up static resources in the ${webappRoot}/resources directory -->
<resources mapping="/resources/**" location="/resources/" />

<!-- Resolves views selected for rendering by @Controllers to .jsp resources
    in the /WEB-INF/views directory -->
<beans:bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
                              ffix" value=".jsp" />

                              board.controller" />

<beans:bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <!-- 파일업로드 용량 (10MB) -->
    <beans:property name="maxUploadSize" value="10485760" />
    <beans:property name="defaultEncoding" value="UTF-8" />
</beans:bean>

<!-- fileDownload -->
```

servlet-context.xml

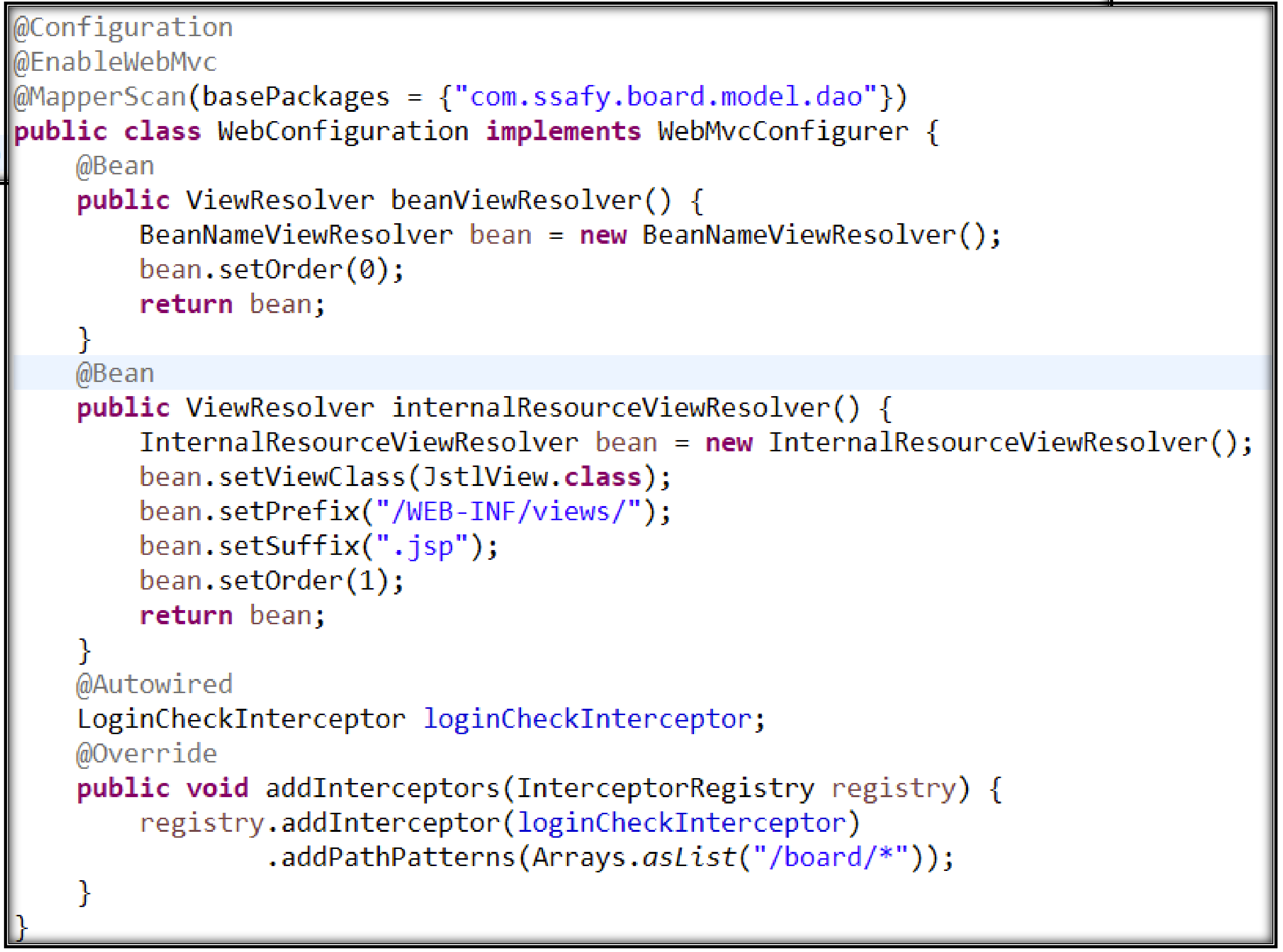
```
<bean id="boardDao" class="org.mybati
    <property name="sqlSessionFactory
    <property name="mapperInterface"
</bean>
<bean id="userDao" class="org.mybatis
    <property name="sqlSessionFactory
    <property name="mapperInterface" value="com.ssafy.board.model.dao.UserDao"></property>
</bean>
<context:component-scan base-package="com.ssafy.board.model.service"></context:component-scan>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource" />
</bean>
<tx:annotation-driven transaction-manager="transactionManager"/>
```

root-context.xml

SpringBoot



```
# db setting
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/ssafy_board?serverTimezone=UTC&useUniCode=yes&characterEncoding=UTF-8
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username=ssafy
spring.datasource.password=ssafy

# mybatis setting
mybatis.mapper-locations=classpath:mapper/**/*.xml
mybatis.type-aliases-package=com.ssafy.board.model.dto
```



