

Spring Boot(스프링부트)

스프링은 EJB(Enterprise Java Beans)의 무겁고 복잡한 플랫폼에서 벗어나, POJO (Plain Old Java Object) 기반의 경량화된 개발 환경을 제공하는 오픈소스 프레임워크 이다.

스프링 프레임워크가 처음 나왔을 때는 단순히 애플리케이션 운용에 필요한 객체들을 생성하고, 객체들 사이에 의존성 (Dependency)을 주입해주는 단순한 컨테이너로서의 기능만 제공했다. 하지만 현재의 스프링은 엔터프라이즈 시스템 개발에 필요한 모든 분야를 지원하는 하나의 플랫폼으로 발전했다. 또한 국내에서는 전자 정부 표준 프레임워크로 채택되면서 자바 개발자들이 알아야 할 사실상 표준이 되었다.

POJO (Plain Old Java Object)란

직역하면 오래된 방식의 간단한 자바 오브젝트라는 뜻이다.
특별한 제한에 종속되지 않고, 클래스 패스(class path)를 필요로 하지 않는 일반적인 Java Object를 의미한다.


프레임워크(Framework)


소프트웨어 애플리케이션이나 솔루션의 개발을 수월하게 하기 위해 소프트웨어의 구체적 기능들에 해당하는 부분의 설계와 구현을 재사용 가능하도록 협업화된 형태로 제공하는 소프트웨어 환경을 말한다. 즉 프레임워크란, 특정한 목적에 맞게 프로그래밍을 쉽게 하기 위한 “설계의 기본틀” 이라 말할 수 있다.

공식사이트 : <https://spring.io>

Spring Boot

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



Why Spring ▾ Learn ▾ Projects ▾ Training Support Community ▾ 

Spring Boot

Spring Framework

Spring Data >

Spring Cloud >

Spring Cloud Data Flow

Spring Security >

Spring GraphQL

Spring Session >

Spring Integration

Spring HATEOAS

Spring REST Docs

Spring Batch

Spring AMQP

Spring Boot 2.5.3

OVERVIEW

LEARN

SAMPLES

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.

