



spring boot기초

안 화 수

# Spring boot의 특징

## ❖ spring boot 의 특징

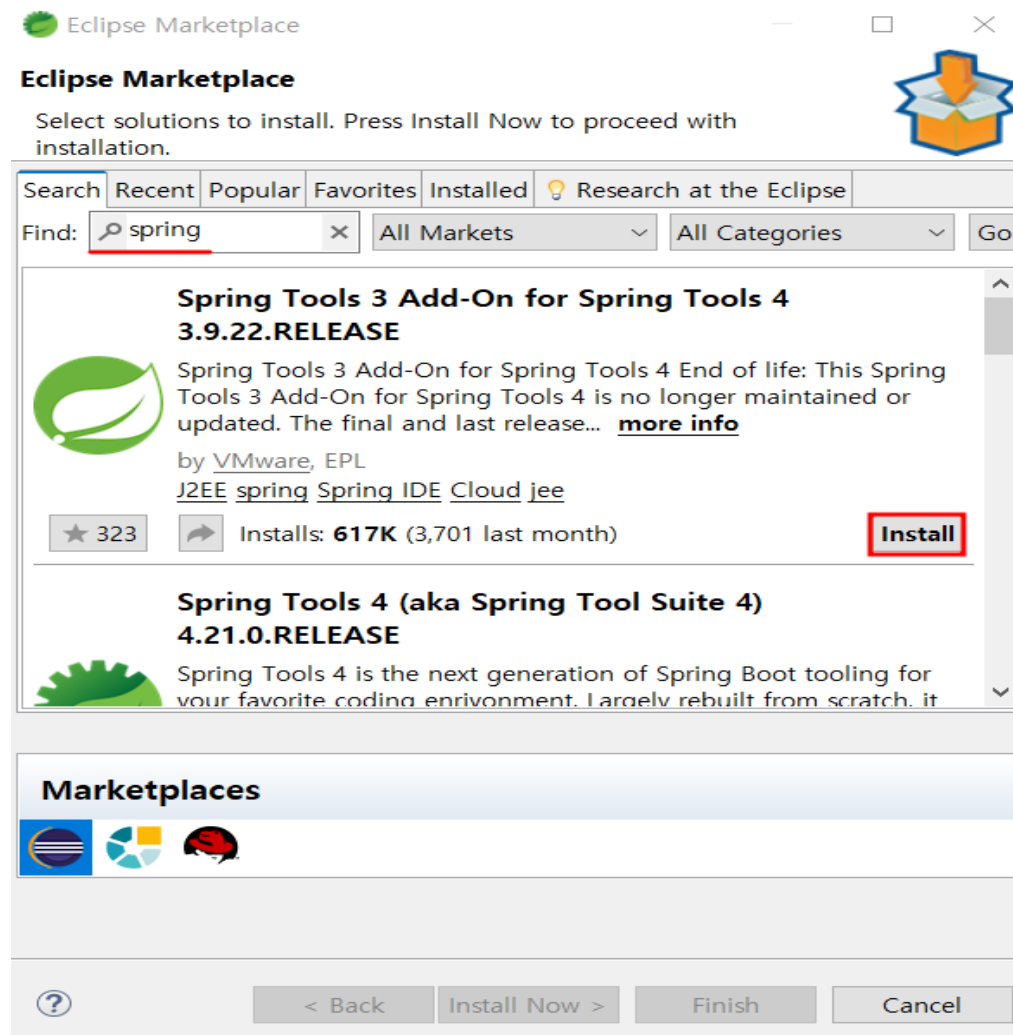
- 독립 실행이 가능한 스프링 애플리케이션 개발 가능(Tomcat, Jetty 내장)
- 통합 Starter를 이용하여 Maven/Gradle 로 라이브러리 관리
- Starter를 통한 자동화된 스프링 설정 제공
- 번거로운 XML 설정을 요구하지 않음
- Spring Actuator 제공 (애플리케이션의 모니터링과 관리를 위해서 사용)