```
@Configuration
@EnableWebMvc
@MapperScan(basePackages = {"com.ssafy.board.model.dao"})
public class WebConfiguration implements WebMvcConfigurer {
    @Bean
    public ViewResolver beanViewResolver() {
        BeanNameViewResolver bean = new BeanNameViewResolver();
        bean.setOrder(0);
        return bean;
    }
    @Bean
    public ViewResolver internalResourceViewResolver() {
        InternalResourceViewResolver bean = new InternalResourceViewResolver();
        bean.setViewClass(JstlView.class);
        bean.setPrefix("/WEB-INF/views/");
        bean.setSuffix(".jsp");
        bean.setOrder(1);
        return bean;
    }
    @Autowired
    LoginCheckInterceptor loginCheckInterceptor;
    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(loginCheckInterceptor)
                .addPathPatterns(Arrays.asList("/board/*"));
    }
}
```

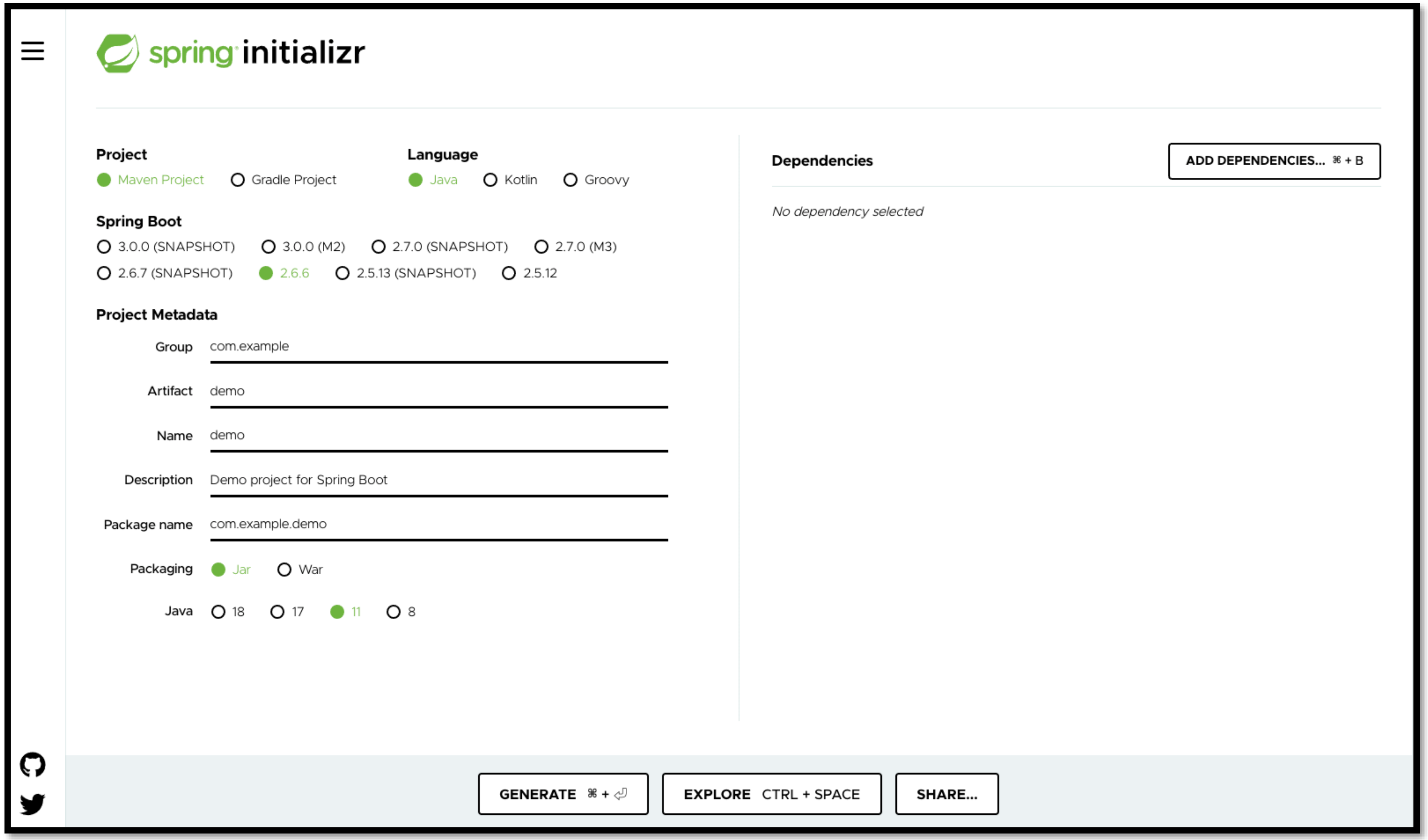
SpringBoot

SpringBoot

프로젝트 생성

SpringBoot Project 생성

https://start.spring.io/



Project
[✓ Maven Project] Gradle Project

Language
[✓ Java] Kotlin Groovy

Spring Boot
3.0.0 (SNAPSHOT) 3.0.0 (M2) 2.7.0 (SNAPSHOT) 2.7.0 (M3)
2.6.7 (SNAPSHOT) [✓ 2.6.6] 2.5.13 (SNAPSHOT) 2.5.12

Project Metadata

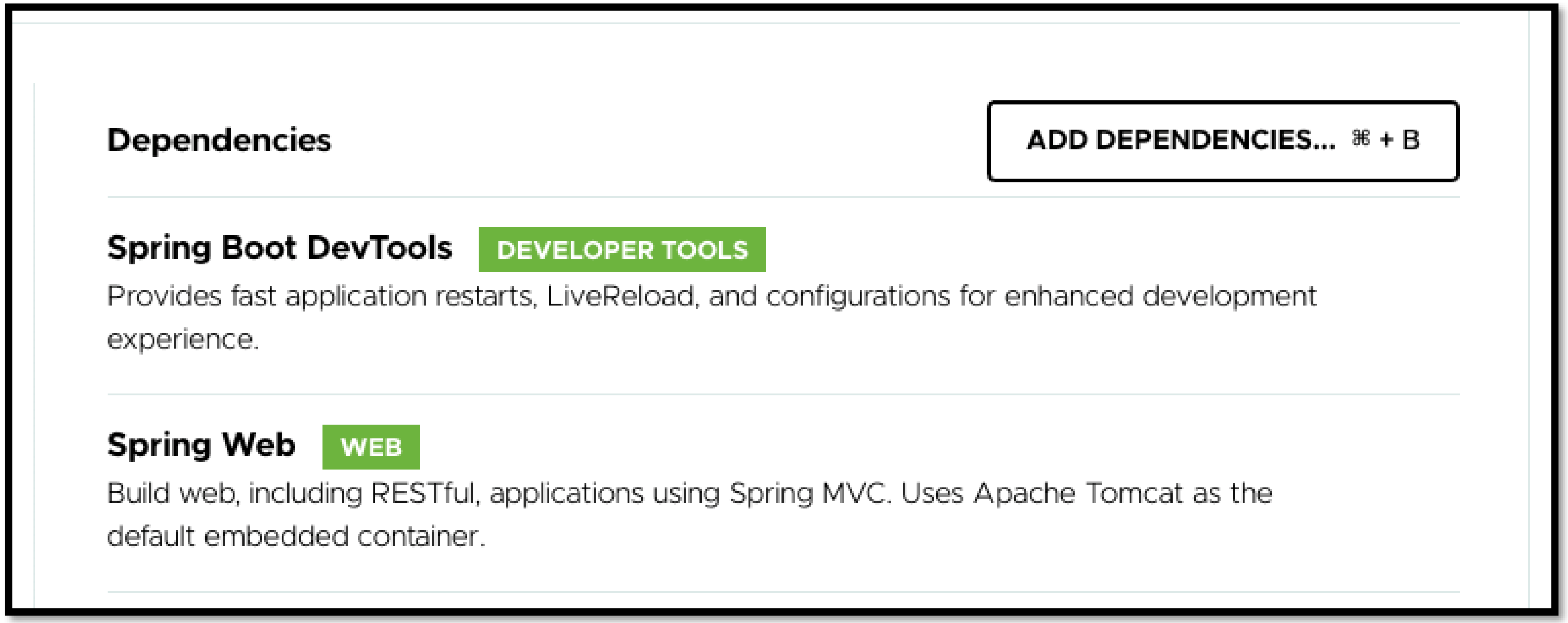
Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	[✓ Jar] War
Java	18 17 [✓ 11] 8

Dependencies

No dependency selected

SpringBoot Project 생성

- Dependencies 추가
- DevTools, Web …



Dependencies

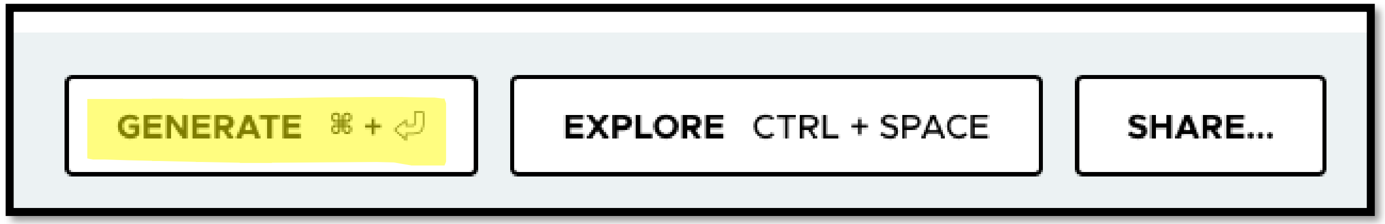
Spring Boot DevTools DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

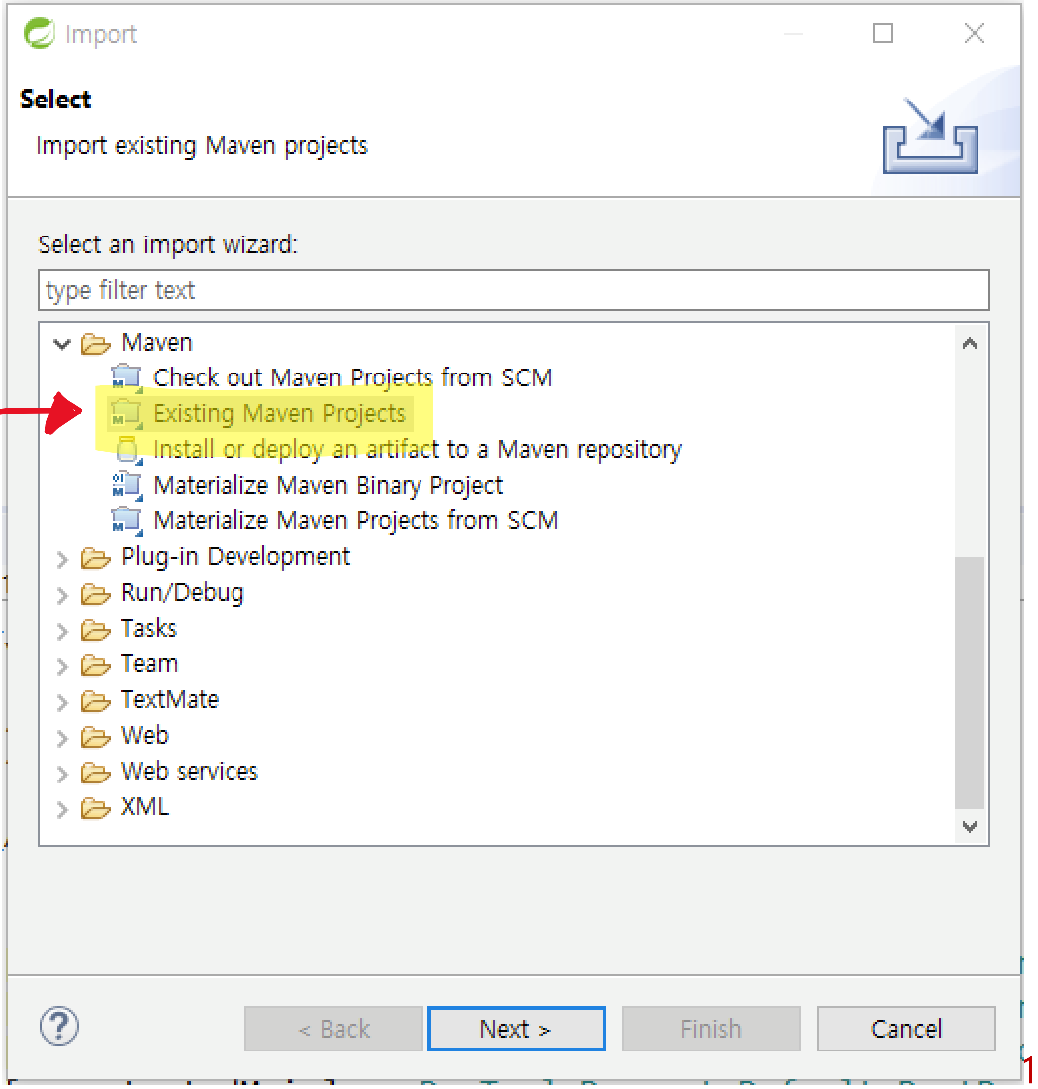
Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

SpringBoot Project 생성

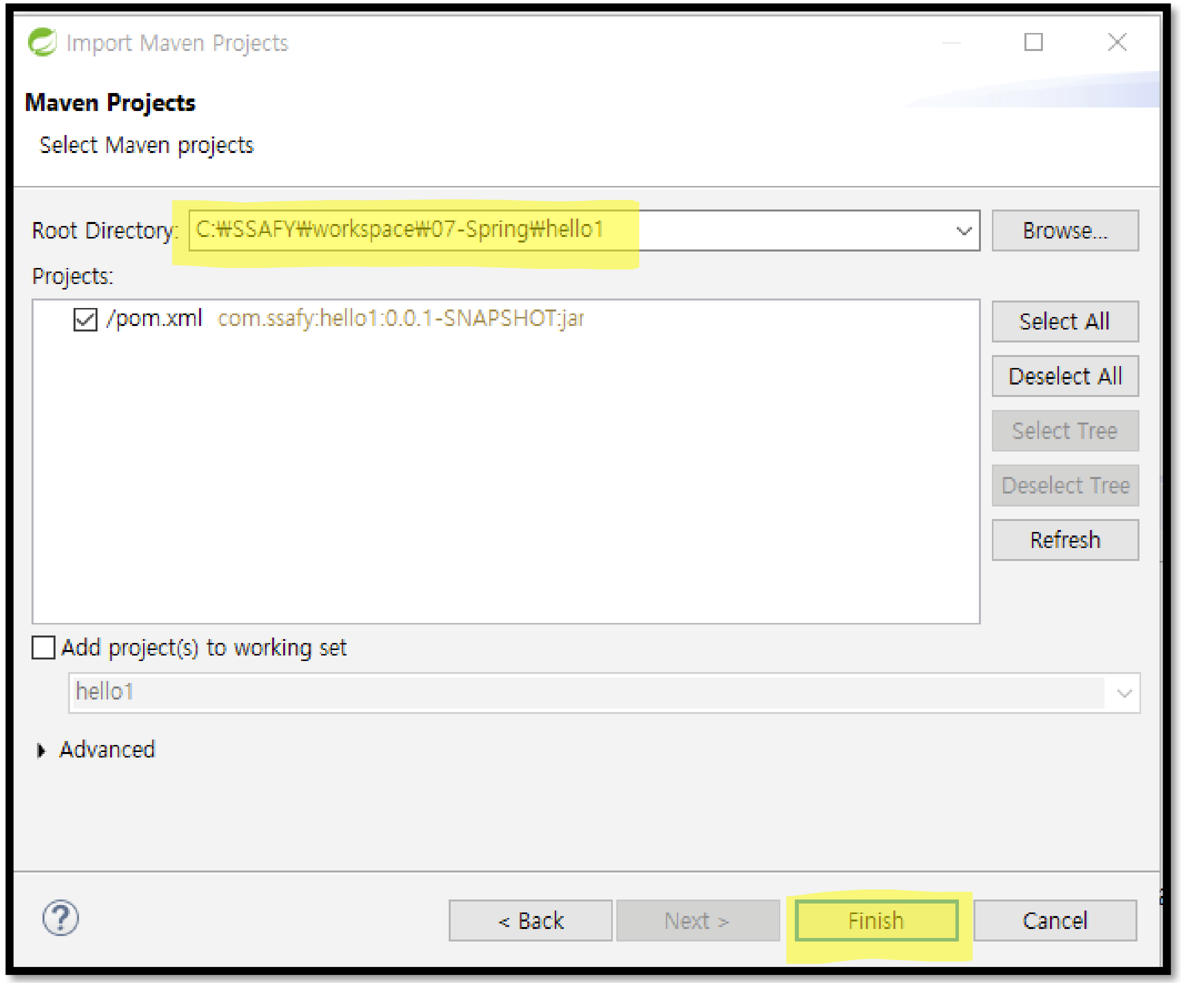


- 파일 압축 해제 후 import


Import
Select
Import existing Maven projects
Select an import wizard:
type filter text
Maven
Check out Maven Projects from SCM
Existing Maven Projects
Install or deploy an artifact to a Maven repository
Materialize Maven Binary Project
Materialize Maven Projects from SCM
Plug-in Development
Run/Debug
Tasks
Team
TextMate
Web
Web services
XML
< Back
Next >
Finish
Cancel

SpringBoot Project 생성

- 압축 해제 폴더 선택 후 Finish



Import Maven Projects

Maven Projects

Select Maven projects

Root Directory: C:₩SSAFY₩workspace₩07-Spring₩hello1

Projects:

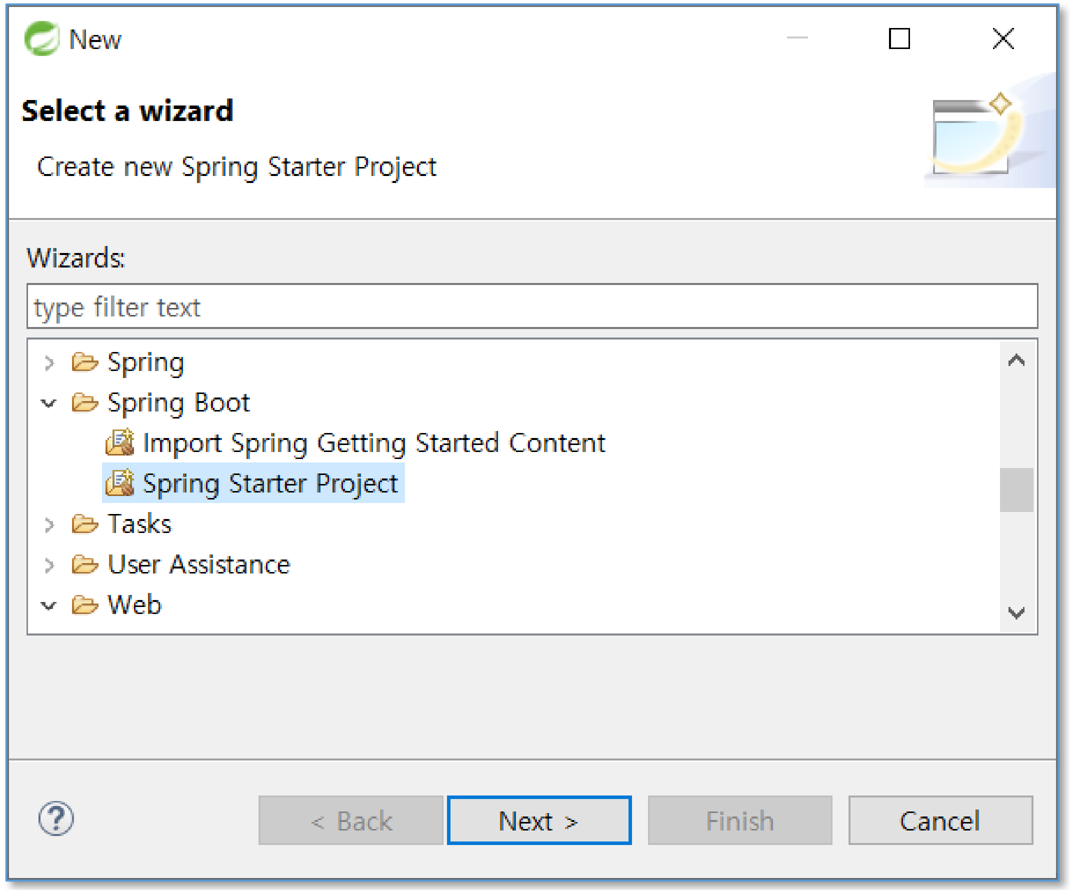
☑ /pom.xml com.ssafy:hello1:0.0.1-SNAPSHOT:jar

☐ Add project(s) to working set

hello1

SpringBoot Project 생성 (STS)

- 기존의 Spring과는 다르게 SpringBoot는 Spring Starter Project를 이용하여 윈도우의 Install Wizard와 같이 손쉽게 SpringBoot기반의 프로젝트를 만들 수 있다.


New

Select a wizard

Create new Spring Starter Project

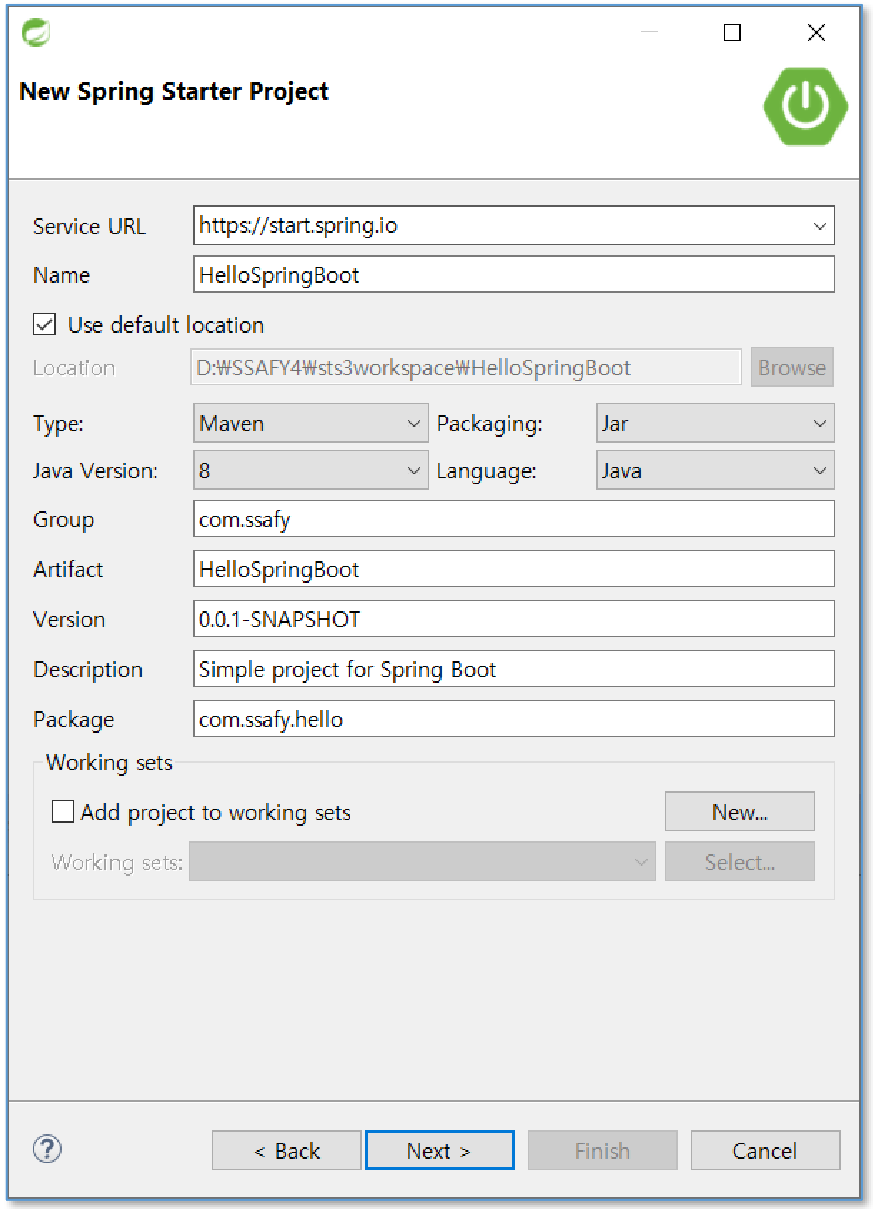
Wizards:

type filter text

- Spring
- Spring Boot
  - Import Spring Getting Started Content
  - Spring Starter Project
- Tasks
- User Assistance
- Web

SpringBoot Project 생성 (STS)

- 그림과 같이 설정 후 next



New Spring Starter Project

Service URL	https://start.spring.io
Name	HelloSpringBoot

☑ Use default location

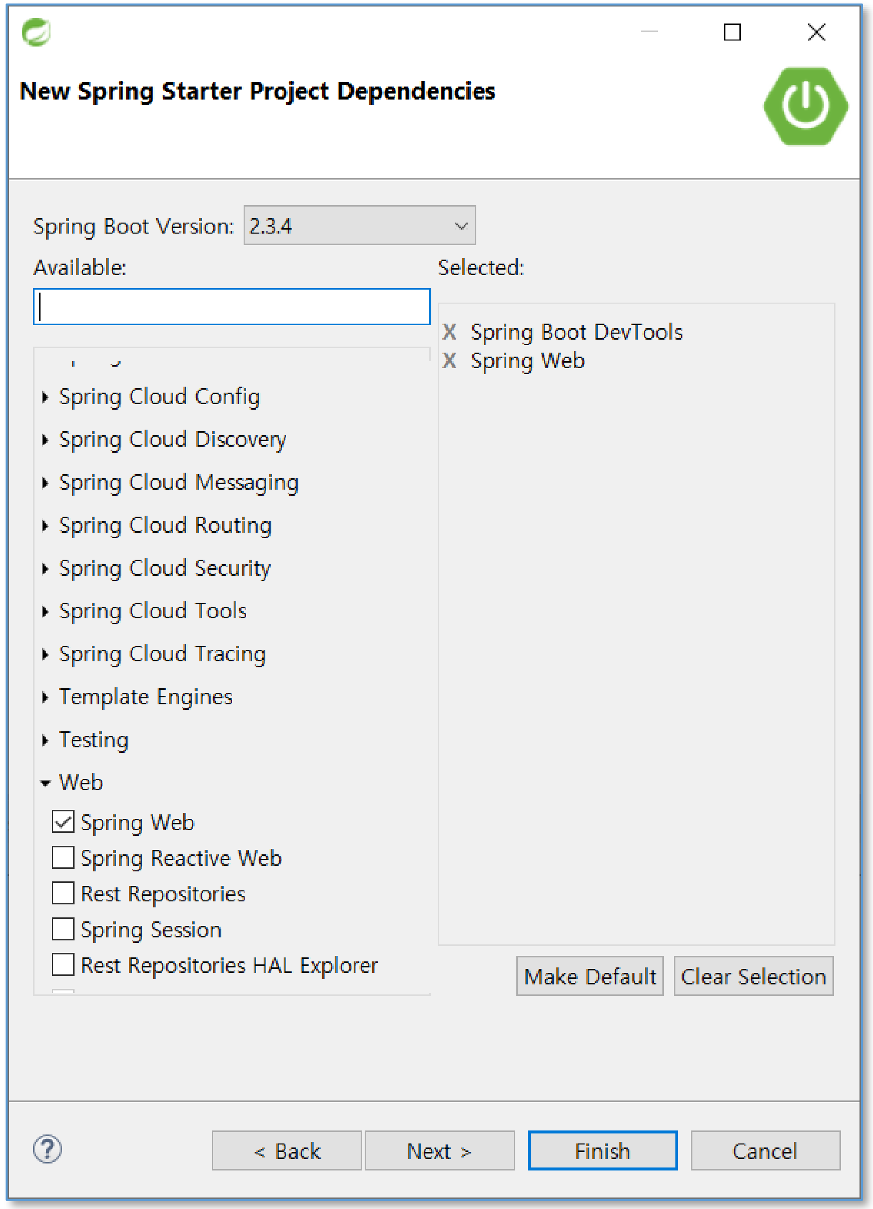
Location	D:₩SSAFY4₩sts3workspace₩HelloSpringBoot
Type:	Maven
Packaging:	Jar
Java Version:	8
Language:	Java
Group	com.ssafy
Artifact	HelloSpringBoot
Version	0.0.1-SNAPSHOT
Description	Simple project for Spring Boot
Package	com.ssafy.hello

Working sets

☐ Add project to working sets

SpringBoot Project 생성 (STS)

- SpringBoot의 버전 및 Dependency 설정


New Spring Starter Project Dependencies

Spring Boot Version: 2.3.4

Available:

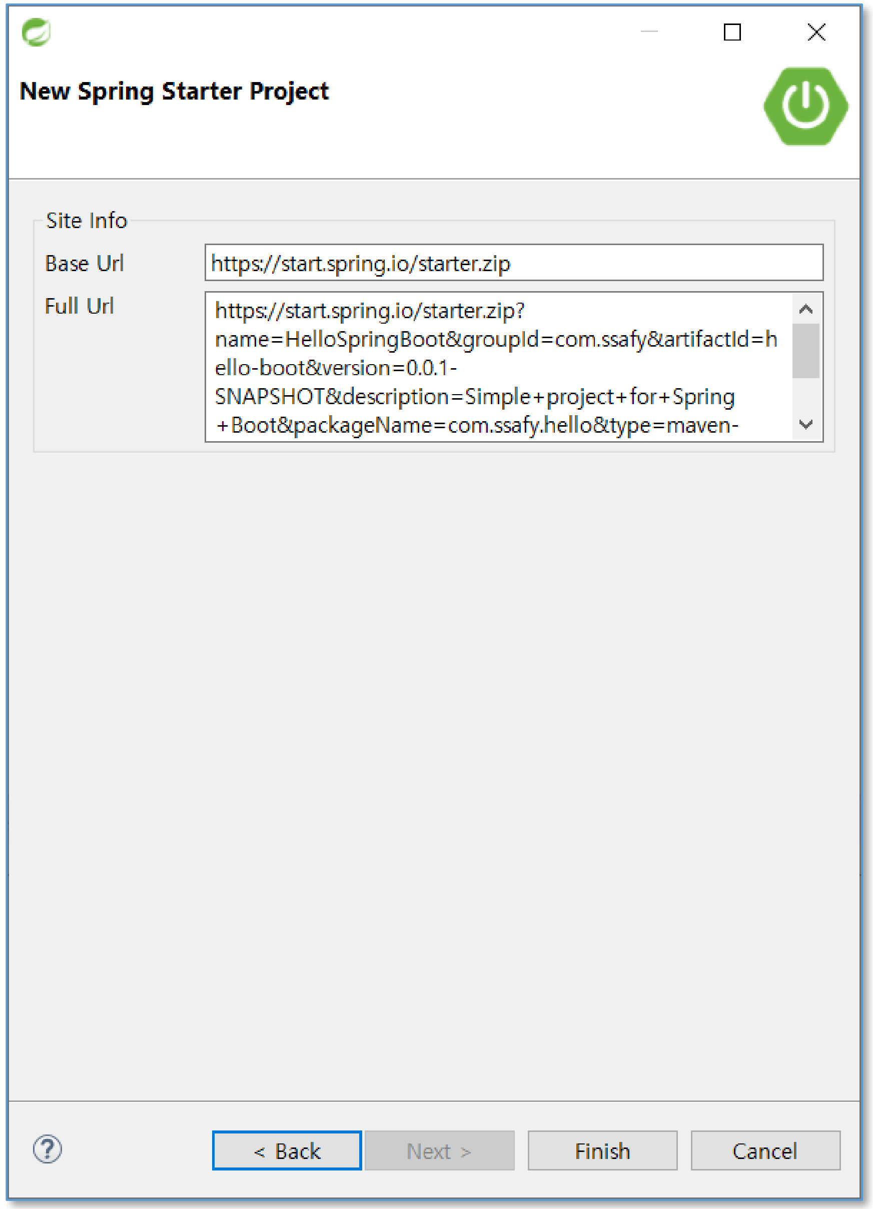
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Messaging
- Spring Cloud Routing
- Spring Cloud Security
- Spring Cloud Tools
- Spring Cloud Tracing
- Template Engines
- Testing
- Web
  - [✓] Spring Web
  - [ ] Spring Reactive Web
  - [ ] Rest Repositories
  - [ ] Spring Session
  - [ ] Rest Repositories HAL Explorer

Selected:

- X Spring Boot DevTools
- X Spring Web

SpringBoot Project 생성 (STS)

- 확인 후 Finish


New Spring Starter Project

Site Info

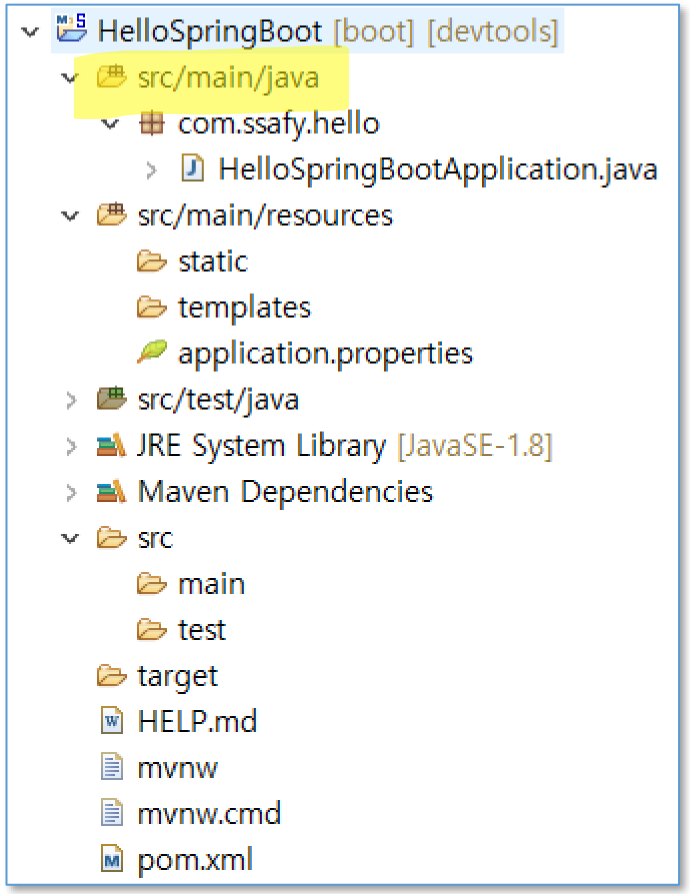
Base Url: https://start.spring.io/starter.zip

Full Url:

```
https://start.spring.io/starter.zip?
name=HelloSpringBoot&groupId=com.ssafy&artifactId=h
ello-boot&version=0.0.1-
SNAPSHOT&description=Simple+project+for+Spring
+Boot&packageName=com.ssafy.hello&type=maven-
```

SpringBoot Project 구조 (STS)

- project 생성 구조 및 주요 구성 폴더/파일

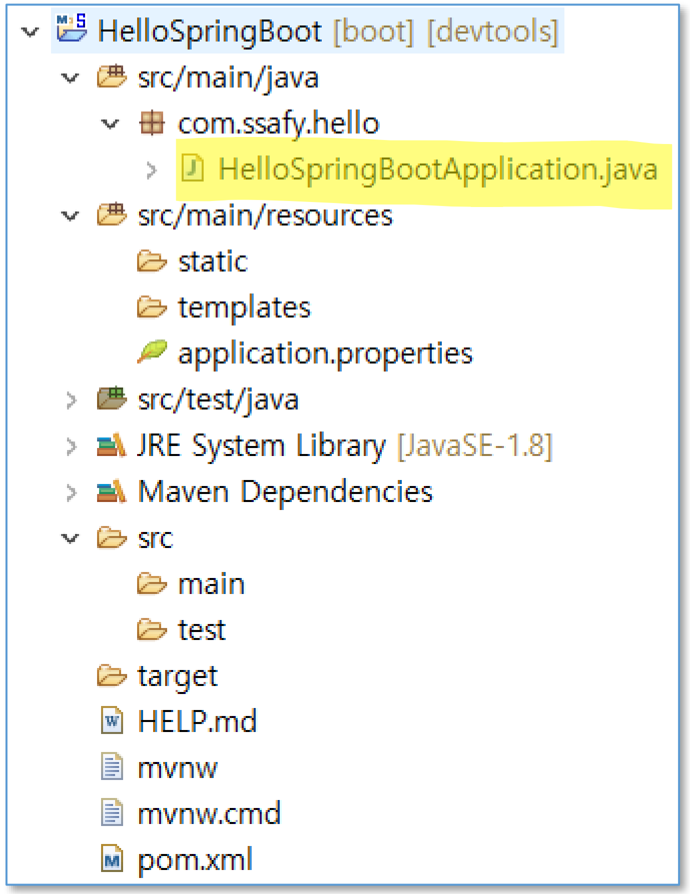


```
HelloSpringBoot [boot] [devtools]
  src/main/java
    com.ssafy.hello
      HelloSpringBootApplication.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

프로젝트의 주요 파일	설명
src/main/java	java source directory
HelloSpringBootApplication.java	application을 시작할 수 있는 main method가 존재하는 스프링 구성 메인 클래스.
static	css, js, img등의 정적 resource directory
templates	SpringBoot에서 사용 가능한 여러가지 View Template(Thymeleaf, Velocity, FreeMarker등) 위치
application.properties	application 및 스프링의 설정 등에서 사용할 여러 가지 property를 정의한 file
src/main	jsp등의 리소스 directory

SpringBoot Project 구조 (STS)

- project 생성 구조 및 주요 구성 폴더/파일

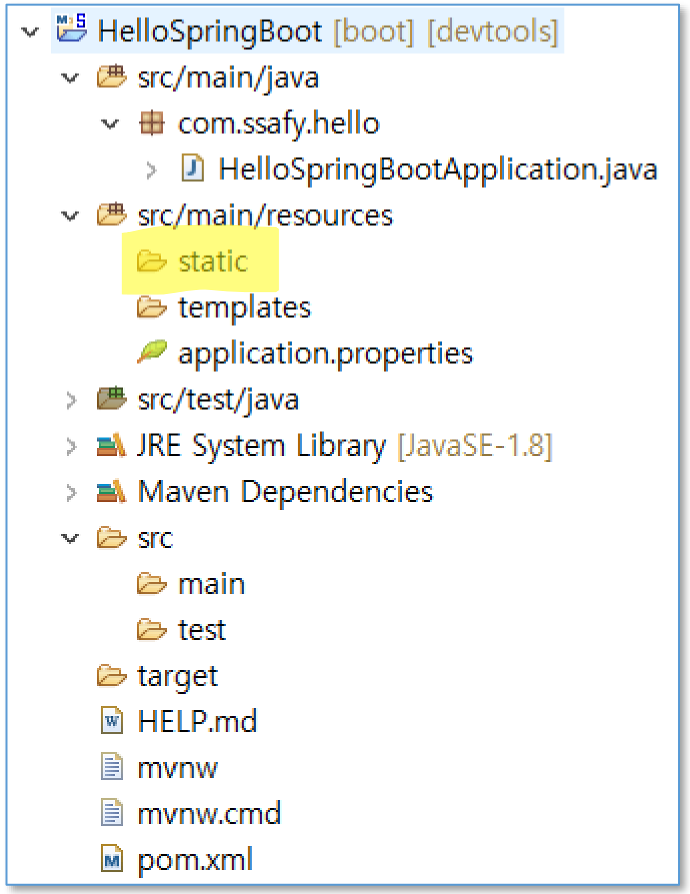


```
HelloSpringBoot [boot] [devtools]
  src/main/java
    com.ssafy.hello
      HelloSpringBootApplication.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

프로젝트의 주요 파일	설명
src/main/java	java source directory
HelloSpringBootApplication.java	application을 시작할 수 있는 main method가 존재하는 스프링 구성 메인 클래스.
static	css, js, img등의 정적 resource directory
templates	SpringBoot에서 사용 가능한 여러가지 View Template(Thymeleaf, Velocity, FreeMarker등) 위치
application.properties	application 및 스프링의 설정 등에서 사용할 여러 가지 property를 정의한 file
src/main	jsp등의 리소스 directory

SpringBoot Project 구조 (STS)

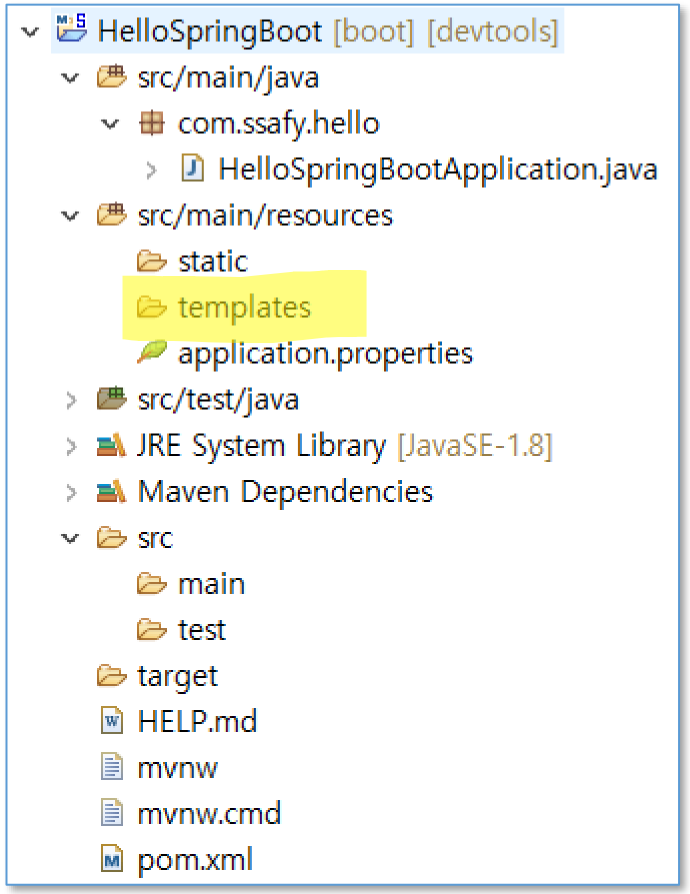
- project 생성 구조 및 주요 구성 폴더/파일


HelloSpringBoot [boot] [devtools]
- src/main/java
  - com.ssafy.hello
    - HelloSpringBootApplication.java
- src/main/resources
  - static
  - templates
  - application.properties
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
  - main
  - test
- target
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml

프로젝트의 주요 파일	설명
src/main/java	java source directory
HelloSpringBootApplication.java	application을 시작할 수 있는 main method가 존재하는 스프링 구성 메인 클래스.
static	css, js, img등의 정적 resource directory
templates	SpringBoot에서 사용 가능한 여러가지 View Template(Thymeleaf, Velocity, FreeMarker등) 위치
application.properties	application 및 스프링의 설정 등에서 사용할 여러 가지 property를 정의한 file
src/main	jsp등의 리소스 directory

SpringBoot Project 구조 (STS)

- project 생성 구조 및 주요 구성 폴더/파일

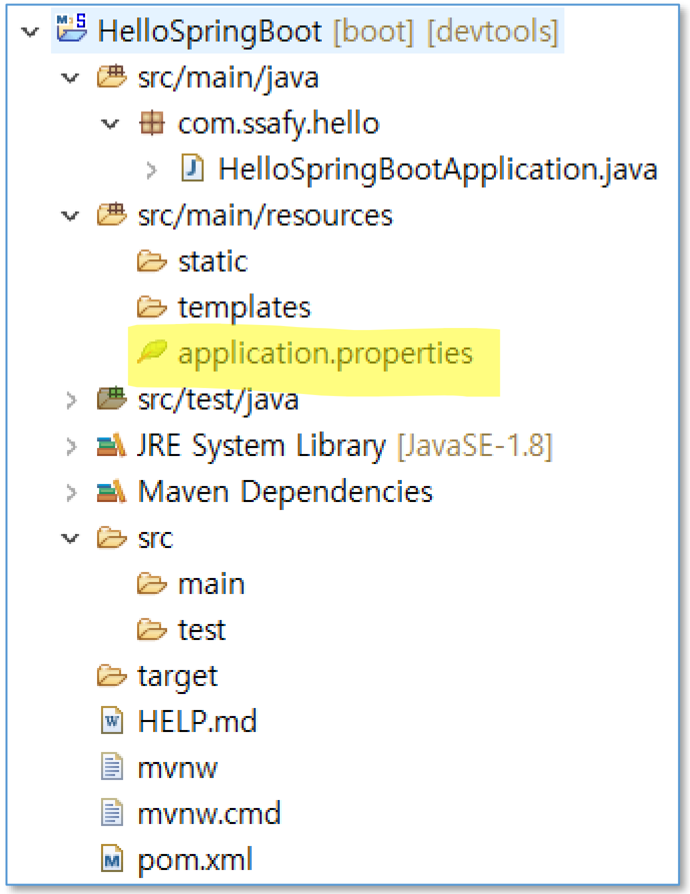


```
HelloSpringBoot [boot] [devtools]
  src/main/java
    com.ssafy.hello
      HelloSpringBootApplication.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

프로젝트의 주요 파일	설명
src/main/java	java source directory
HelloSpringBootApplication.java	application을 시작할 수 있는 main method가 존재하는 스프링 구성 메인 클래스.
static	css, js, img등의 정적 resource directory
templates	SpringBoot에서 사용 가능한 여러가지 View Template(Thymeleaf, Velocity, FreeMarker등) 위치
application.properties	application 및 스프링의 설정 등에서 사용할 여러 가지 property를 정의한 file
src/main	jsp등의 리소스 directory

SpringBoot Project 구조 (STS)