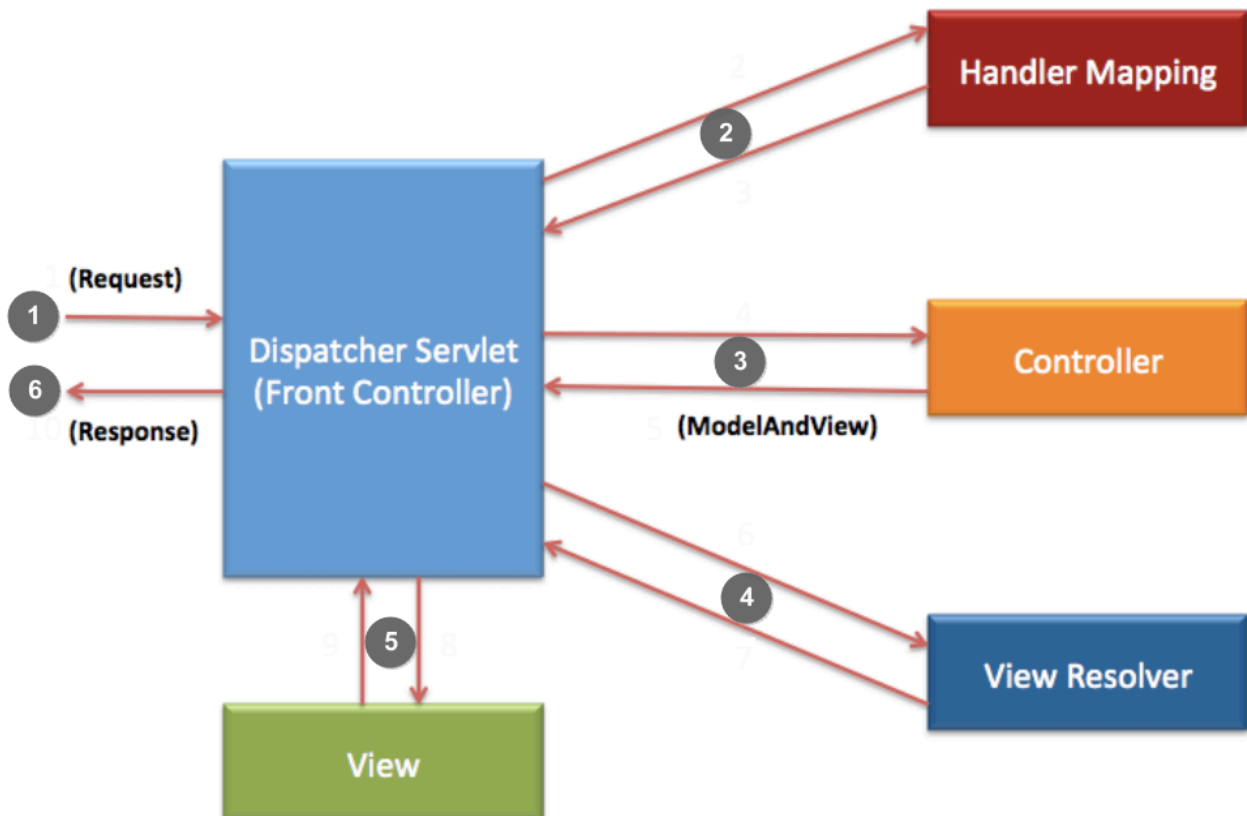
Features

- Create stand-alone Spring applications

스프링 부트의 특징

- 실행 가능한 단독 애플리케이션을 만들 수 있다.
- 외부 WAS(와스) 없이 내장된 톰캣, 제티 또는 언더토우 서버를 사용할 수 있다.
- 라이브러리 관리를 위한 스프링부트 스타터를 제공한다.
- 스프링 라이브러리와 서드 파티 라이브러리를 위한 자동설정을 지원한다.
- Xml 설정을 사용하지 않는다.

스프링 구동 과정



① 클라이언트의 요청은 **DispatcherServlet**이 제일 먼저 받는다.

- 프로젝트명Application 클래스에는 @SpringBootApplication 어노테이션이 부착되어 있다.
- 내부의 여러가지 설정 중 @ComponentScan 어노테이션은 위 클래스가 포함된 기본패키지를 scan(검색)한다.
- 이때 @Controller, @Service, @Repository, @Component 등의 어노테이션이 부착된 클래스의 빈(인스턴스)이 자동으로 생성된다.

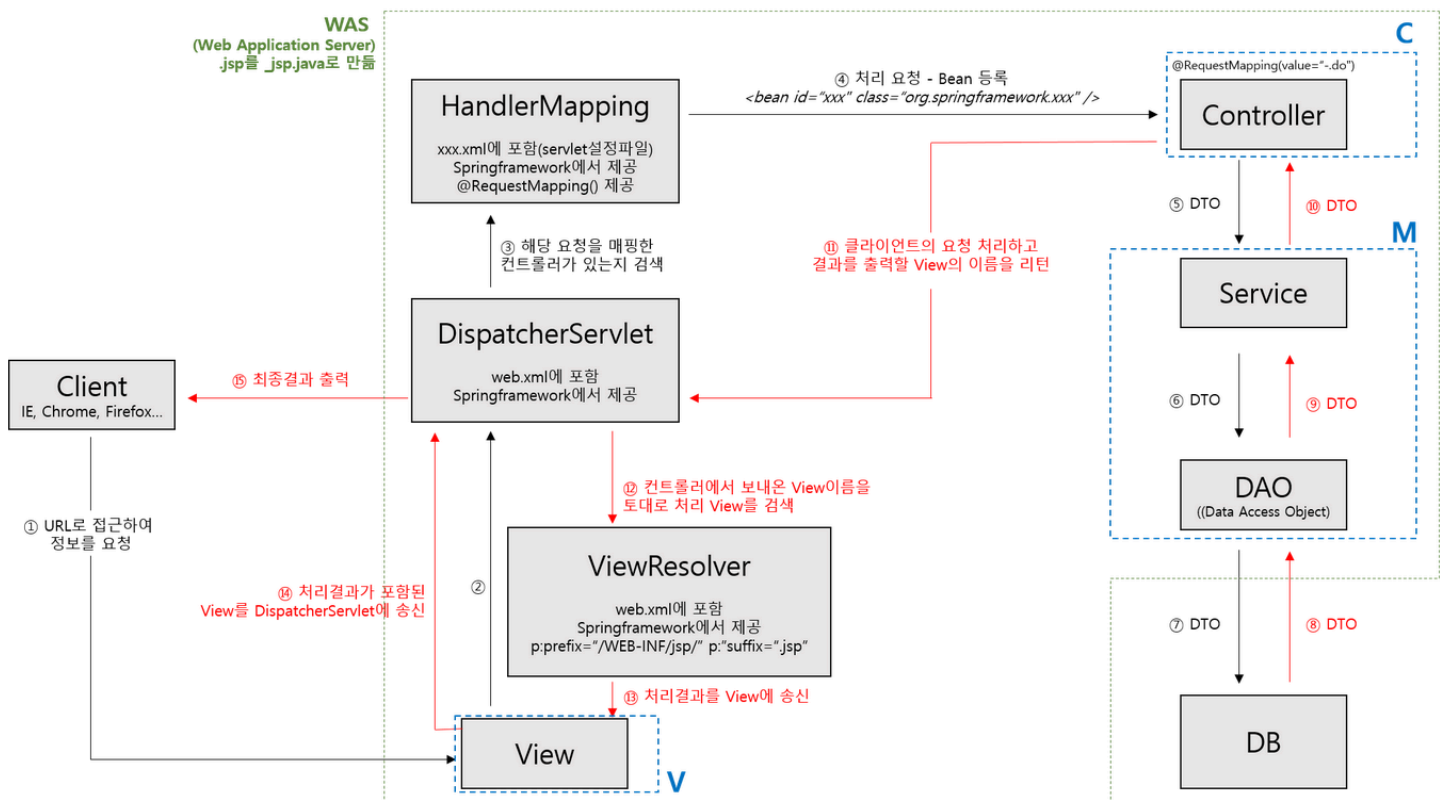
```
51 @Target(ElementType.TYPE)
52 @Retention(RetentionPolicy.RUNTIME)
53 @Documented
54 @Inherited
55 @SpringBootConfiguration
56 @EnableAutoConfiguration
57 @ComponentScan(excludeFilters = { @Filter(type = FilterType.CUSTOM, classes = TypeEx
58     @Filter(type = FilterType.CUSTOM, classes = AutoConfigurationExcludeFilter.c
59 public @interface SpringBootApplication { ←-----
```