# spring boot환경구축

## ❖ Eclipse에 STS 3.x plug-in 설치

[Help] - Marketplace

Find : spring 검색



# spring boot환경구축

## ❖ STS(Spring Tool Suite)

### 1. STS4 다운로드

<https://spring.io/tools> 접속

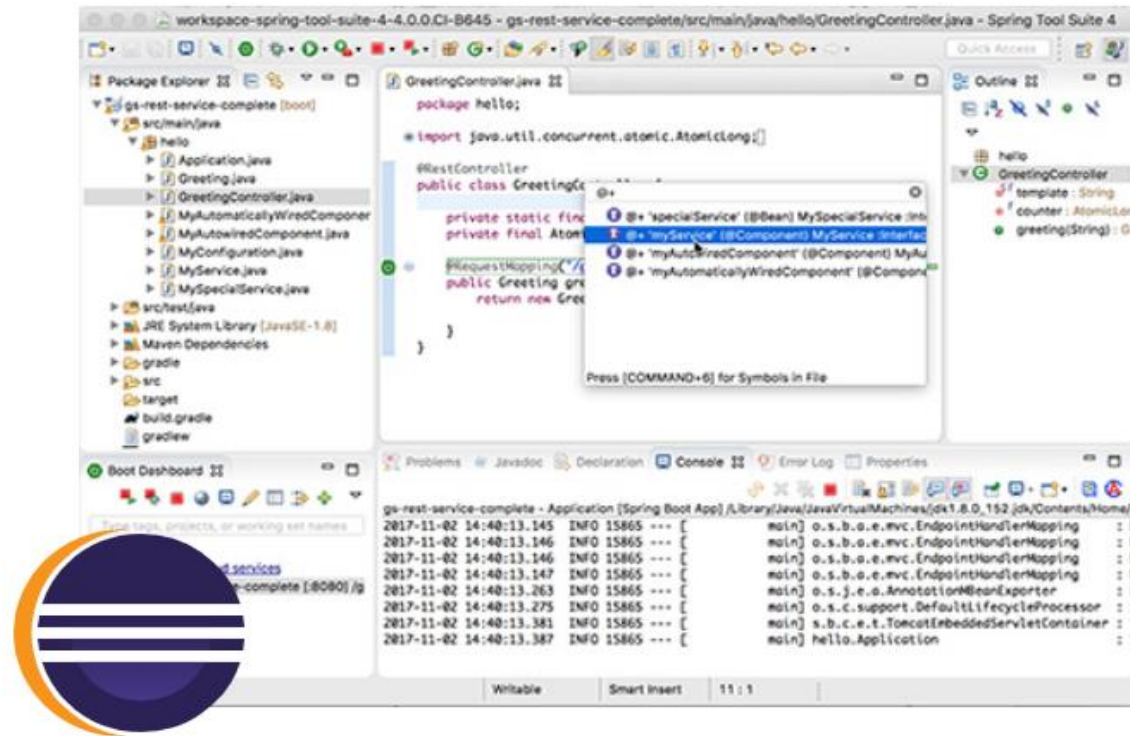
## Spring Tools 4 for Eclipse

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

LINUX 64-BIT

MACOS 64-BIT

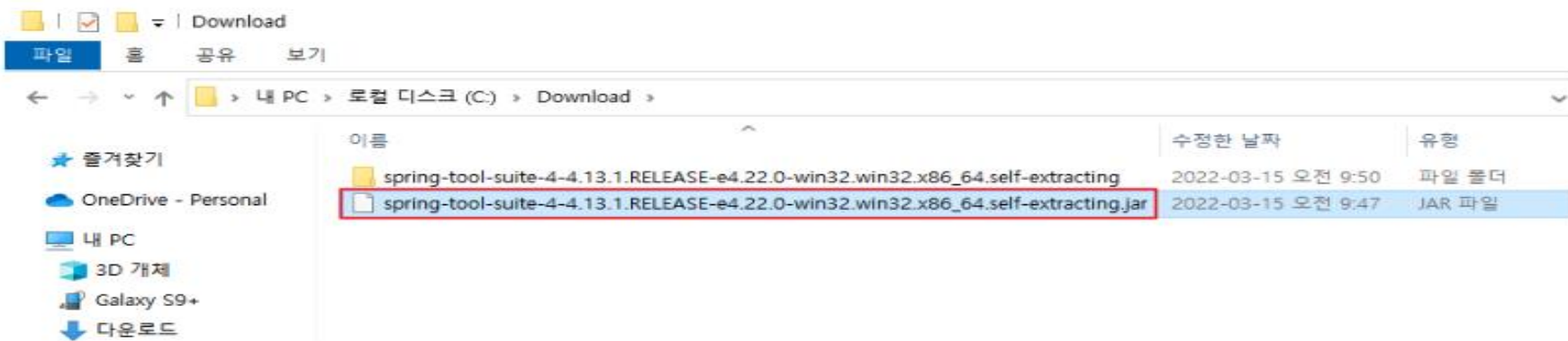
WINDOWS 64-BIT



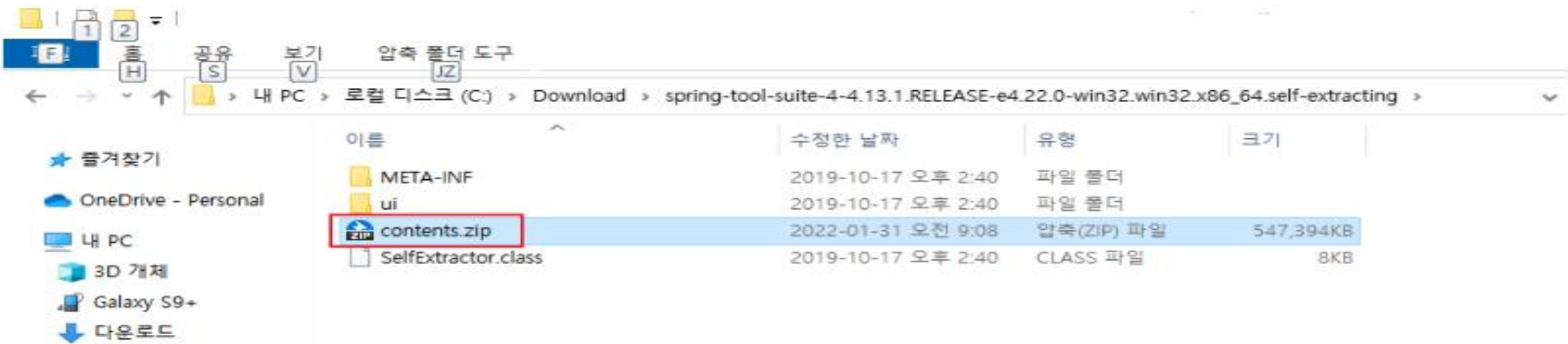
# spring 환경구축

## ❖ STS 4.x 압축해제 및 실행

1. 다운로드 받은 jar 파일을 압축 해제 한다.



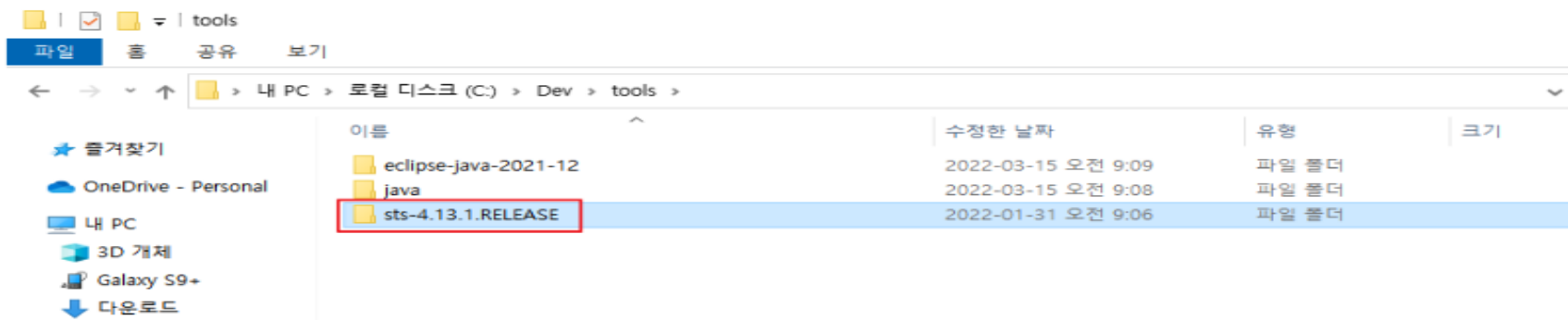
2. 다시 한번 contents.zip 파일을 압축 해제한다.