- project 생성 구조 및 주요 구성 폴더/파일

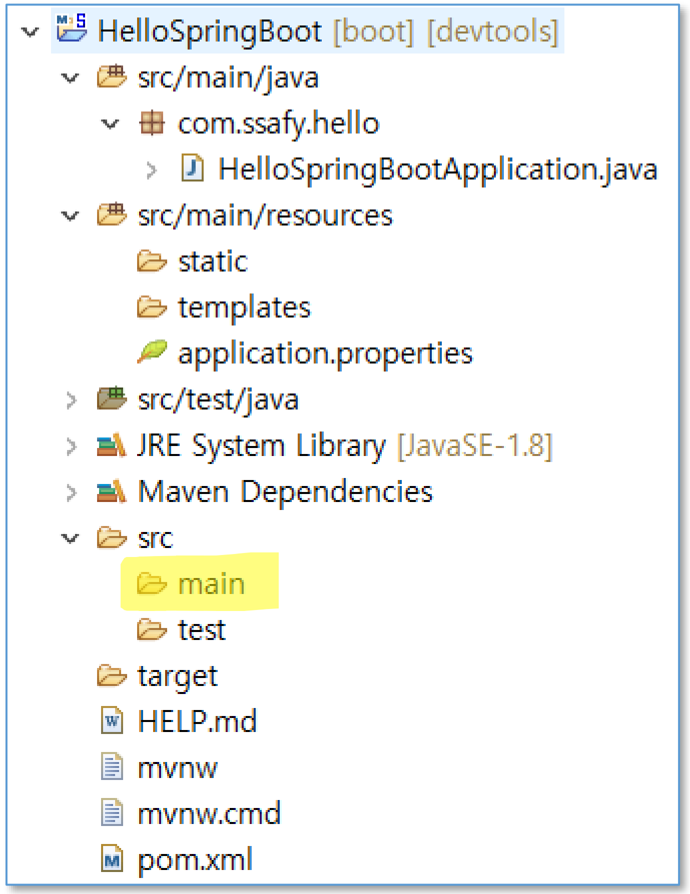


```
HelloSpringBoot [boot] [devtools]
  src/main/java
    com.ssafy.hello
      HelloSpringBootApplication.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

프로젝트의 주요 파일	설명
src/main/java	java source directory
HelloSpringBootApplication.java	application을 시작할 수 있는 main method가 존재하는 스프링 구성 메인 클래스.
static	css, js, img등의 정적 resource directory
templates	SpringBoot에서 사용 가능한 여러가지 View Template(Thymeleaf, Velocity, FreeMarker등) 위치
application.properties	application 및 스프링의 설정 등에서 사용할 여러 가지 property를 정의한 file
src/main	jsp등의 리소스 directory

SpringBoot Project 구조 (STS)

- project 생성 구조 및 주요 구성 폴더/파일

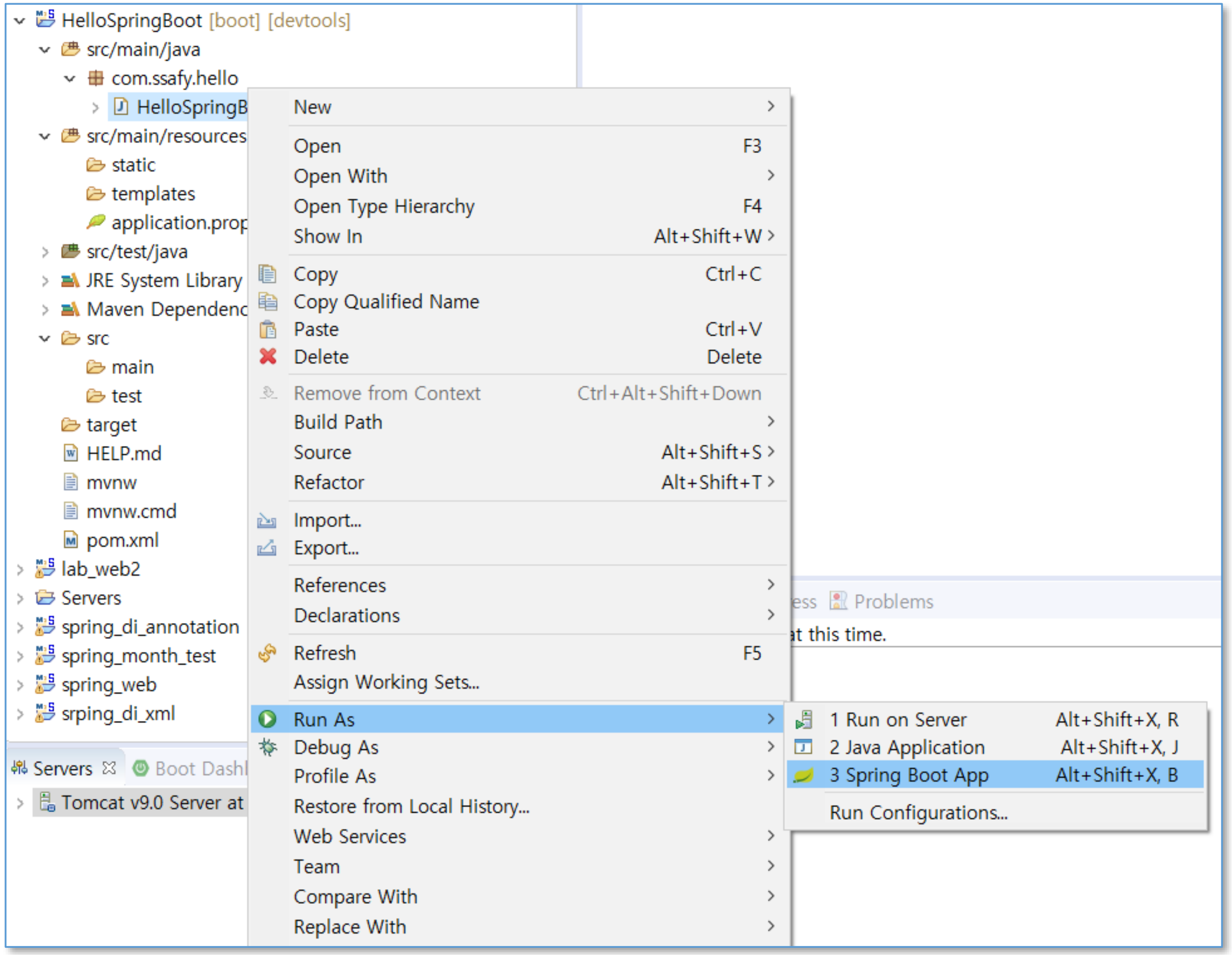


```
HelloSpringBoot [boot] [devtools]
  src/main/java
    com.ssafy.hello
      HelloSpringBootApplication.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

프로젝트의 주요 파일	설명
src/main/java	java source directory
HelloSpringBootApplication.java	application을 시작할 수 있는 main method가 존재하는 스프링 구성 메인 클래스.
static	css, js, img등의 정적 resource directory
templates	SpringBoot에서 사용 가능한 여러가지 View Template(Thymeleaf, Velocity, FreeMarker등) 위치
application.properties	application 및 스프링의 설정 등에서 사용할 여러 가지 property를 정의한 file
src/main	jsp등의 리소스 directory

SpringBoot Project 실행 (STS)

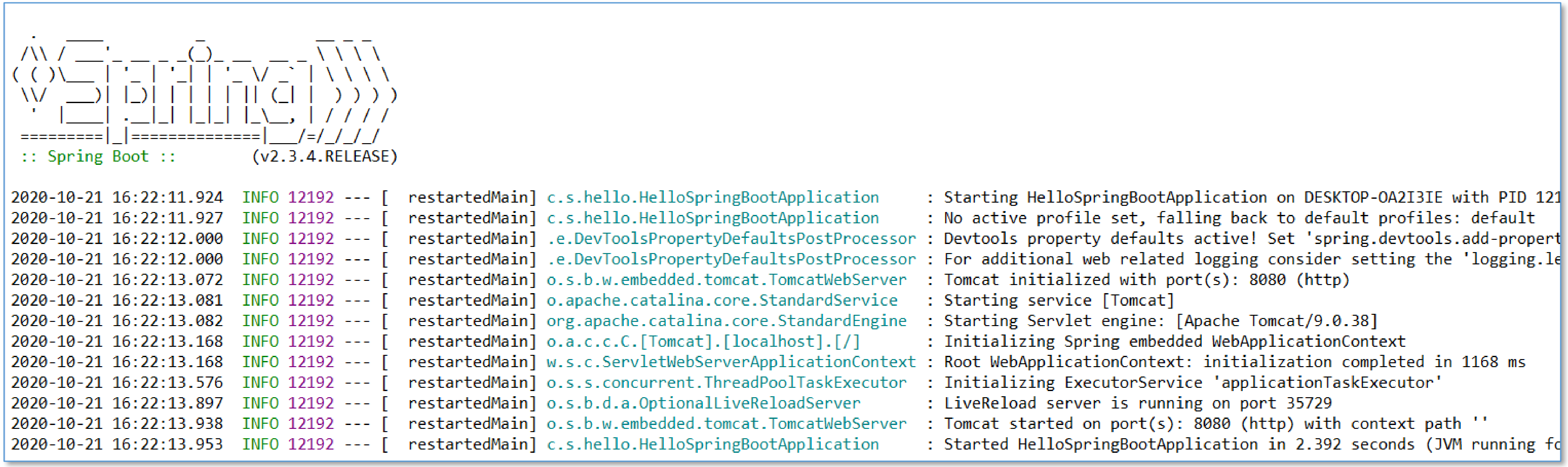
- 실행



SpringBoot

SpringBoot Project 실행 (STS)

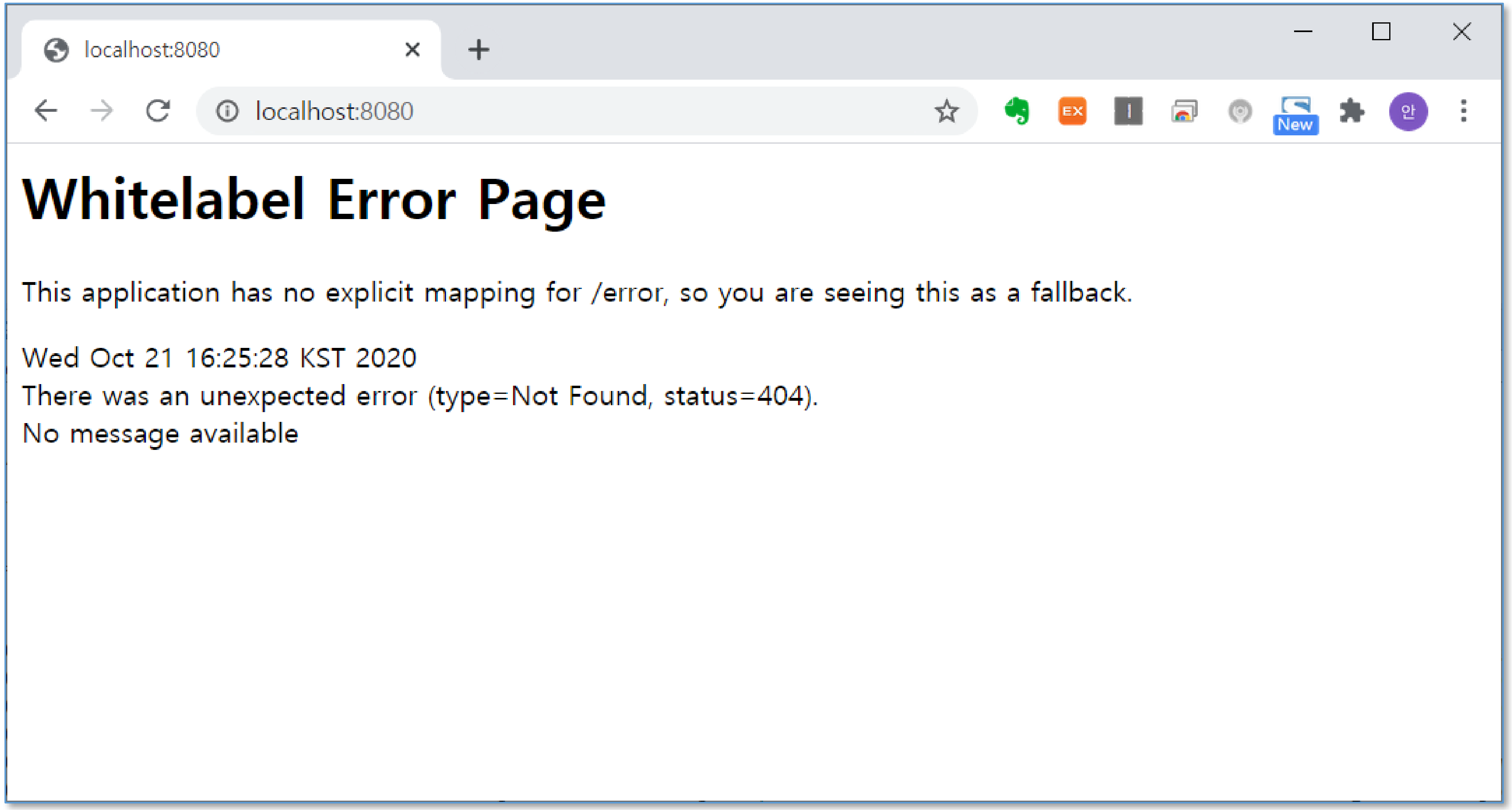
- 실행 콘솔 창



```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.3.4.RELEASE)

2020-10-21 16:22:11.924  INFO 12192 --- [  restartedMain] c.s.hello.HelloSpringBootApplication     : Starting HelloSpringBootApplication on DESKTOP-OA2I3IE with PID 121
2020-10-21 16:22:11.927  INFO 12192 --- [  restartedMain] c.s.hello.HelloSpringBootApplication     : No active profile set, falling back to default profiles: default
2020-10-21 16:22:12.000  INFO 12192 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-propert
2020-10-21 16:22:12.000  INFO 12192 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.le
2020-10-21 16:22:13.072  INFO 12192 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2020-10-21 16:22:13.081  INFO 12192 --- [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2020-10-21 16:22:13.082  INFO 12192 --- [  restartedMain] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.38]
2020-10-21 16:22:13.168  INFO 12192 --- [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2020-10-21 16:22:13.168  INFO 12192 --- [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1168 ms
2020-10-21 16:22:13.576  INFO 12192 --- [  restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2020-10-21 16:22:13.897  INFO 12192 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2020-10-21 16:22:13.938  INFO 12192 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2020-10-21 16:22:13.953  INFO 12192 --- [  restartedMain] c.s.hello.HelloSpringBootApplication     : Started HelloSpringBootApplication in 2.392 seconds (JVM running fo
```

SpringBoot Project 실행 (STS)

- 브라우저를 열고 http://localhost:8080 실행



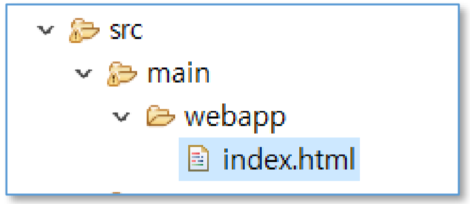
Whitelabel Error Page

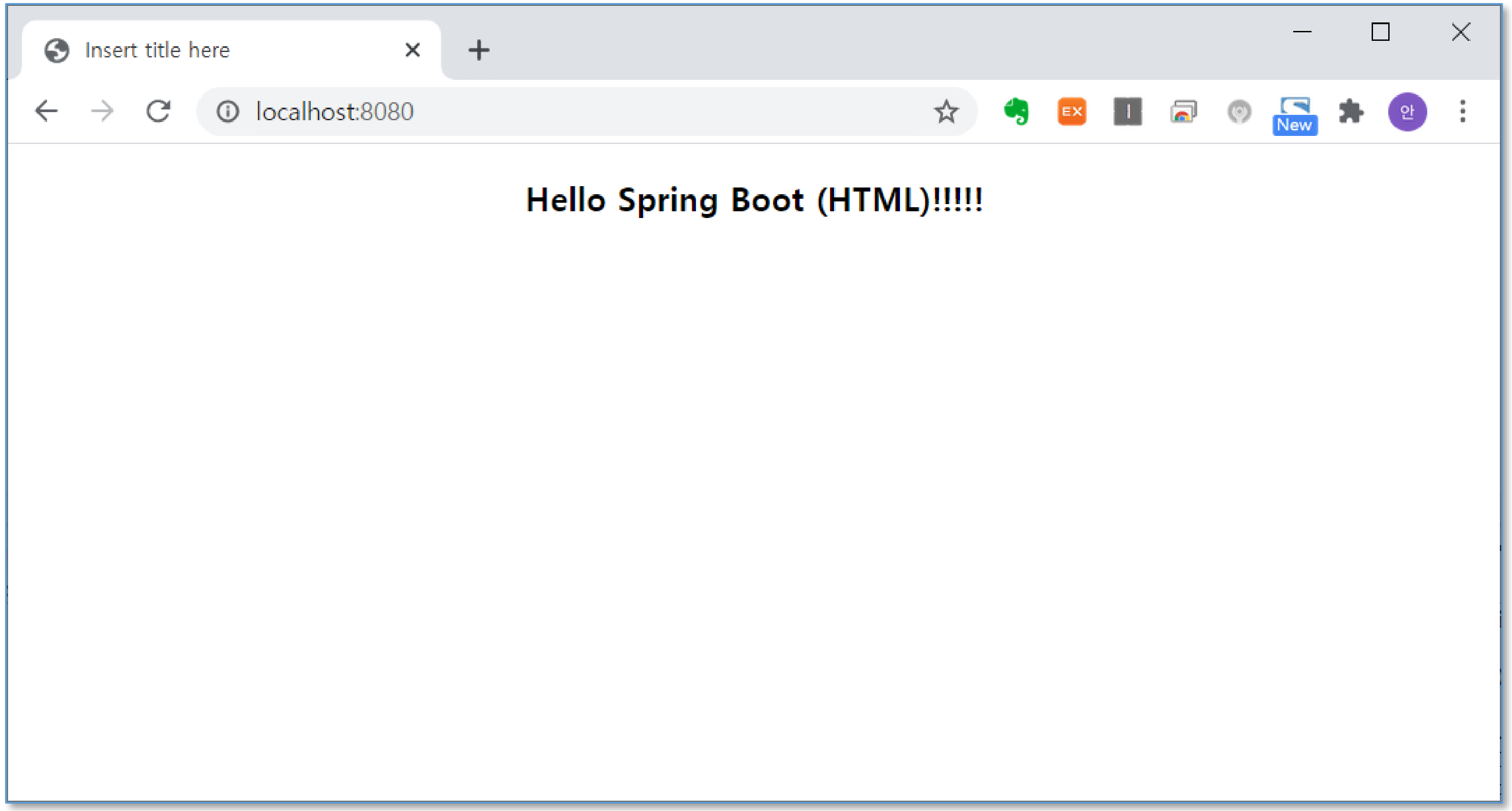
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Oct 21 16:25:28 KST 2020
There was an unexpected error (type=Not Found, status=404).
No message available

SpringBoot Project 실행 (STS)

- src/main에 webapp 폴더 생성 후 index.html 작성.
- 브라우저 : http://localhost:8080 실행.