- ② **HandlerMapping** : scan(검색)을 통해 Request URL(요청명)에 매핑된 컨트롤러를 찾는다.
 - @RequestMapping 어노테이션을 통해 설정한다.
- ③ **Controller** : 클라이언트의 요청을 처리한다.
 - @Controller 어노테이션을 통해 설정한다.
 - 처리된 결과는 **ModelAndView** 객체를 통해 정보를 저장 및 View 경로를 설정한다.
- ④ **ViewResolver** : 컨트롤러에서 반환한 View정보를 통해 실제 View의 경로를 조립한다.
 - application.properties에서 view prefix, suffix 항목으로 설정되어 있다.



- ⑤ **View** : 컨트롤러가 처리한 결과값을 웹 브라우저를 통해 출력한다.
- ⑥ 결과를 출력하면서 응답이 완료된다.

상세한 흐름. Model(Database) 포함.



스프링 부트 개발을 위한 STS4 설치

spring.io 에서 Spring Tools 4(Spring Tool Suite)를 다운로드 한다.

<https://spring.io/> > Projects > Spring Tools 4

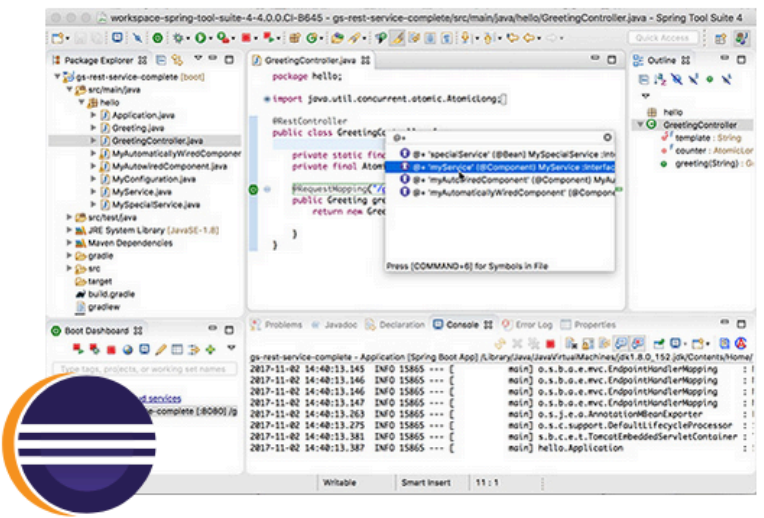
Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4.
Free. Open source.