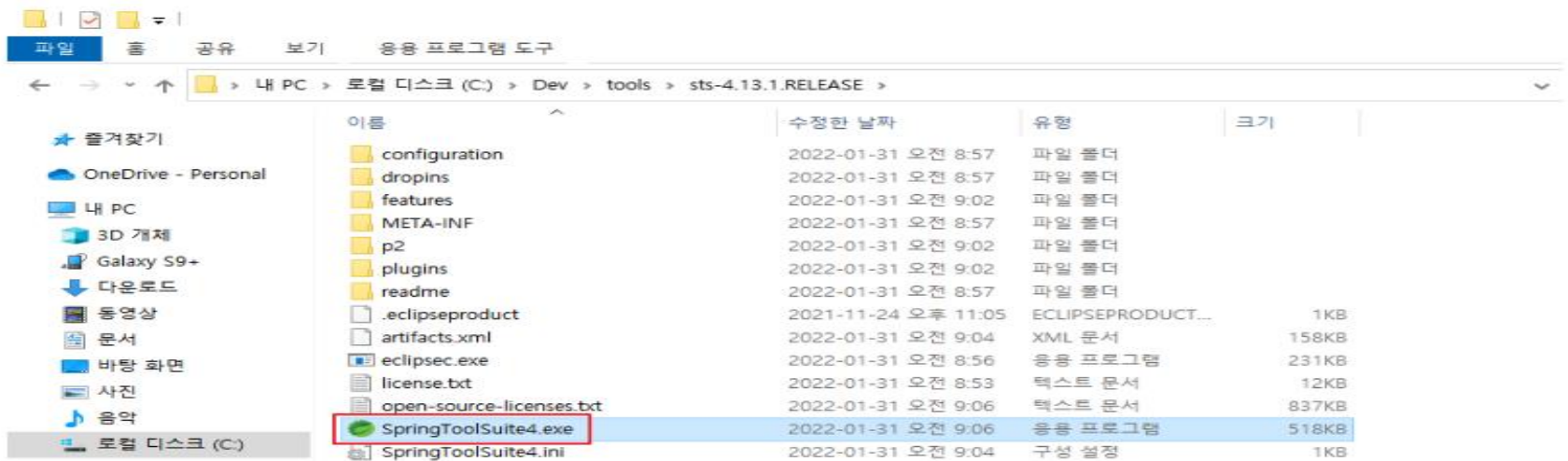
# spring 환경구축

## ❖ STS 4.x 압축해제 및 실행

3. 압축 해제된 contents 폴더로 들어간다.



4. sts4 실행파일을 클릭해서 실행한다.