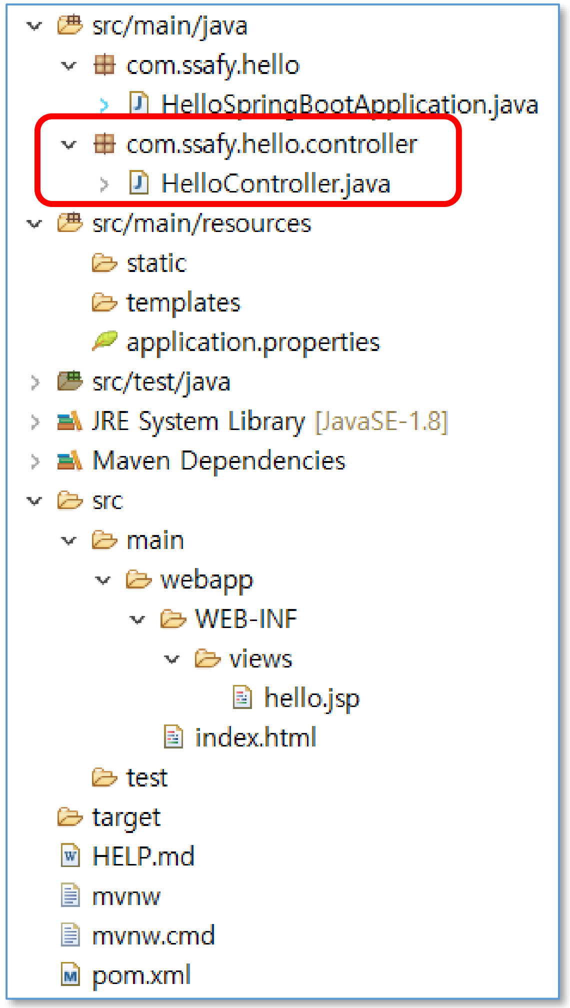




Hello Spring Boot (HTML)!!!!!

SpringBoot Project (STS)

- com/ssafy/hello/controller/HelloController 생성

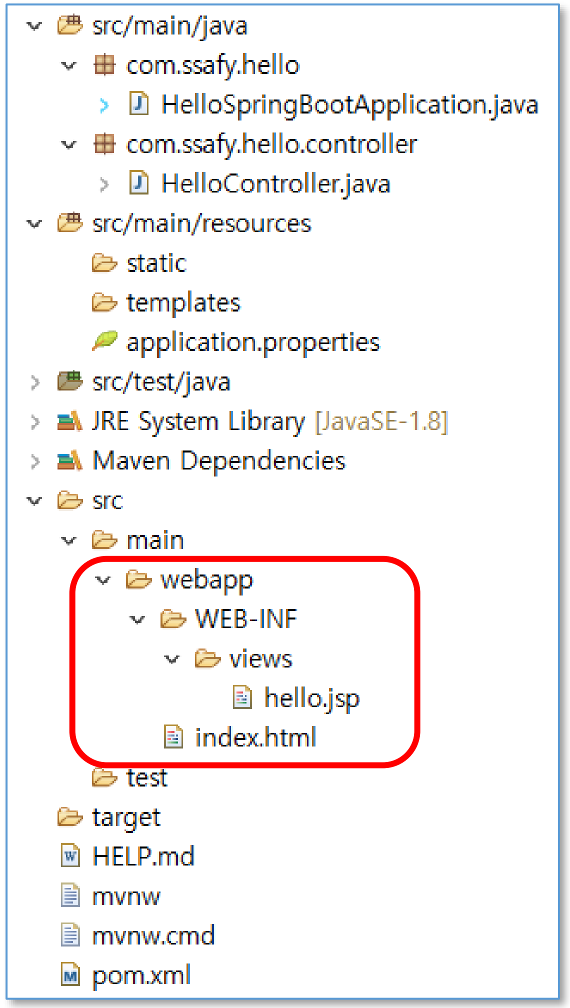


```
src/main/java
  com.ssafy.hello
    HelloSpringBootApplication.java
  com.ssafy.hello.controller
    HelloController.java
src/main/resources
  static
  templates
  application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  main
    webapp
      WEB-INF
        views
          hello.jsp
      index.html
  test
target
HELP.md
mvnw
mvnw.cmd
pom.xml
```

```java
package com.ssafy.hello.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HelloController {

    @RequestMapping("/")
    public String hello(Model model) {
        model.addAttribute("msg", "안녕 하세요 스프링 부트입니다.~~~");
        return "hello";
    }

}
```

SpringBoot Project (STS)

- webapp/WEB-INF/views/hello.jsp 생성

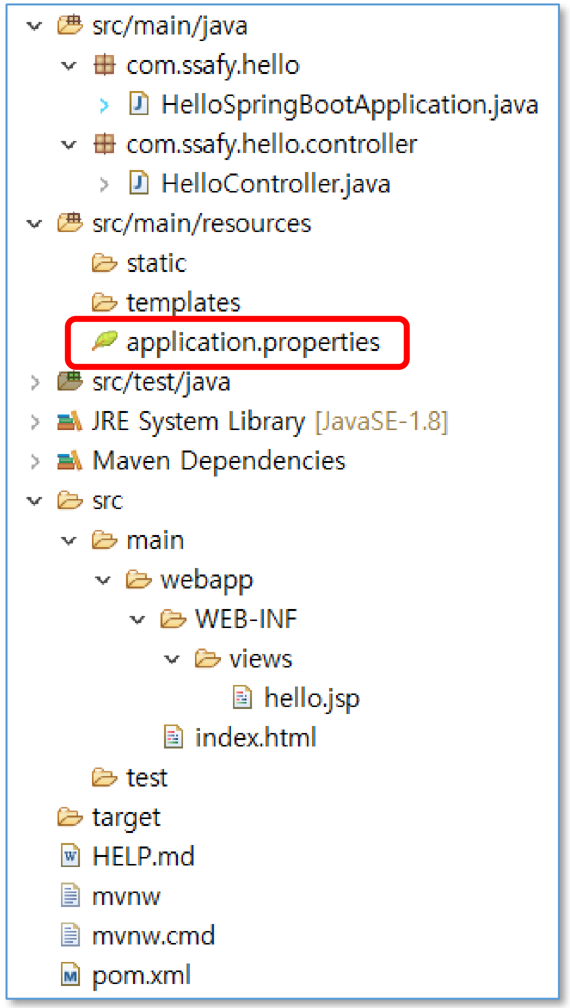


```
src/main/java
  com.ssafy.hello
    HelloSpringBootApplication.java
  com.ssafy.hello.controller
    HelloController.java
src/main/resources
  static
  templates
  application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  main
    webapp
      WEB-INF
        views
          hello.jsp
      index.html
  test
target
HELP.md
mvnw
mvnw.cmd
pom.xml
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>SpringBoot</title>
</head>
<body>
<body>
    <div align="center">
        <h3>${msg}</h3>
    </div>
</body>
</html>
```

SpringBoot Project (STS)

- application.properties → 파일에 추가

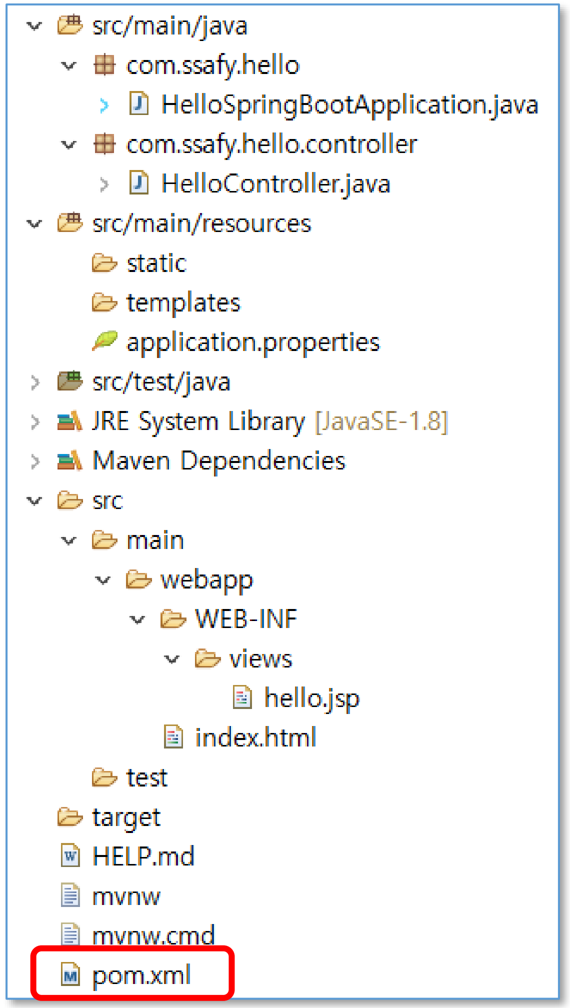


```
src/main/java
  com.ssafy.hello
    HelloSpringBootApplication.java
  com.ssafy.hello.controller
    HelloController.java
src/main/resources
  static
  templates
  application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  main
    webapp
      WEB-INF
        views
          hello.jsp
      index.html
  test
target
HELP.md
mvnw
mvnw.cmd
pom.xml
```

```
# Web ContextRootPath and PortNumber Settings
server.servlet.context-path=/ssafy
server.port=80

# JSP Path (ViewResolver)
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

SpringBoot Project (STS)

- pom.xml 에 dependency 추가

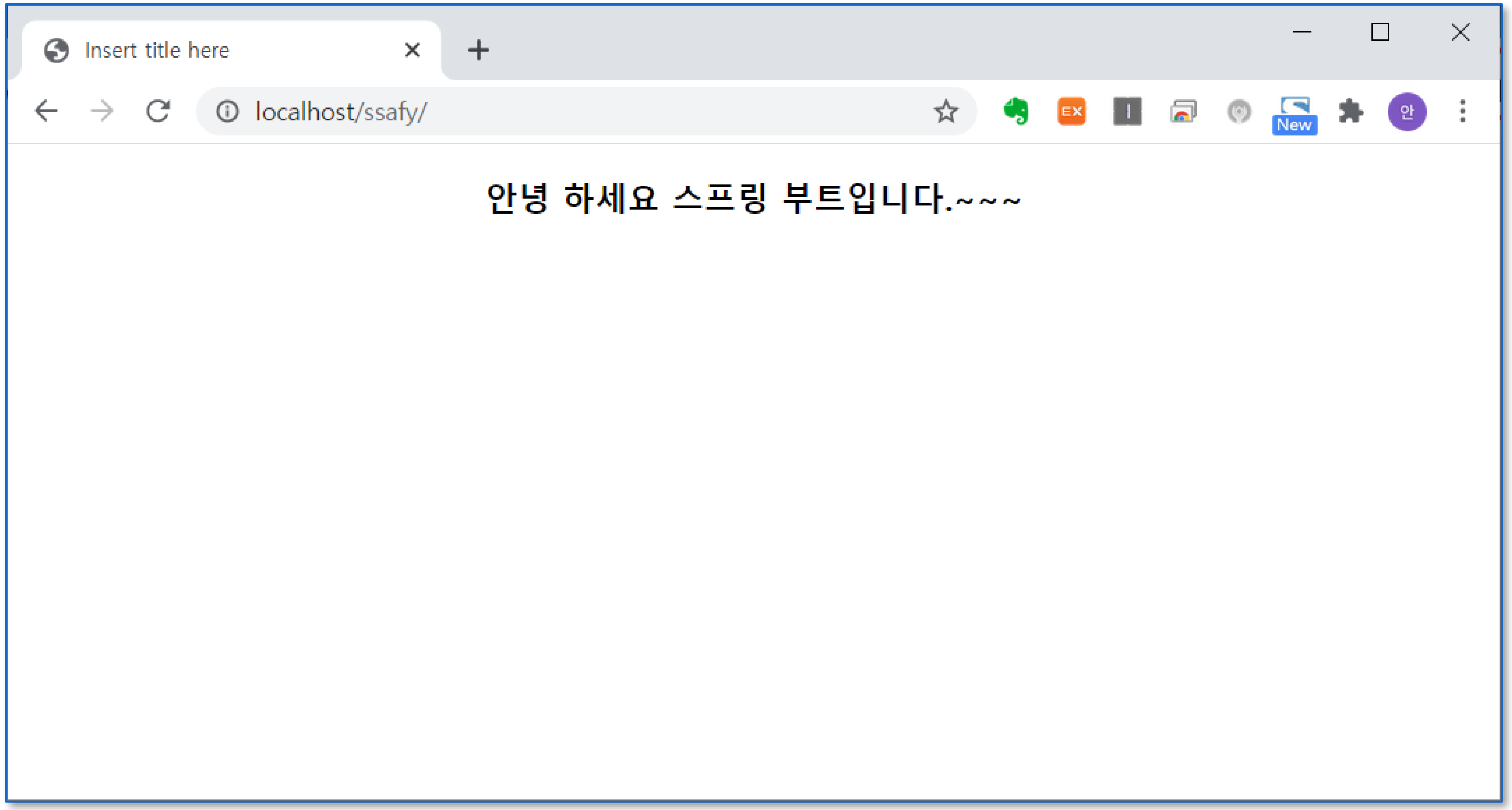


```
src/main/java
  com.ssafy.hello
    HelloSpringBootApplication.java
  com.ssafy.hello.controller
    HelloController.java
src/main/resources
  static
  templates
  application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  main
    webapp
      WEB-INF
        views
          hello.jsp
      index.html
  test
target
HELP.md
mvnw
mvnw.cmd
pom.xml
```

```xml
<!-- jsp 설정 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
```

SpringBoot Project (STS)

- HelloSpringBootApplication 실행
- 브라우저 : http://localhost/ssafy