4.11.0 - LINUX X86_64

4.11.0 - MACOS X86_64

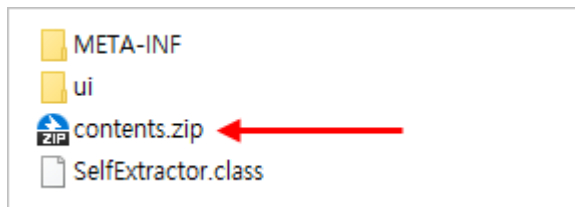
4.11.0 - WINDOWS X86_64



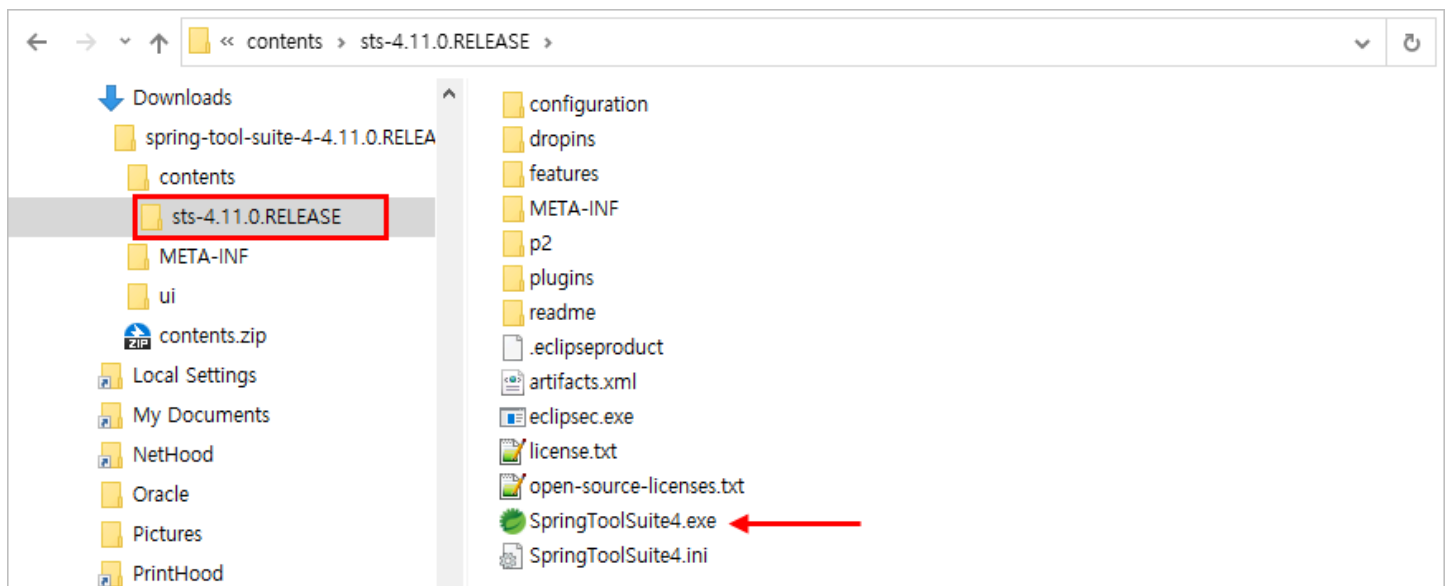
The screenshot shows the Spring Tools 4 for Eclipse download page with three operating system options. The Windows X86_64 option is highlighted with a red box. To the right, a snippet of Java code is shown, featuring a class named GreetingController with methods like @RequestMapping and public Greeting get.

Spring Tool Suite 4는 이클립스와 동일하지만 Spring Boot를 개발하기 위한 모든 설치가 완료된 상태로 배포되는 버전이다.

다운로드 받으면 확장자가 zip 이 아니라서 조금 당황(?)스러울 수도 있다. 하지만 압축 해제 프로그램으로 압축을 풀면 된다.



1차로 압축을 푼 후 contents.zip을 한번 더 압축 해제 해야한다.