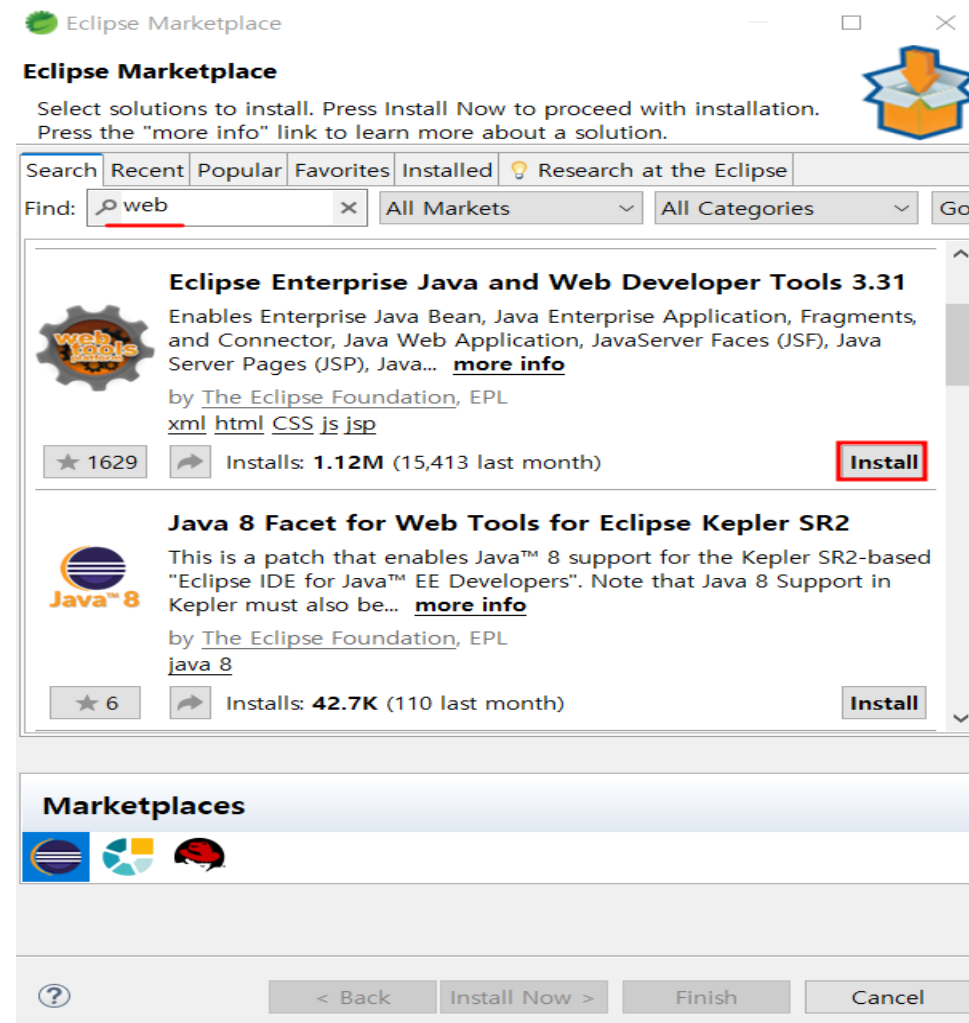


# spring 환경구축

❖ STS4에 JSP를 사용하기 위해서 plugin 추가하기

[Help] - MarketPlace

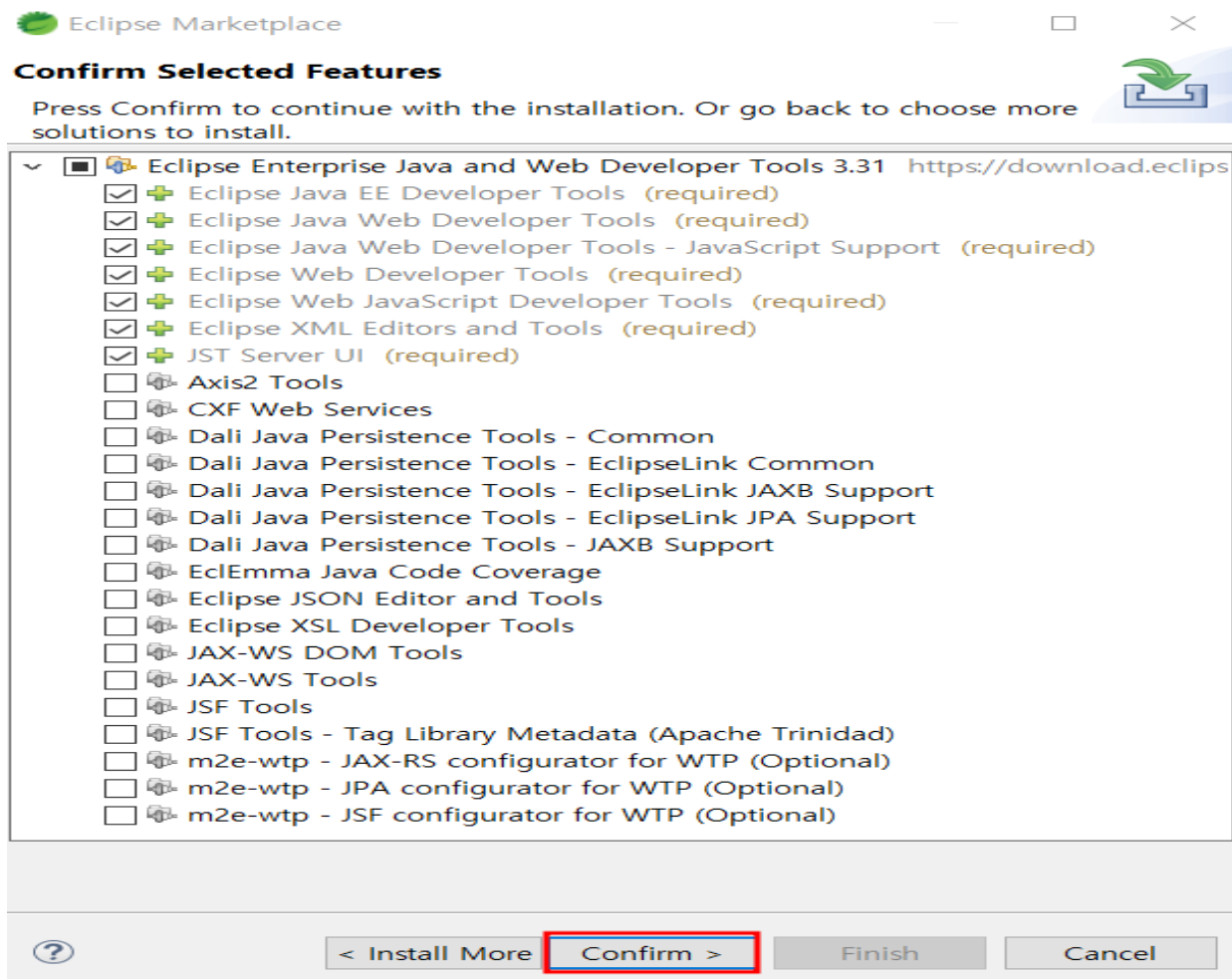
Find : web 검색





# spring 환경구축

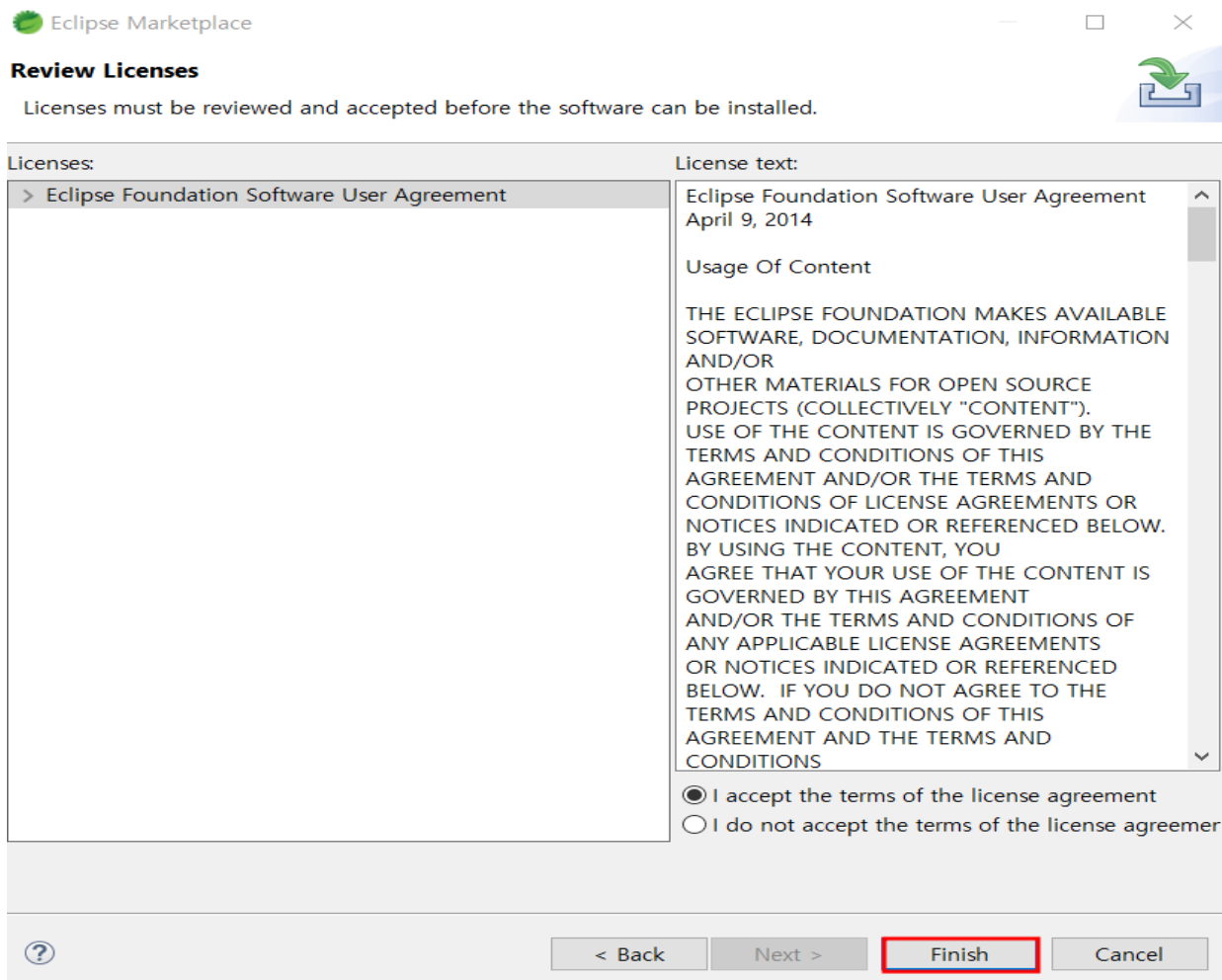
## ❖ STS4에 JSP를 사용하기 위해서 plugin 추가하기





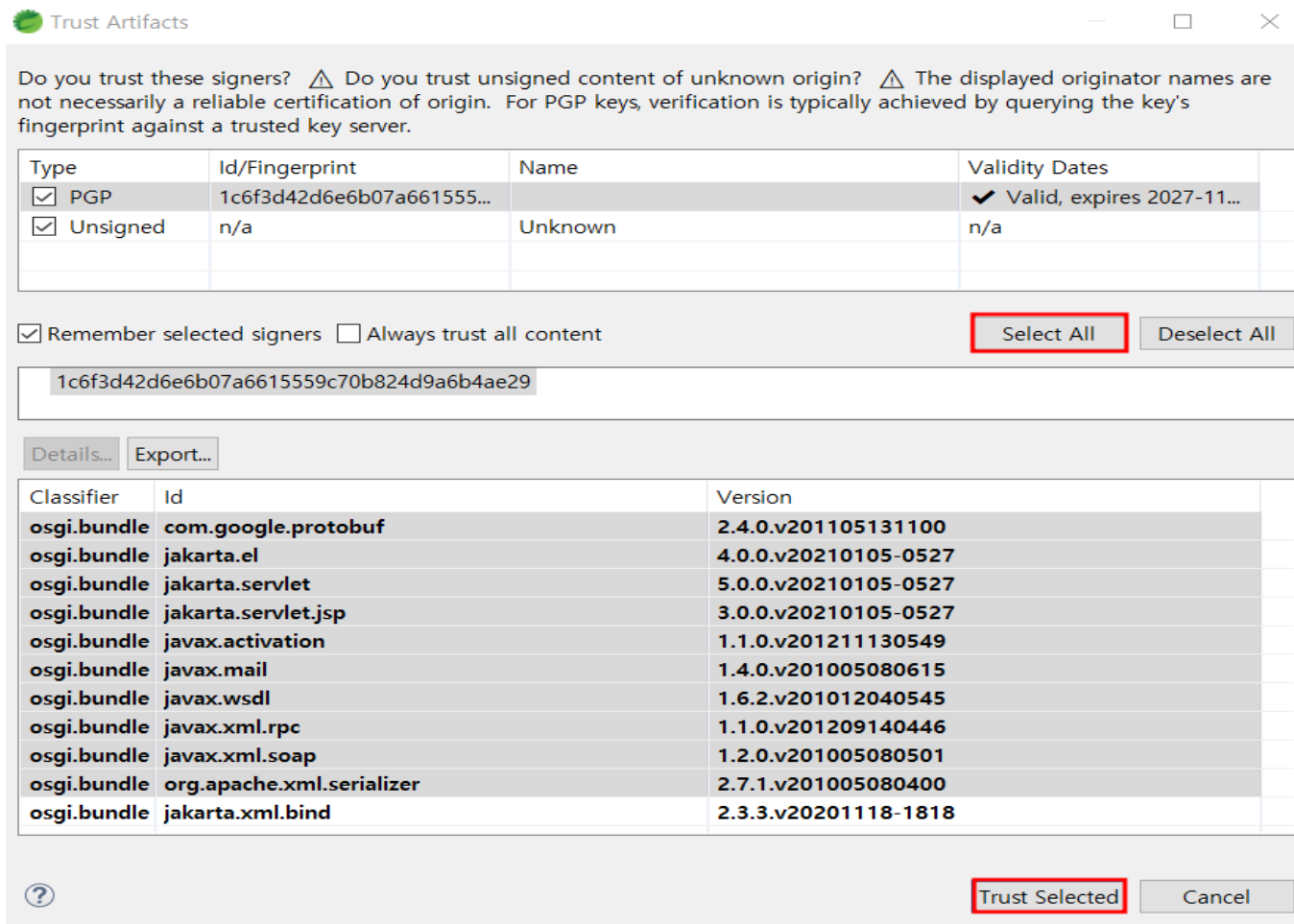
# spring 환경구축

## ❖ STS4에 JSP를 사용하기 위해서 plugin 추가하기



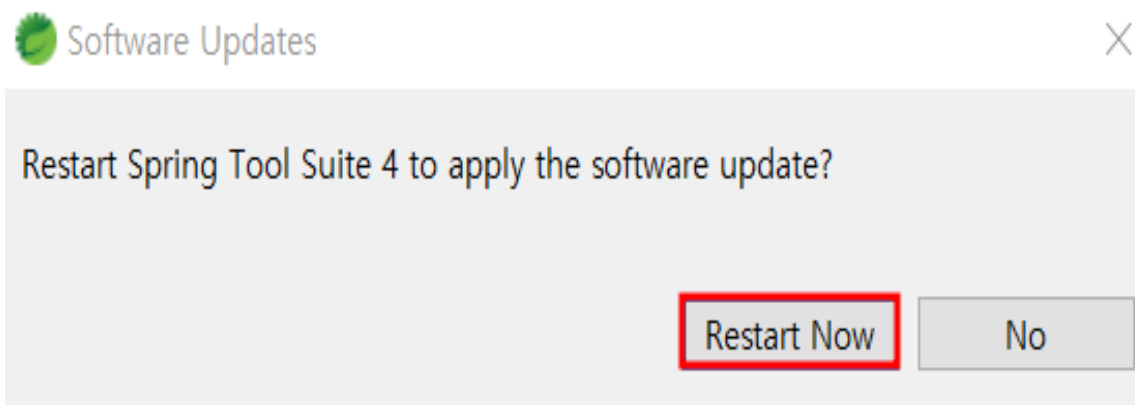