```
안녕 하세요 스프링 부트입니다.~~~
```

SpringBoot Project 특징 (STS)

- Spring 사용시 개발자가 직접 해야만 했던 복잡한 설정을 해결한다.
- SpringBoot의 장점
  - 간편하고 자동화된 빌드 및 설정을 제공한다.
  - project에 따라 자주 사용되는 library들이 미리 조합되어 있다.(Best Practice)
  - 복잡한 설정(XML)을 하지 않아도 된다.(자동)
  - 내장 서버를 제공해서 WAS를 추가로 설치하지 않아도 개발과 손쉬운 배포가 가능하다.(독립실행)
  - WAS에 배포하지 않고도 실행할 수 있는 JAR파일로 Web Application을 개발 할 수 있다.

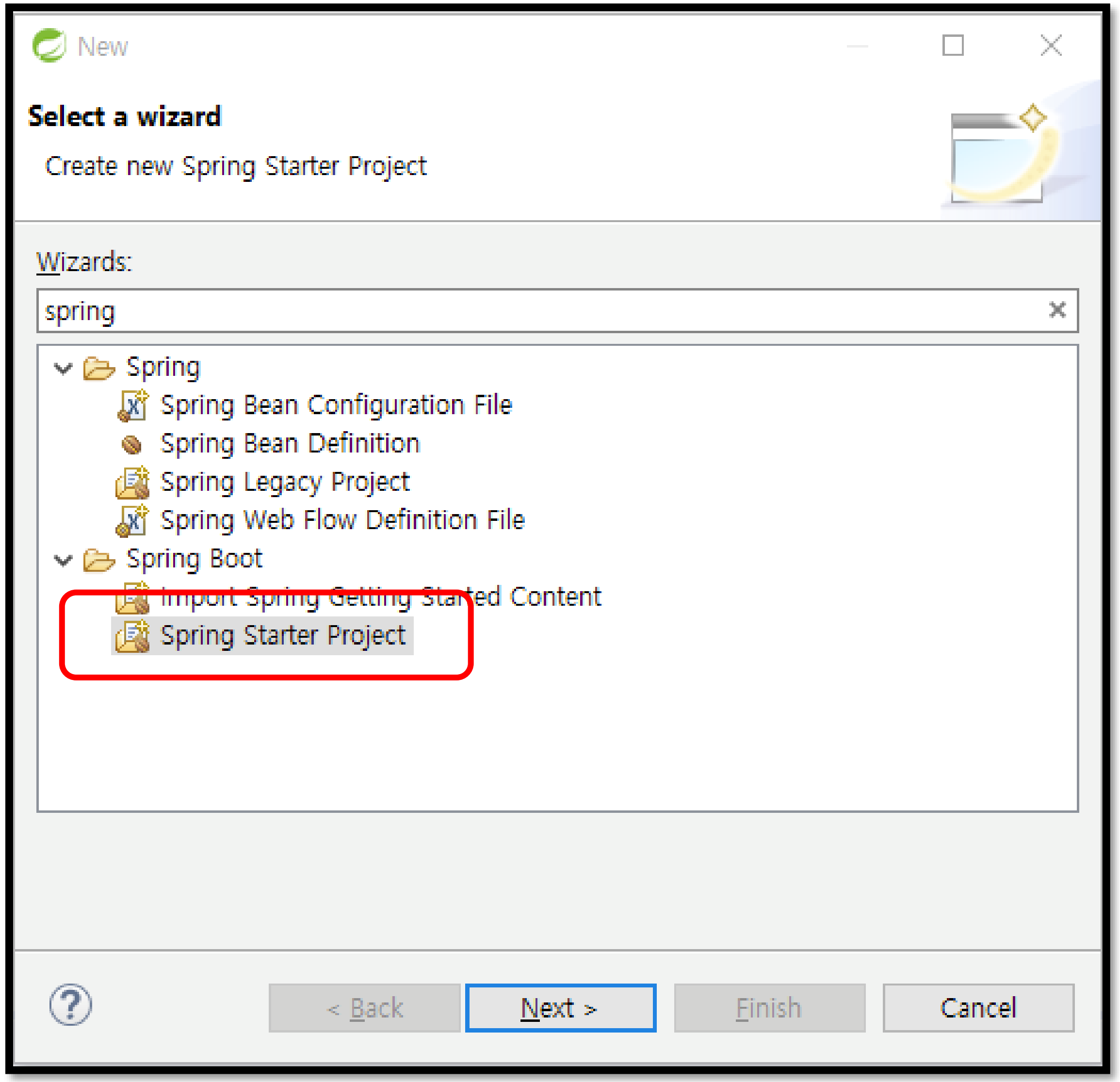
SAMSUNG
SW
ACADEMY
FOR
YOUTH

함께가요 미래로!
Enabling People

SpringBoot 게시판

SpringBoot 게시판

Spring Starter Project 선택


New

Select a wizard

Create new Spring Starter Project

Wizards:

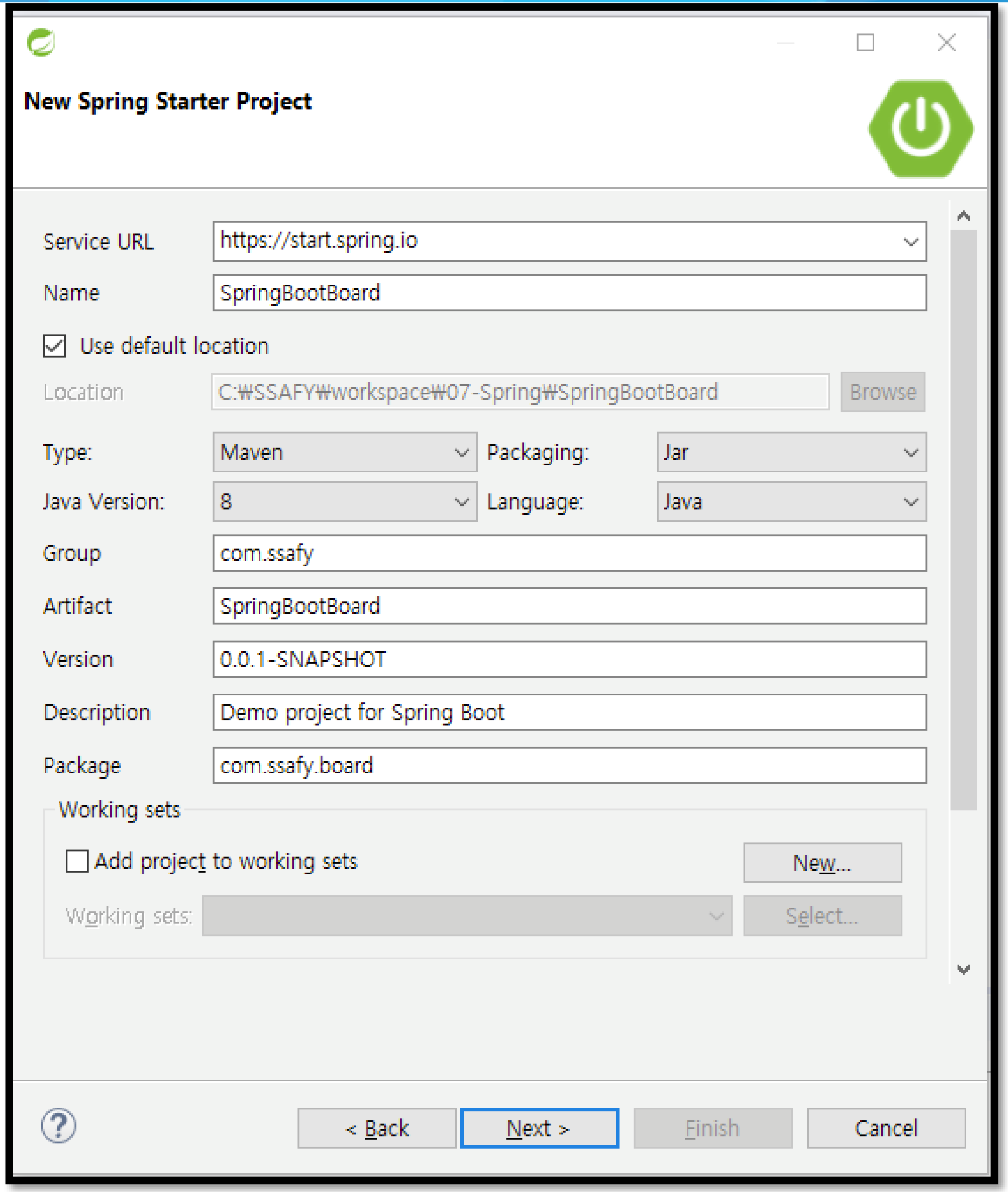
spring

- Spring
  - Spring Bean Configuration File
  - Spring Bean Definition
  - Spring Legacy Project
  - Spring Web Flow Definition File
- Spring Boot
  - Import Spring Getting Started Content
  - Spring Starter Project

SpringBoot 게시판

SpringBootBoard 프로젝트 생성

- Name : SpringBootBoard
- Group : com.ssafy
- Artifact : SpringBootBoard
- Package : com.ssafy.board


New Spring Starter Project

Service URL	https://start.spring.io
Name	SpringBootBoard
☑ Use default location	
Location	C:₩SSAFY₩workspace₩07-Spring₩SpringBootBoard
Type:	Maven
Packaging:	Jar
Java Version:	8
Language:	Java
Group	com.ssafy
Artifact	SpringBootBoard
Version	0.0.1-SNAPSHOT
Description	Demo project for Spring Boot
Package	com.ssafy.board

Working sets

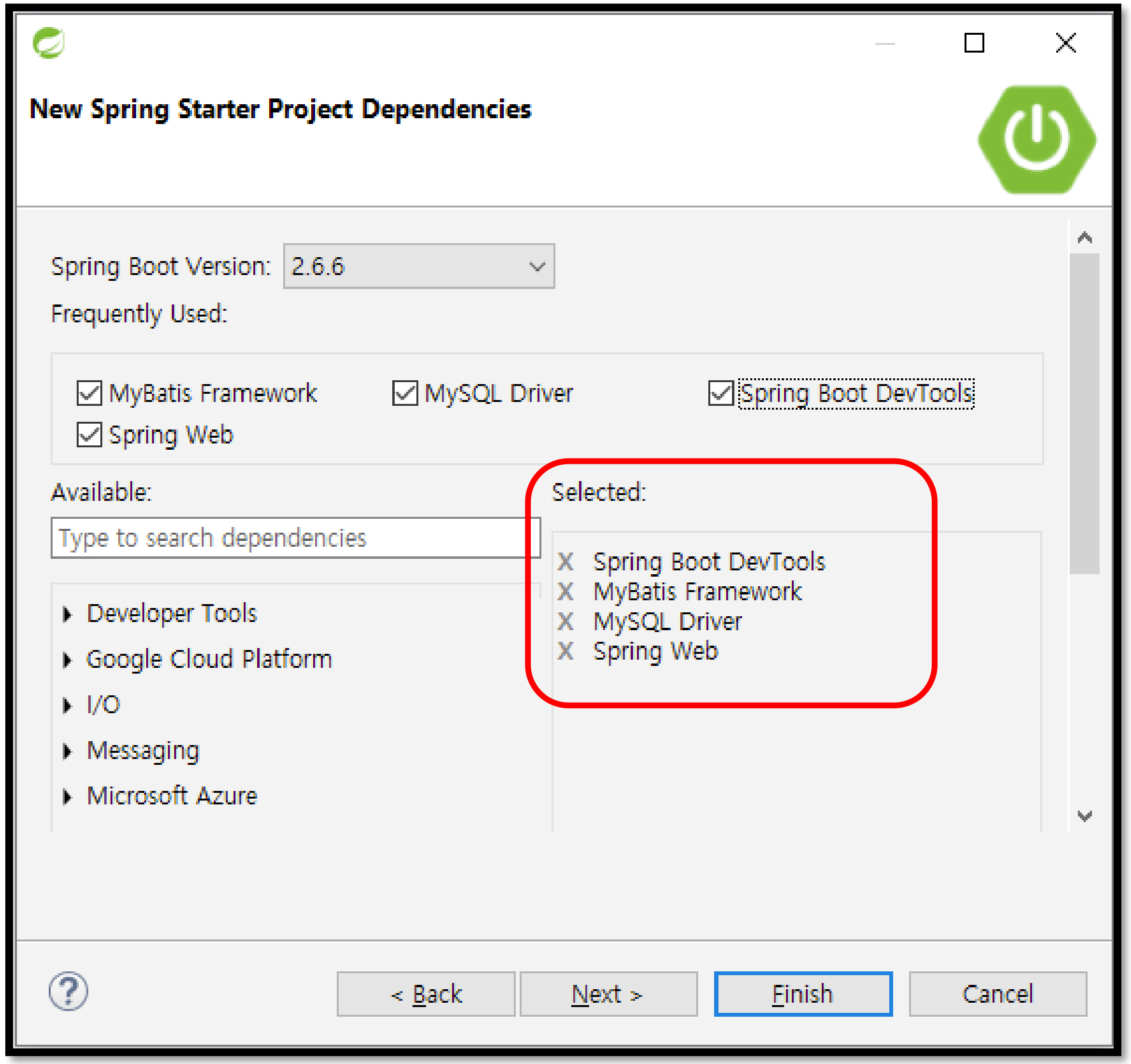
☐ Add project to working sets

Working sets:

SpringBoot 게시판

SpringBootBoard 프로젝트 생성

- DevTools
- MyBatis Framework
- MySQL Driver
- Spring Web



New Spring Starter Project Dependencies

Spring Boot Version: 2.6.6

Frequently Used:

☑ MyBatis Framework ☑ MySQL Driver ☑ Spring Boot DevTools
☑ Spring Web

Available:

Type to search dependencies

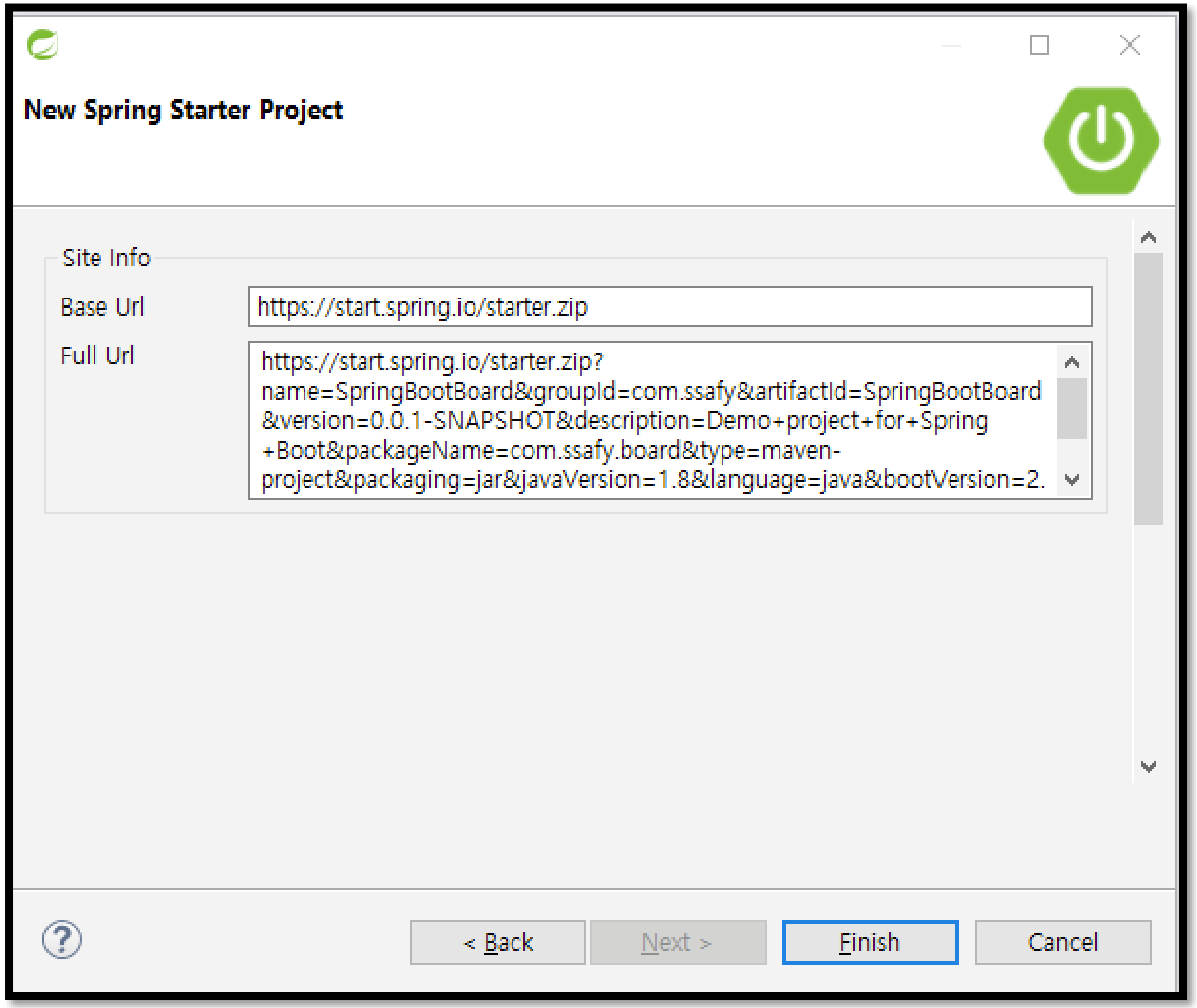
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure

Selected:

X Spring Boot DevTools
X MyBatis Framework
X MySQL Driver
X Spring Web

SpringBoot 게시판

SpringBootBoard 생성 완료



Site Info

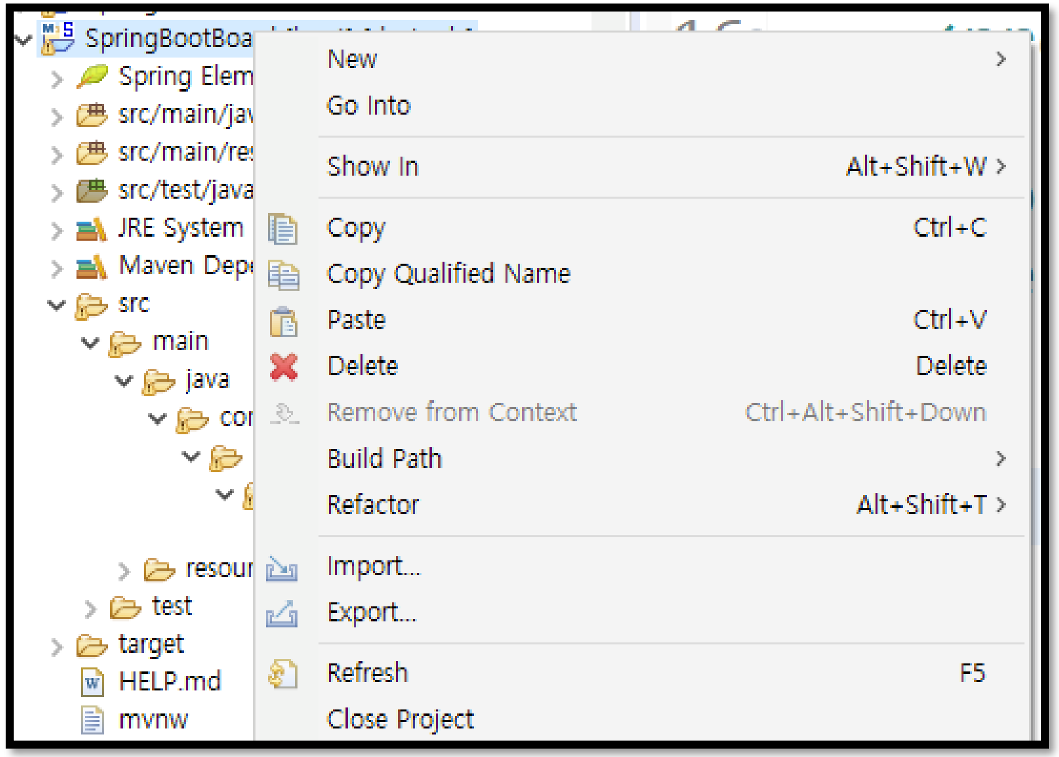
Base Url https://start.spring.io/starter.zip

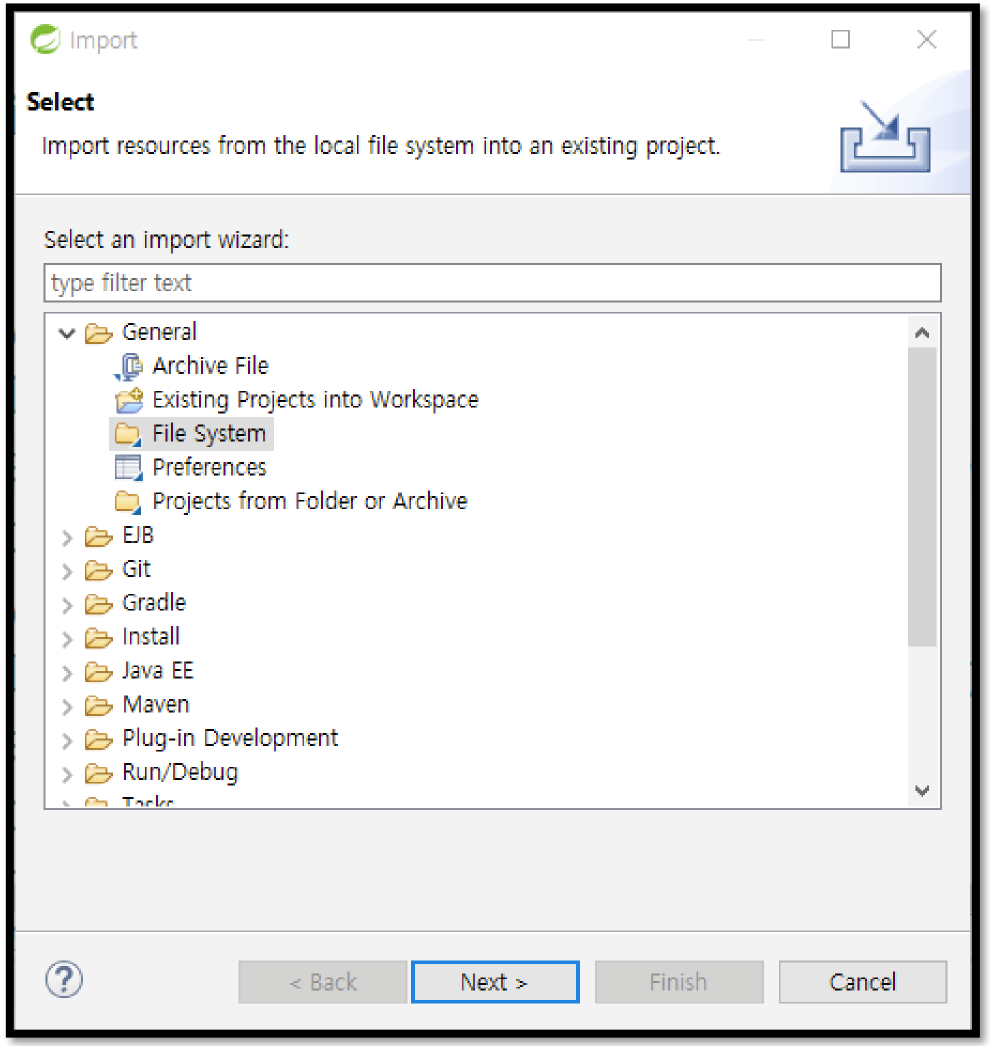
Full Url https://start.spring.io/starter.zip?name=SpringBootBoard&groupId=com.ssafy&artifactId=SpringBootBoard&version=0.0.1-SNAPSHOT&description=Demo+project+for+Spring+Boot&packageName=com.ssafy.board&type=maven-project&packaging=jar&javaVersion=1.8&language=java&bootVersion=2.

SpringBoot 게시판

기존 게시판 코드 가져오기

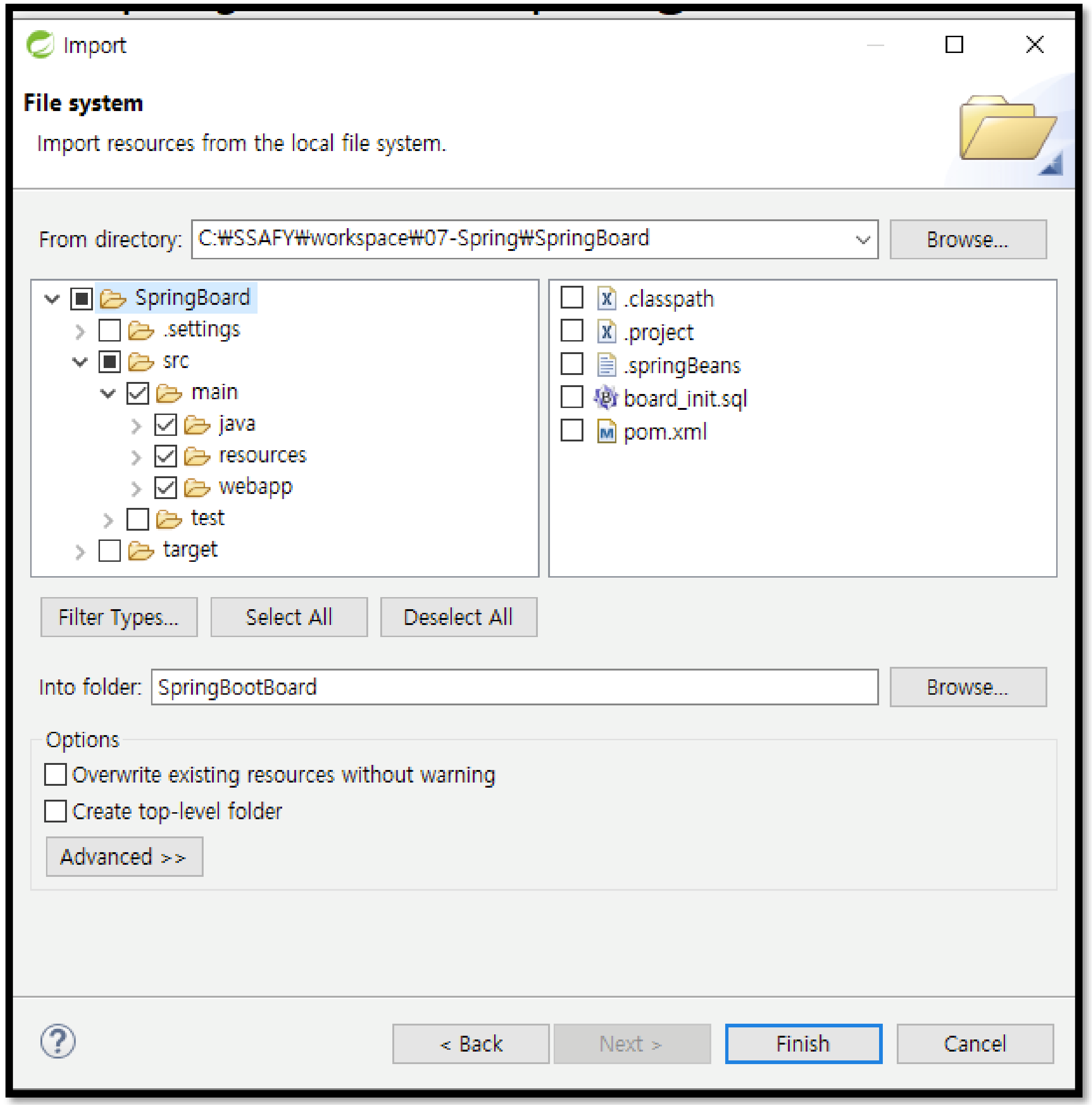
- 프로젝트 선택 후 import
- General → File System





SpringBoot 게시판

- 기존 프로젝트 src/main 선택


Import

File system

Import resources from the local file system.

From directory: C:₩SSAFY₩workspace₩07-Spring₩SpringBoard