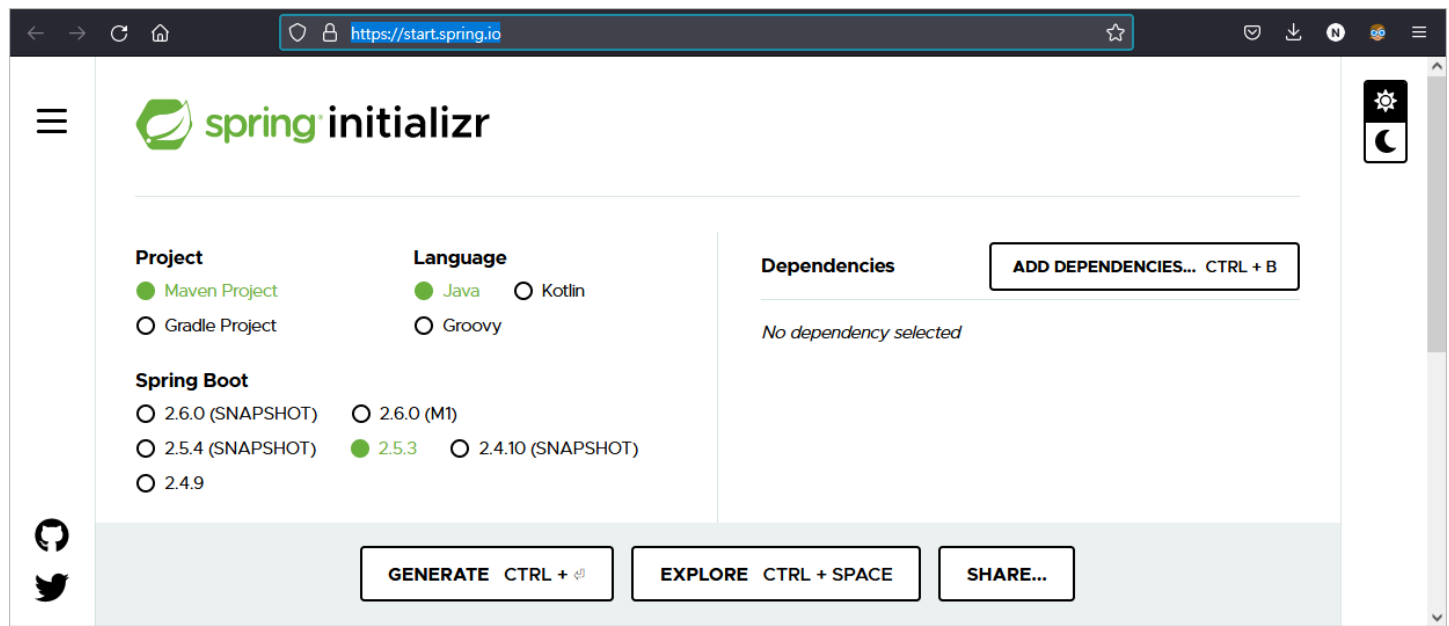


sts-4.xx.x.RELEASE 폴더를 01DevelopKits 하위로 이동한다.

SpringToolSuite4.exe 를 실행한다.

첫번째 프로젝트 생성

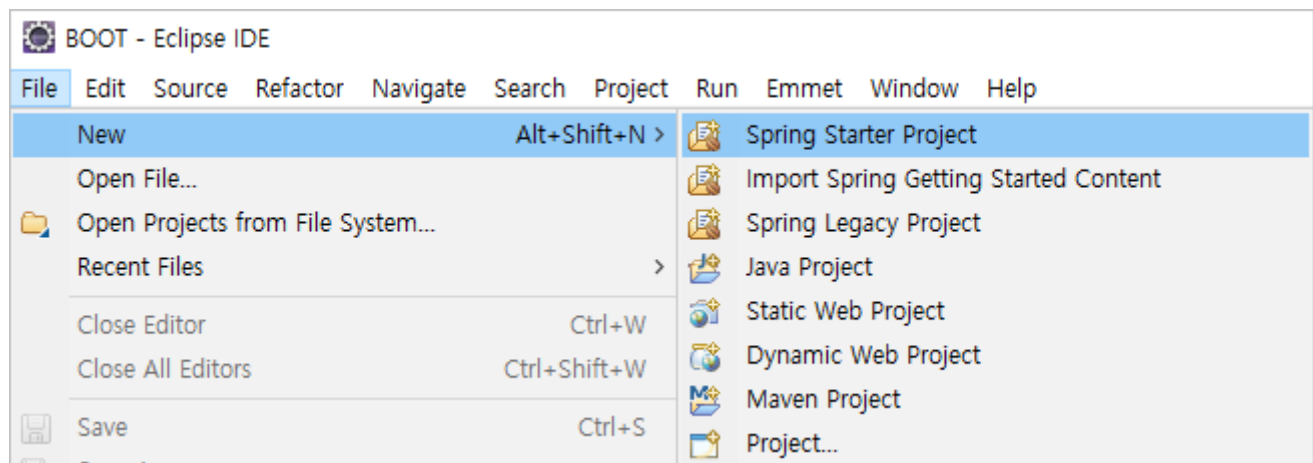
스프링 부트 프로젝트는 아래와 같이 <https://start.spring.io/> 에 접속한 후 생성할 수 있다.



이렇게 생성한 프로젝트는 압축을 푼 후 Import > Gradle > Existing Gradle Project로 импорт하면 된다.

하지만 우리는 기존에 사용했던 이클립스를 통해 첫번째 프로젝트를 생성해 보겠다. 이클립스를 이용하면 우리의 설정값을 GET방식으로 전송한 후, 프로젝트 파일을 다운로드 하고 импорт까지 자동으로 해주므로 훨씬 편리하게 처리할 수 있다.