# spring 환경구축

## ❖ STS4에 JSP를 사용하기 위해서 plugin 추가하기



# spring 환경구축

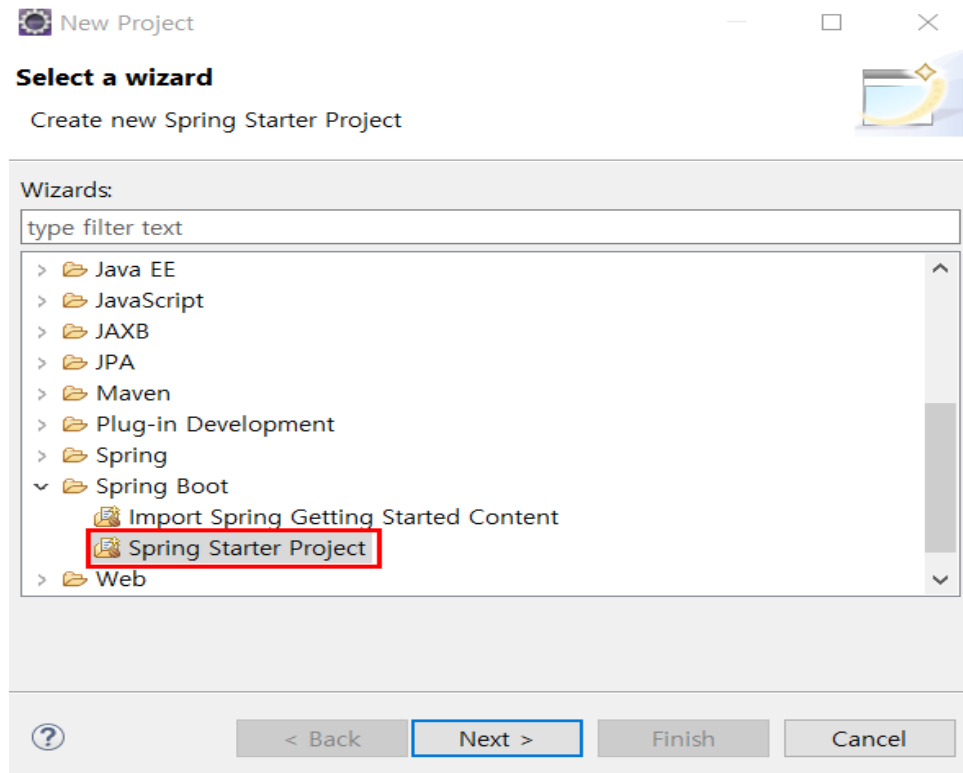
- ❖ STS4에 JSP를 사용하기 위해서 plugin 추가하기



# spring boot 프로젝트

## ❖ demo project 생성

[File] – New - Project



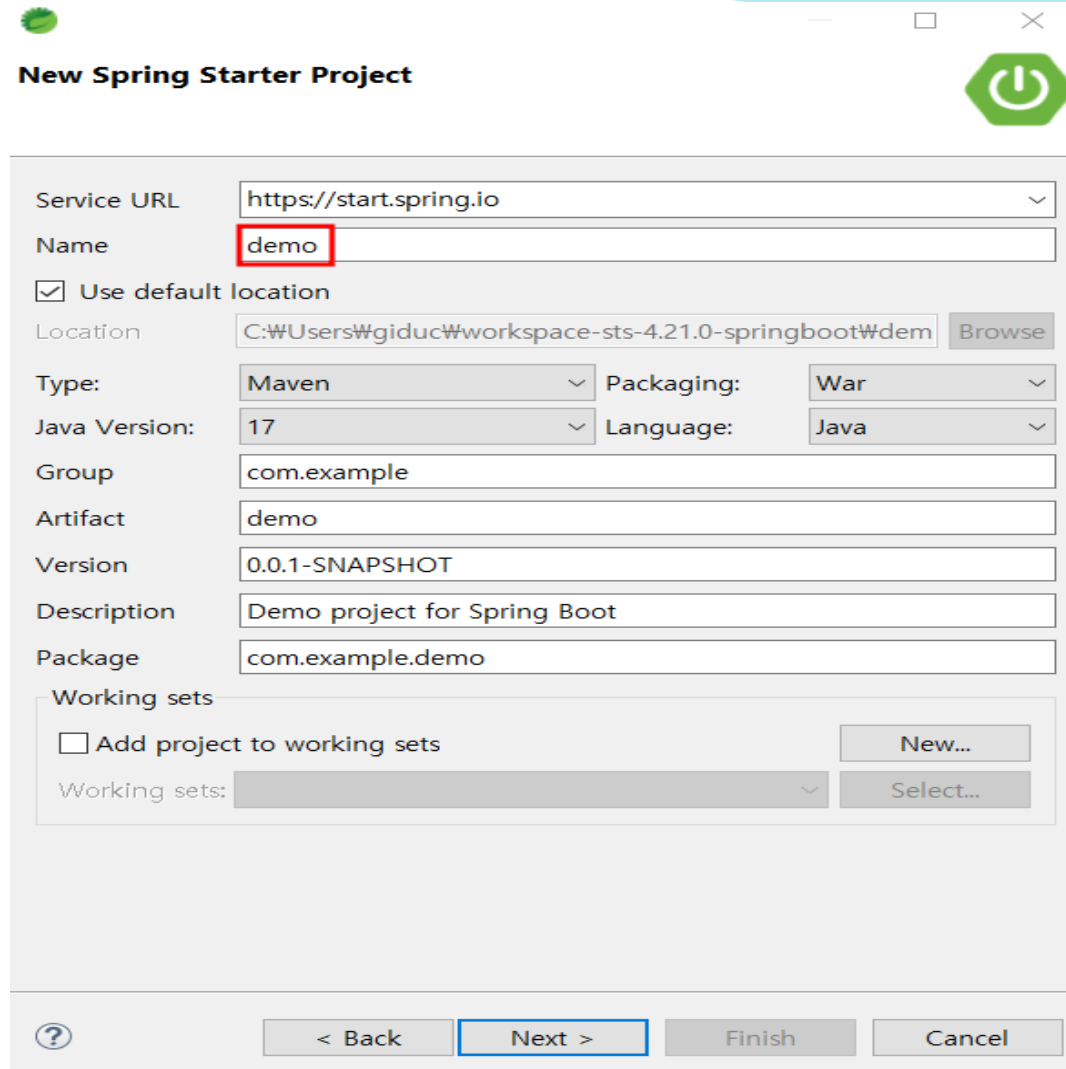
# spring boot 프로젝트

## ❖ demo project 생성

➤ Name : **demo**

➤ Type : **Maven**, Gradle

➤ Packaging : **War**, Jar



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type:  Packaging:

Java Version:  Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

☐ Add project to working sets

Working sets:

# spring boot 프로젝트

❖ demo project 생성

➤ Web

Spring Web 체크



New Spring Starter Project Dependencies



Spring Boot Version: 3.1.7

Frequently Used:

☐ Lombok ☐ Spring Data JPA ☒ Spring Web

Available:

Type to search dependencies

- ▶ Spring Cloud Config
- ▶ Spring Cloud Discovery
- ▶ Spring Cloud Messaging
- ▶ Spring Cloud Routing
- ▶ Template Engines
- ▶ Testing
- ▶ VMware Tanzu Application Service
- ▼ Web
  - ☒ Spring Web
  - ☐ Spring Reactive Web
  - ☐ Spring for GraphQL
  - ☐ Rest Repositories

Selected:

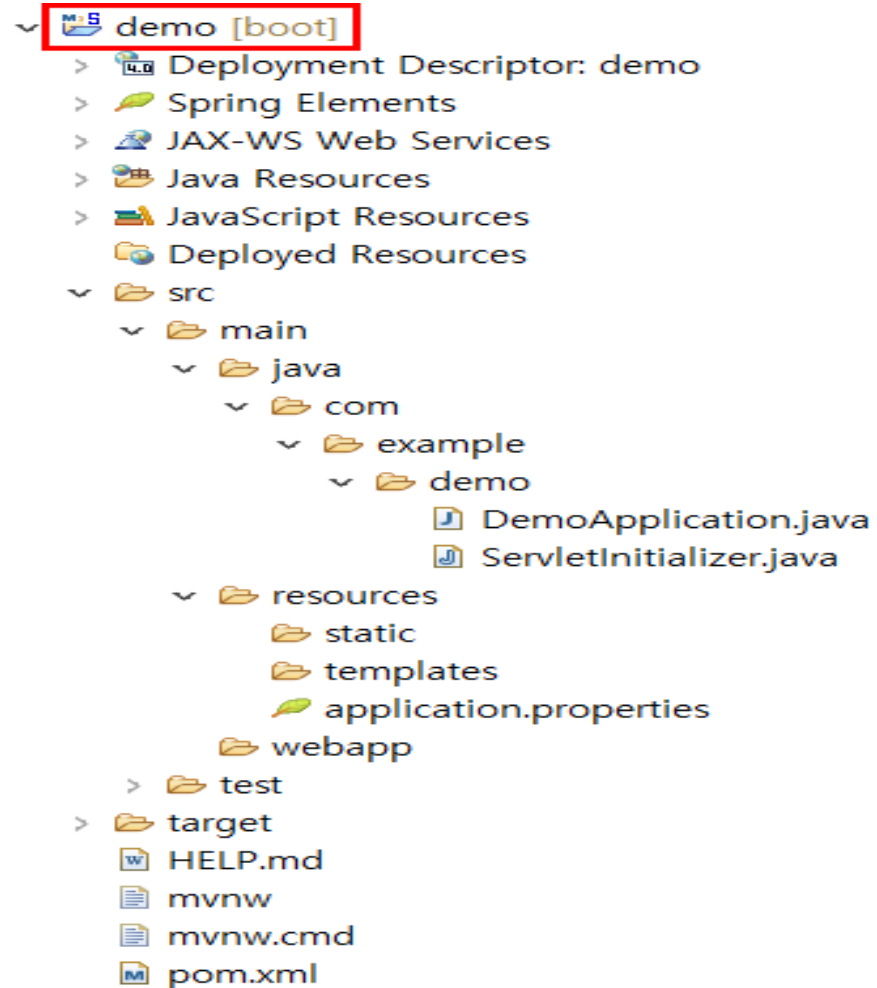
X Spring Web

Make Default Clear Selection

? < Back Next > Finish Cancel

# spring boot 프로젝트

## ❖ demo project 생성





# spring boot 서버 실행

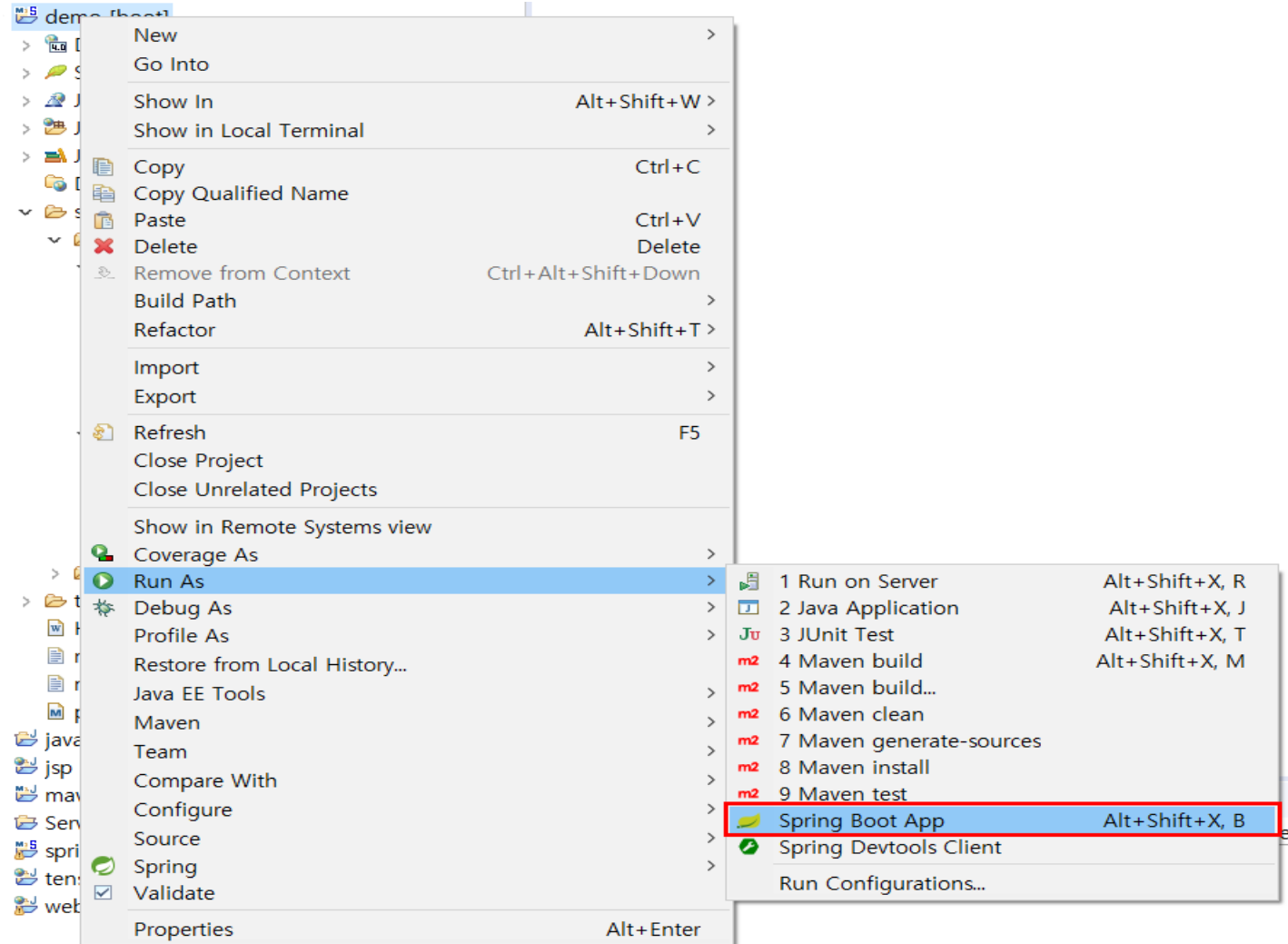
## ❖ demo project 실행

내장 Tomcat 실행

demo - 오른마우스 클릭

- Run As

- Spring Boot App



# spring boot 서버 실행

## ❖ demo project 실행 결과

다음과 같이 출력되면 성공~!!

```
demo - DemoApplication [Spring Boot App]

  ____
 /    \ ____
(  ___/  __/
 \___/   \___/

:: Spring Boot ::                (v2.2.6.RELEASE)

2020-05-03 02:37:45.844 INFO 21636 --- [main] com.example.demo.DemoApplication
2020-05-03 02:37:45.847 INFO 21636 --- [main] com.example.demo.DemoApplication
2020-05-03 02:37:46.565 INFO 21636 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer
2020-05-03 02:37:46.572 INFO 21636 --- [main] o.apache.catalina.core.StandardService
2020-05-03 02:37:46.573 INFO 21636 --- [main] org.apache.catalina.core.StandardEngine
2020-05-03 02:37:46.654 INFO 21636 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/]
2020-05-03 02:37:46.655 INFO 21636 --- [main] o.s.web.context.ContextLoader
2020-05-03 02:37:46.786 INFO 21636 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor
2020-05-03 02:37:46.925 INFO 21636 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer
2020-05-03 02:37:46.928 INFO 21636 --- [main] com.example.demo.DemoApplication
```

# Controller 생성

❖ Controller를 추가해서 Hello World 출력

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

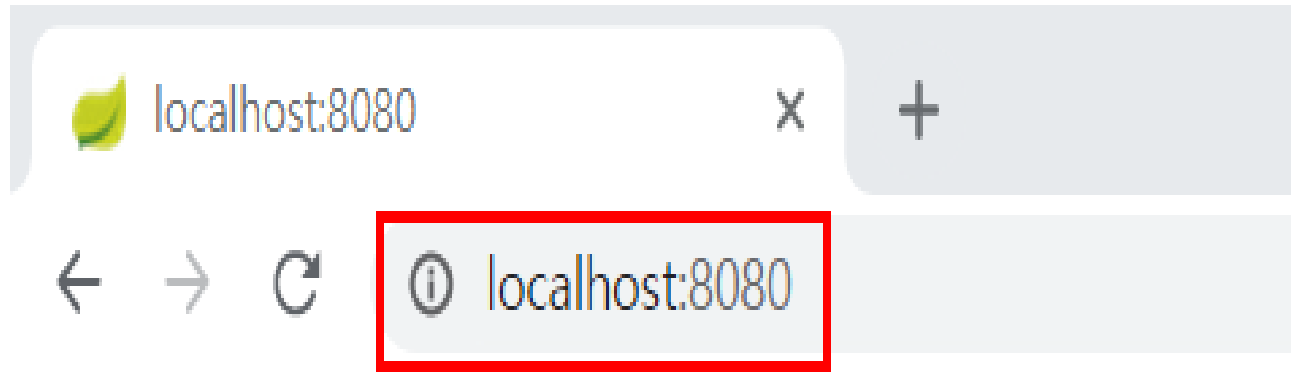
@RestController
public class SampleController {

    @RequestMapping("/")
    public String hello() {
        return "Hello World~!!";
    }
}
```

# demo 프로젝트 접속

## ❖ Controller를 추가해서 Hello World 출력

demo project를 중지하고, 재시작 한 후에 웹브라우저에 <http://localhost:8080> 요청한다.



Hello World~!!

# port 번호 설정

## ❖ port 번호 설정

spring boot 에 내장된 tomcat은 8080 이 기본 port 번호로 되어 있지만, 필요에 따라서 port 번호를 변경할 수 있다.

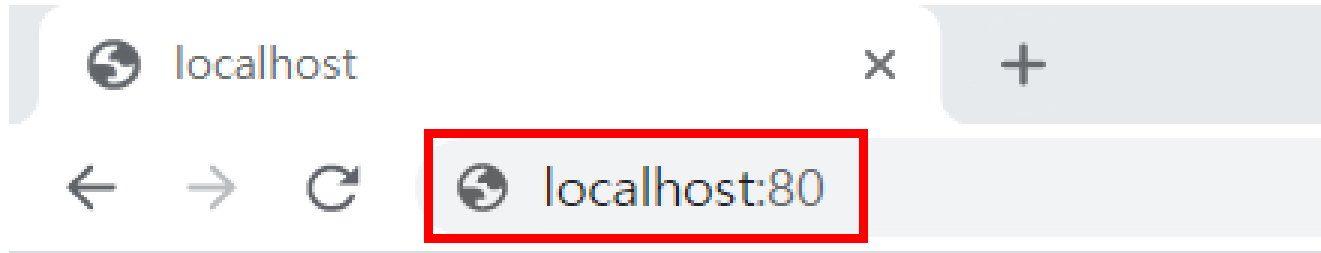
## ➤ main / resources / application.properties

```
server.port = 80
```

# demo 프로젝트 접속

## ❖ port 번호 변경후 Hello World 출력

demo project를 중지하고, 재시작 한 후에 웹브라우저에 `http://localhost:80` 요청한다.



Hello World~!!

# Properties와 Yaml(야믈)

## ❖ application.properties와 application.yml

### ■ application.properties

server.port=80

### ■ application.yml

server:

port: 80



# Properties와 Yaml(야믈)

❖ Properties와 Yaml 변환 사이트

<https://mageddo.com/tools/yaml-converter>

## Yaml to properties / Properties to Yaml converter

<< to yaml

to properties >>

Yaml

```
server:  
  port: '80'
```

convert >>

<< convert

Properties

```
# port  
server.port=80
```

<< convert

convert >>