```
■ SpringBoard
  ☐ .settings
  ■ src
    ☑ main
      ☑ java
      ☑ resources
      ☑ webapp
    ☐ test
  ☐ target
```

```
☐ .classpath
☐ .project
☐ .springBeans
☐ board_init.sql
☐ pom.xml
```

Into folder: SpringBootBoard

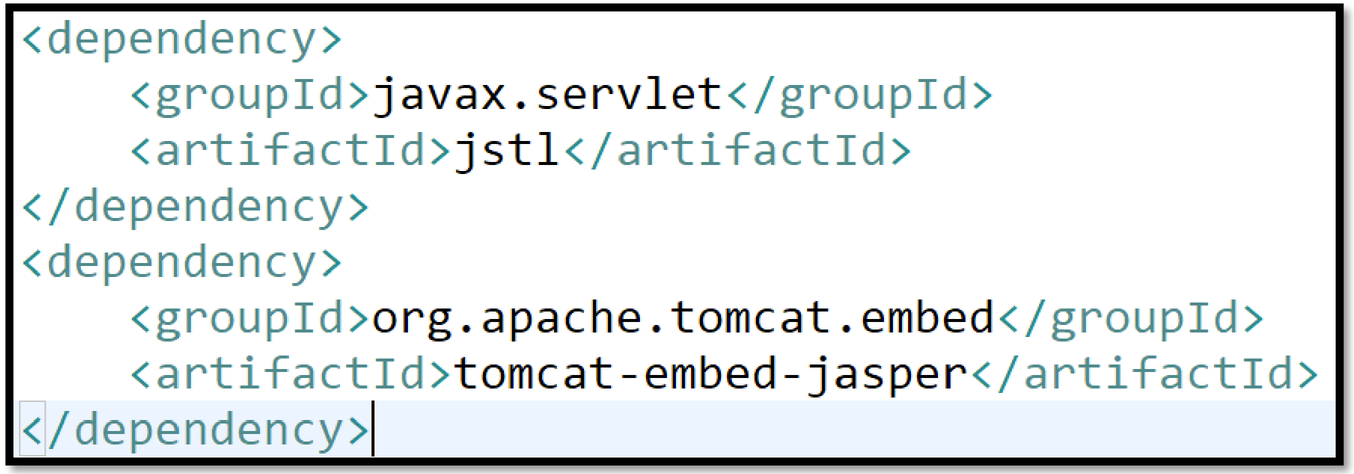
Options

☐ Overwrite existing resources without warning

☐ Create top-level folder

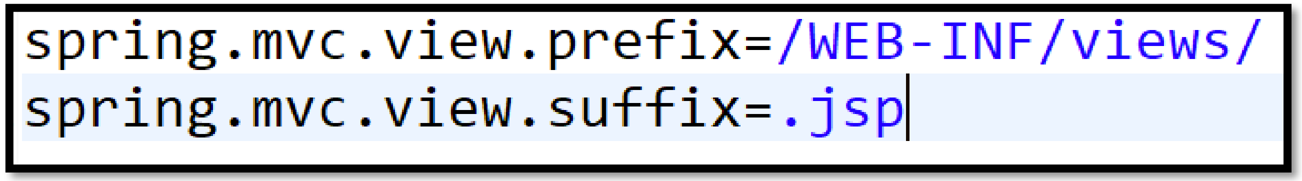
JSP 부분 설정

- pom.xml



```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
```

- application.properties

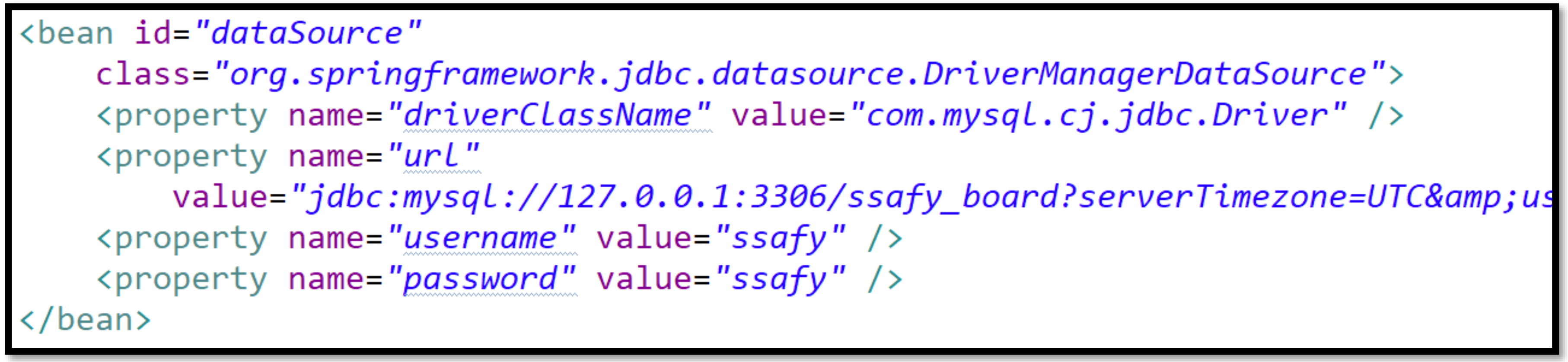


```
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

SpringBoot 게시판

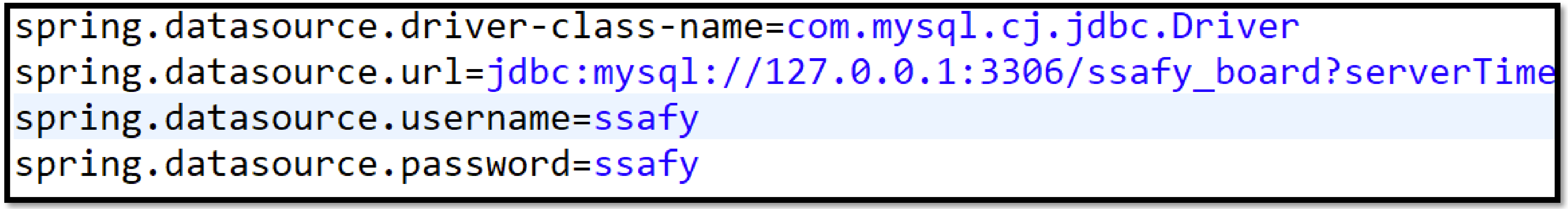
DB 설정 : MySQL

- root-context.xml



```
<bean id="dataSource"
    class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />
    <property name="url"
        value="jdbc:mysql://127.0.0.1:3306/ssafy_board?serverTimezone=UTC&amp;us
    <property name="username" value="ssafy" />
    <property name="password" value="ssafy" />
</bean>
```

- application.properties



```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/ssafy_board?serverTime
spring.datasource.username=ssafy
spring.datasource.password=ssafy
```

SpringBoot 게시판

DB 설정 : MyBatis

- root-context.xml

```xml
<bean class="org.mybatis.spring.SqlSessionFactoryBean" id="sqlSessionFactory">
    <property name="dataSource" ref="dataSource"></property>
    <property name="mapperLocations" value="classpath:mapper/*.xml"></property>
    <property name="typeAliasesPackage" value="com.ssafy.board.model.dto"></property>
</bean>
```

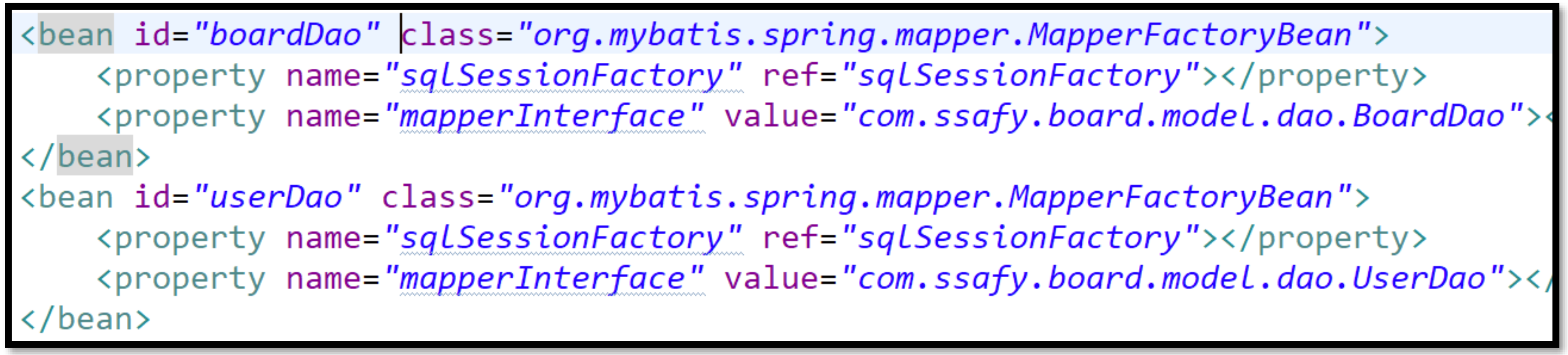
- application.properties

```
mybatis.mapper-locations=classpath:mapper/*.xml
mybatis.type-aliases-package=com.ssafy.board.model.dto
```

SpringBoot 게시판

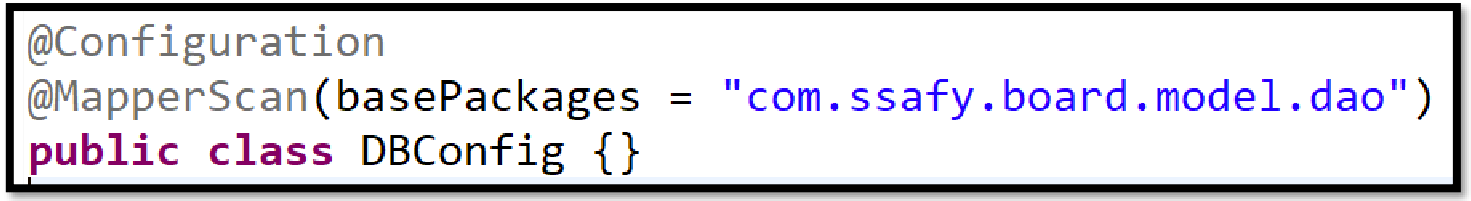
DB 설정 : MyBatis

- root-context.xml



```xml
<bean id="boardDao" class="org.mybatis.spring.mapper.MapperFactoryBean">
    <property name="sqlSessionFactory" ref="sqlSessionFactory"></property>
    <property name="mapperInterface" value="com.ssafy.board.model.dao.BoardDao"><
</bean>
<bean id="userDao" class="org.mybatis.spring.mapper.MapperFactoryBean">
    <property name="sqlSessionFactory" ref="sqlSessionFactory"></property>
    <property name="mapperInterface" value="com.ssafy.board.model.dao.UserDao"></
</bean>
```