메뉴 > File > New > Spring Starter Project 를 선택



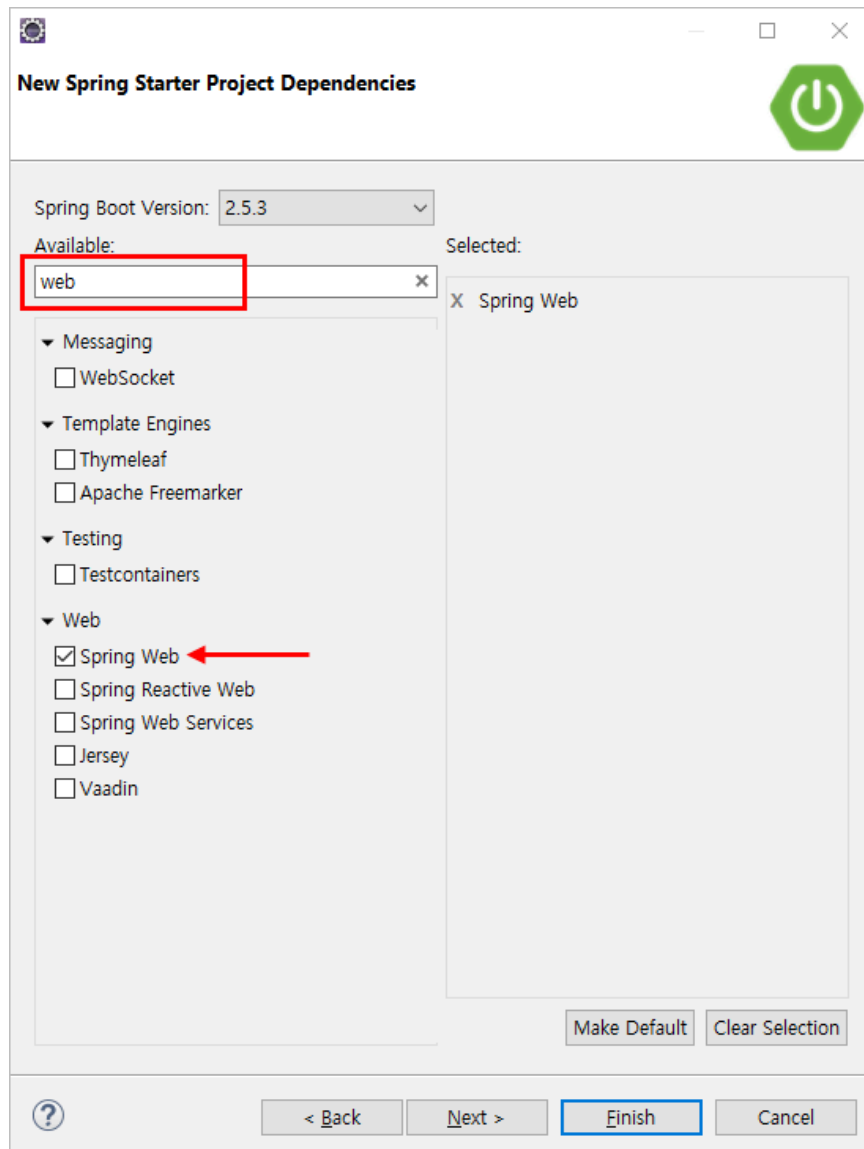
프로젝트명 : B00SpringBootProject

Name : 프로젝트의 이름
Type : 라이브러리 관리 및 빌드 도구
Packaging : 패키징 파일의 형식
Group : 프로젝트를 만들 회사명
Artifact : 산출물의 이름.
일반적으로 프로젝트명과 동일함.
Package : 프로젝트의 기본 패키지명

패키징을 War로 선택하면 배포했을 때에도 JSP가 정상 동작한다. 만약 Jar로 선택하면 JSP가 동작하지 않는다.

Next를 누르면 의존성을 추가할 수 있다.

Available에서 web을 검색한다. 여기서 Spring Web을 선택하면 웹 애플리케이션 개발에 필요한 라이브러리들이 프로젝트에 포함된다.

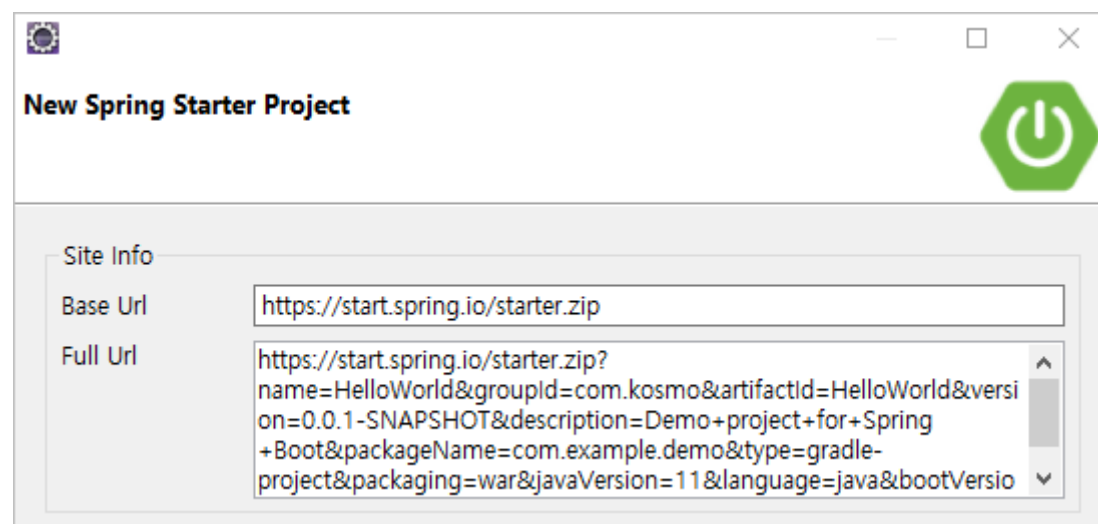


The dialog box titled "New Spring Starter Project Dependencies" shows the "Spring Boot Version" set to 2.5.3. In the "Available:" list, "web" is selected and highlighted with a red box. Below it, the "Web" category is expanded, and "Spring Web" is checked with a red arrow pointing to it. The "Selected:" list on the right shows "X Spring Web". At the bottom, there are buttons for "< Back", "Next >", "Finish" (highlighted with a blue border), and "Cancel".

Spring Boot Version

22년 11월	2.5.3
23년 10월	3.1.5
24년 8월	3.3.2

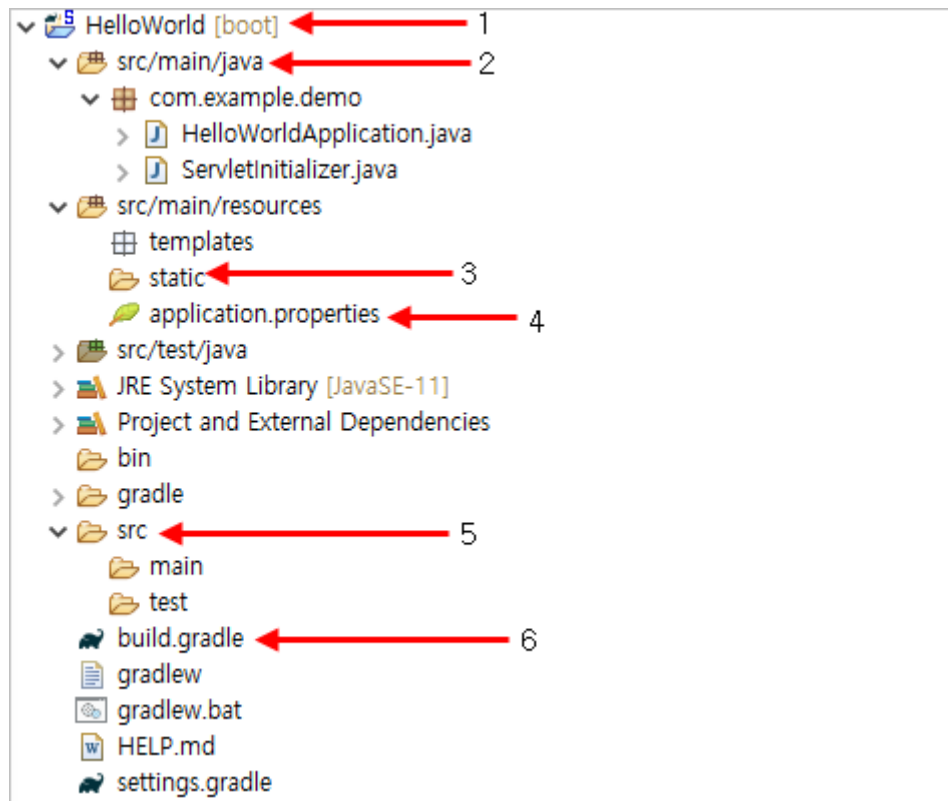
Next를 눌러보면 우리가 선택한 항목이 Full Url 부분에 쿼리스트링 형태의 파라미터로 생성된것을 볼 수 있다.



The dialog box titled "New Spring Starter Project" shows the "Site Info" section. The "Base Url" is "https://start.spring.io/starter.zip". The "Full Url" is a long string with many query parameters, including "name=HelloWorld", "groupId=com.kosmo", "artifactId=HelloWorld", "version=0.0.1-SNAPSHOT", "description=Demo+project+for+Spring+Boot", "packageName=com.example.demo", "type=gradle-project", "packaging=war", "javaVersion=11", and "language=java".

Finish를 눌러 프로젝트를 생성한다.

생성된 프로젝트의 구조는 다음과 같다.