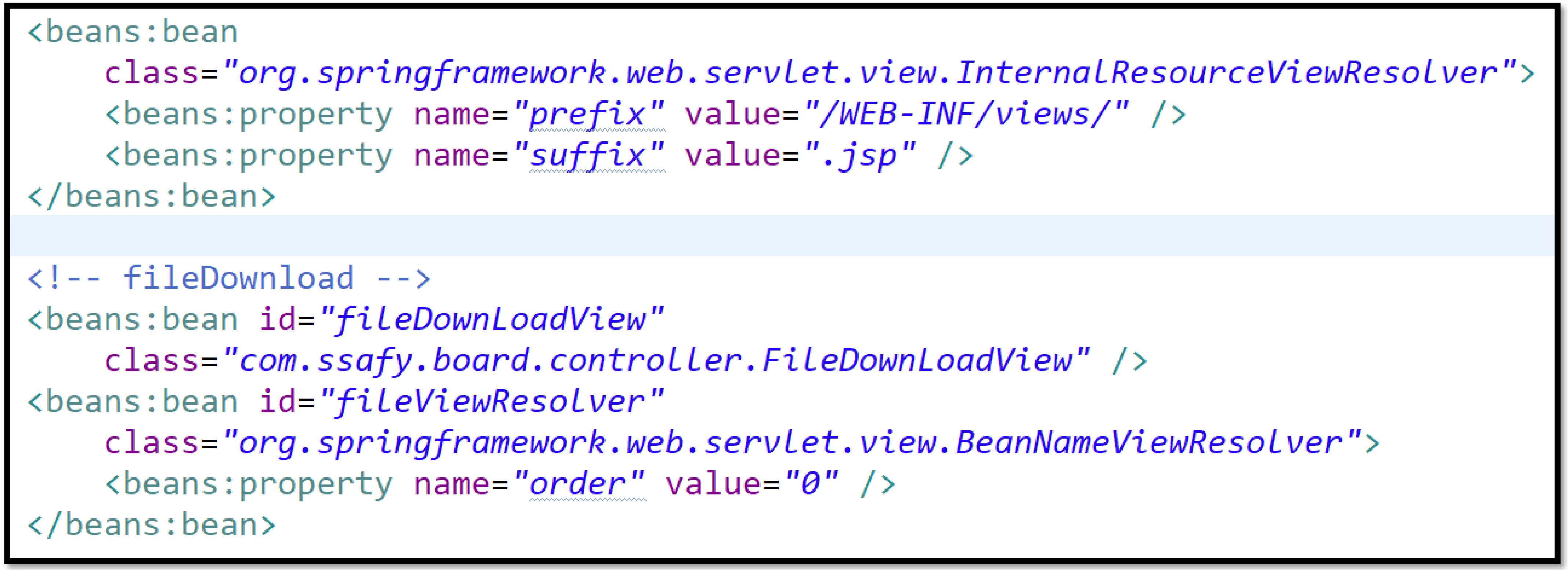
- 자바 설정 파일



```java
@Configuration
@MapperScan(basePackages = "com.ssafy.board.model.dao")
public class DBConfig {}
```

SpringBoot 게시판

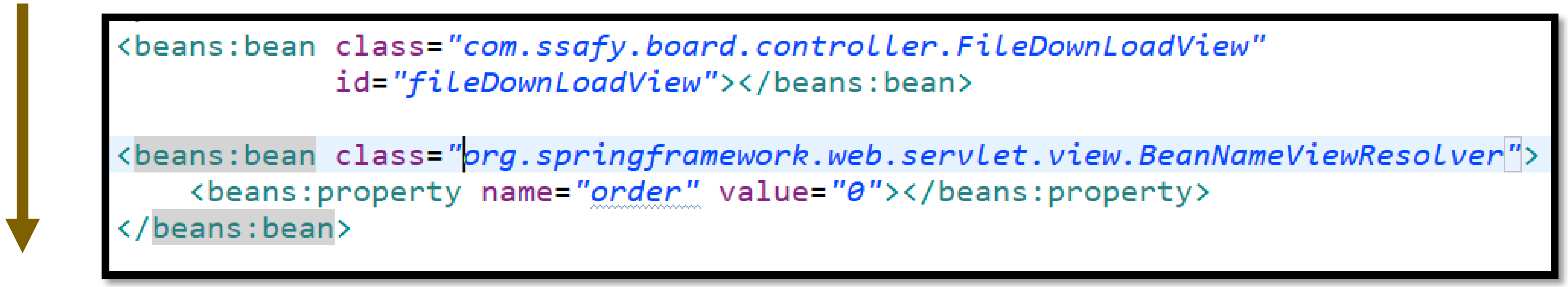
View Resolver 설정



```
<beans:bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>

<!-- fileDownload -->
<beans:bean id="fileDownLoadView"
    class="com.ssafy.board.controller.FileDownLoadView" />
<beans:bean id="fileViewResolver"
    class="org.springframework.web.servlet.view.BeanNameViewResolver">
    <beans:property name="order" value="0" />
</beans:bean>
```

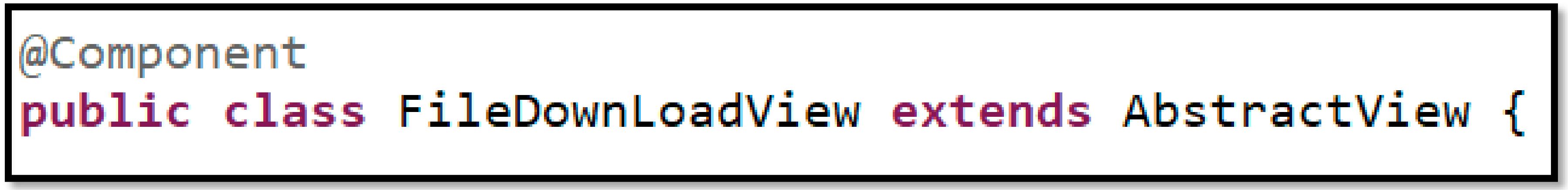
View Resolver

- servlet-context.xml



```
<beans:bean class="com.ssafy.board.controller.FileDownLoadView"
            id="fileDownLoadView"></beans:bean>

<beans:bean class="org.springframework.web.servlet.view.BeanNameViewResolver">
    <beans:property name="order" value="0"></beans:property>
</beans:bean>
```

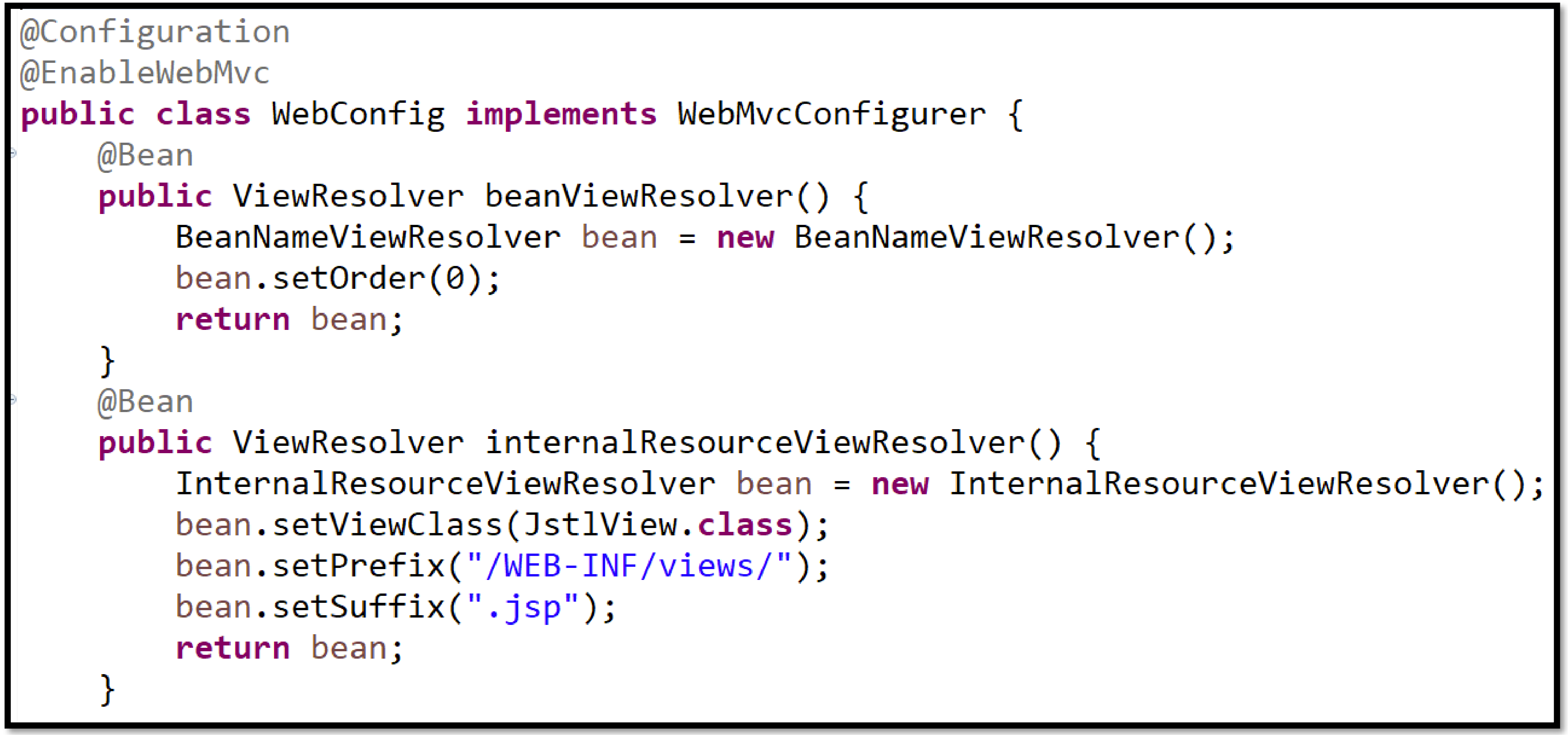
- 빈 등록



```
@Component
public class FileDownLoadView extends AbstractView {
```

View Resolver

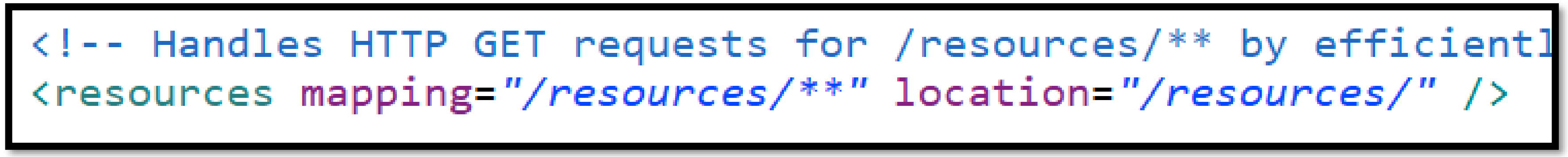
- 자바 설정 파일



```java
@Configuration
@EnableWebMvc
public class WebConfig implements WebMvcConfigurer {
    @Bean
    public ViewResolver beanViewResolver() {
        BeanNameViewResolver bean = new BeanNameViewResolver();
        bean.setOrder(0);
        return bean;
    }
    @Bean
    public ViewResolver internalResourceViewResolver() {
        InternalResourceViewResolver bean = new InternalResourceViewResolver();
        bean.setViewClass(JstlView.class);
        bean.setPrefix("/WEB-INF/views/");
        bean.setSuffix(".jsp");
        return bean;
    }
```

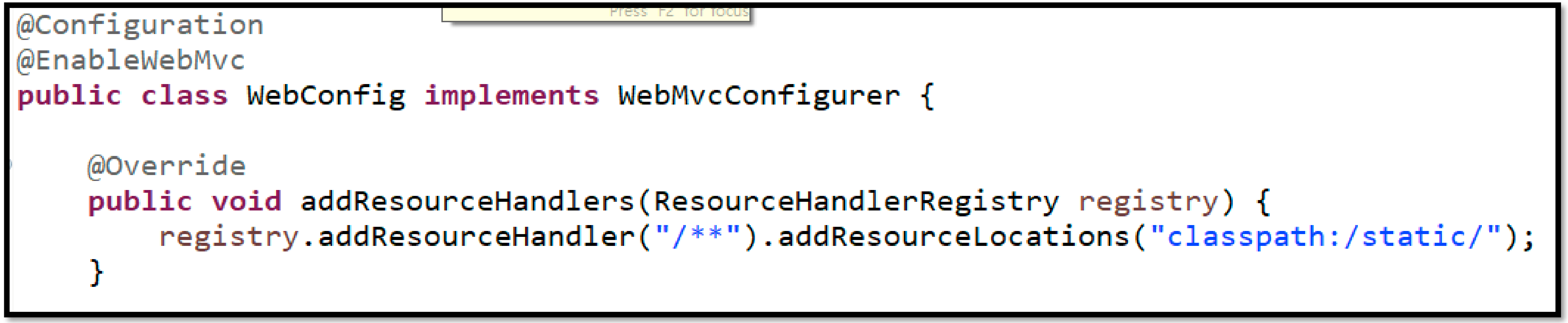
View Resolver

- servlet-context.xml



```
<!-- Handles HTTP GET requests for /resources/** by efficientl
<resources mapping="/resources/**" location="/resources/" />
```

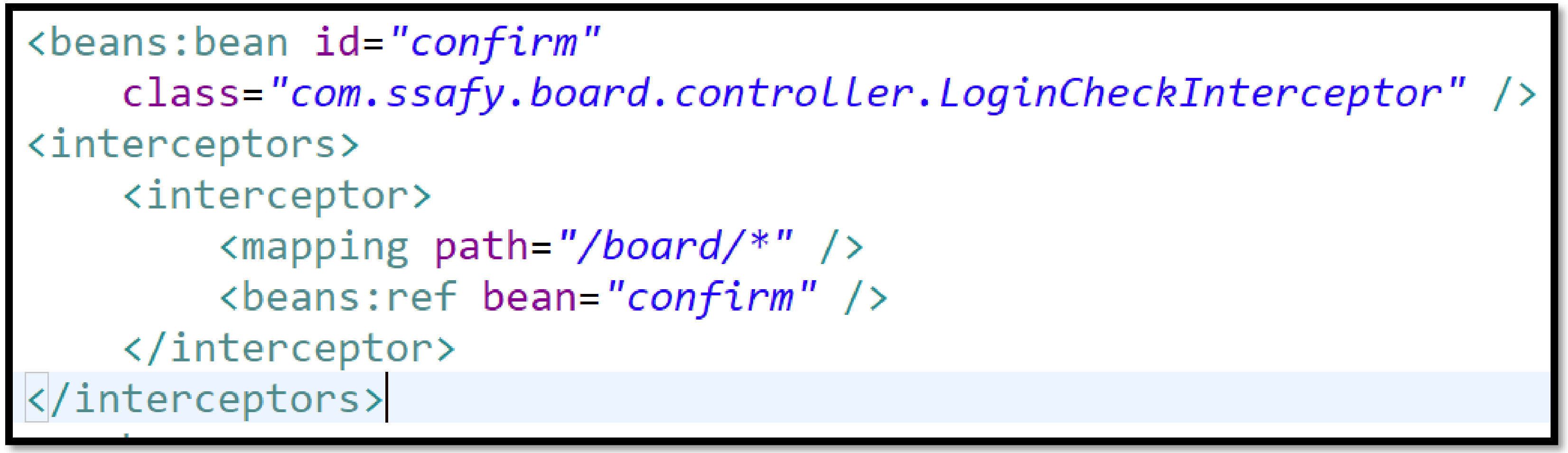
- 자바 설정 클래스 생성



```
@Configuration
@EnableWebMvc
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/**").addResourceLocations("classpath:/static/");
    }
```

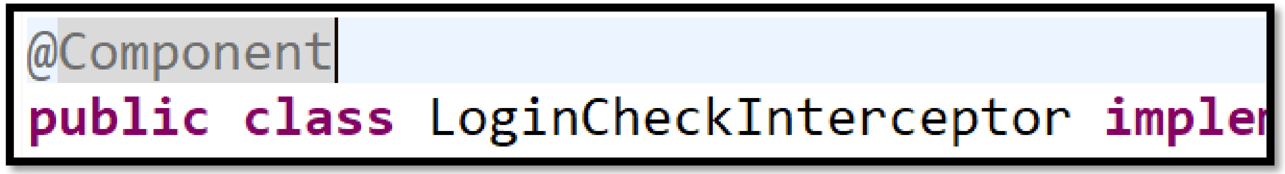
Interceptor

- servlet-context.xml



```
<beans:bean id="confirm"
    class="com.ssafy.board.controller.LoginCheckInterceptor" />
<interceptors>
    <interceptor>
        <mapping path="/board/*" />
        <beans:ref bean="confirm" />
    </interceptor>
</interceptors>
```

- 빈 등록



```
@Component
public class LoginCheckInterceptor imple
```

SpringBoot 게시판

Interceptor

- servlet-context.xml

```xml
<interceptors>
    <interceptor>
        <mapping path="/board/*" />
        <beans:ref bean="confirm" />
    </interceptor>
</interceptors>
```

- 자바 설정 클래스

```java
@Autowired
LoginCheckInterceptor loginCheckInterceptor;
@Override
public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(loginCheckInterceptor)
            .addPathPatterns("/board/*");
}
```

SAMSUNG
SW
ACADEMY
FOR
YOUTH

함께가요 미래로!
Enabling People

다음 방송에서 만나요!

삼성 청년 SW 아카데미