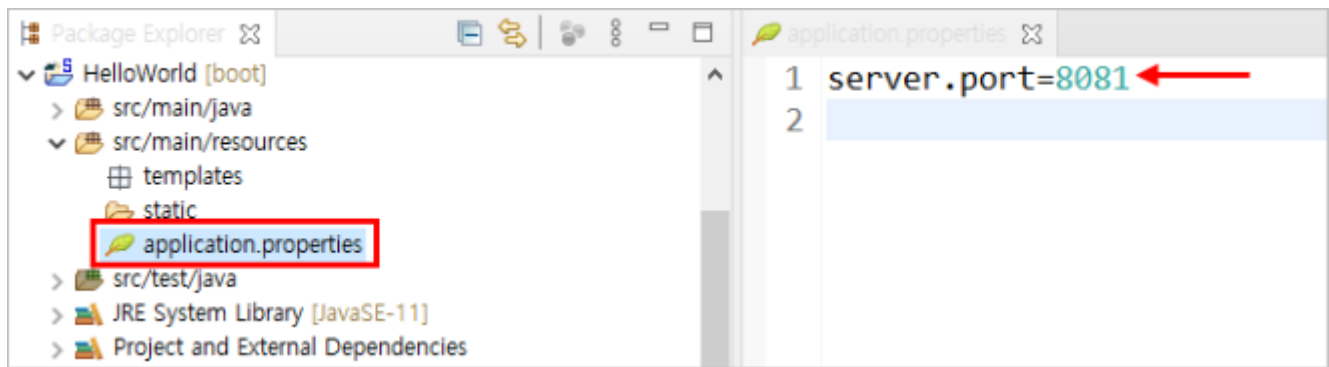
- 1번 : 프로젝트 생성 및 임포트가 완료되면 우측에 [boot]라는 표시가 붙는다.
- 2번 : 기본패키지가 포함된 자바 소스 파일을 만들 위치
- 3번 : HTML, CSS, JS 및 이미지 파일 등과 같은 정적 웹 리소스를 저장할 위치
- 4번 : 프로젝트 전체에서 사용할 여러가지 속성을 저장하는 파일. Port번호, View파일의 위치 등
- 5번 : 동적 웹 리소스인 JSP를 저장할 위치
- 6번 : 라이브러리 의존설정 및 프로젝트 전반의 환경설정을 저장하는 파일.

프로젝트 실행

포트번호변경

오라클11g XE가 설치되어 있다면 8080포트를 사용할 수 없다. 그게 아니더라도 8080은 다른 웹 애플리케이션에서도 주로 사용하는 포트번호이므로 변경하는 것이 좋다.

application.properties 열고 다음과 같이 입력한다.

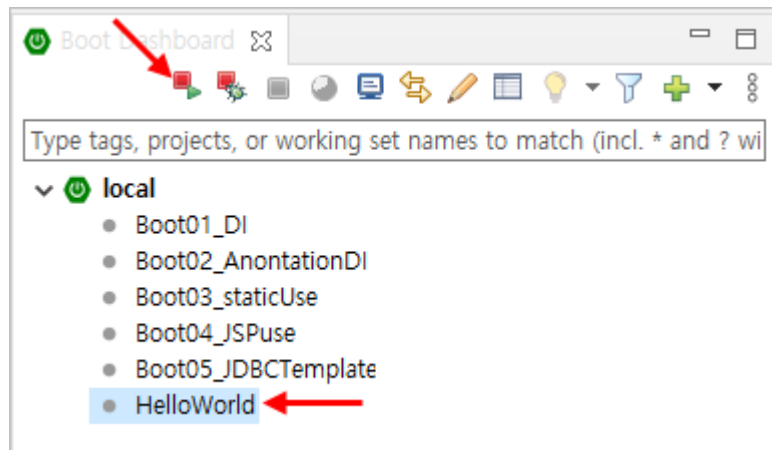


Boot Dashboard

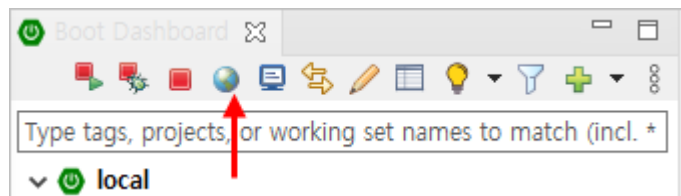
스프링 부트는 실행을 위해 Boot Dashboard 가 필요하다.



프로젝트를 선택하면 Start 버튼이 활성화된다.



프로젝트가 시작되면 아래와 같이 포트번호가 표시되면서 지구본 모양의 아이콘도 활성화된다.
아이콘을 눌러서 웹 브라우저를 실행한다.



하지만 아직까지 아무것도 작성한 것이 없으므로 에러가 발생한다.

