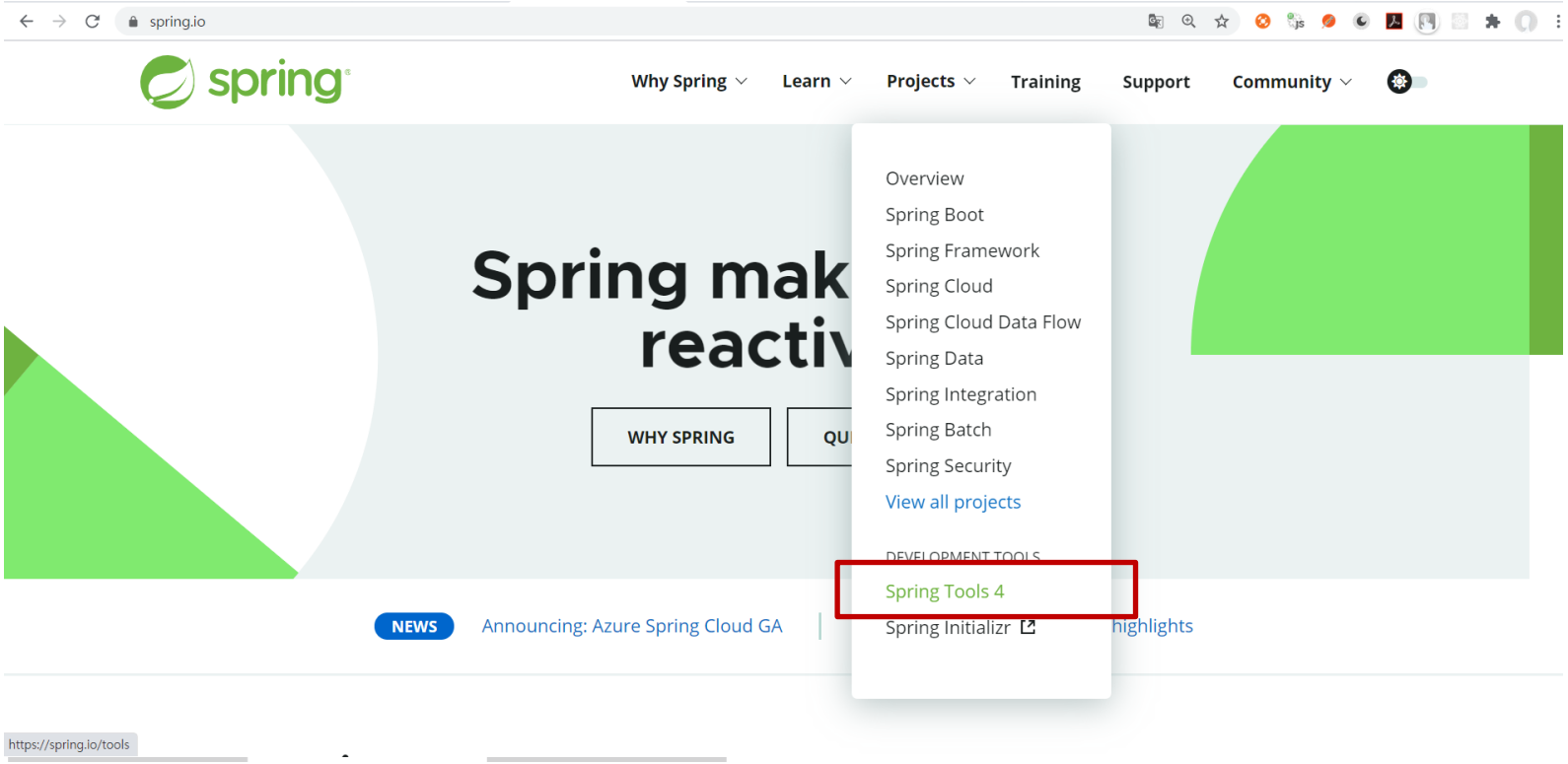
스프링 부트

스프링 부트

- http://spring.io



스프링 부트

spring.io/tools

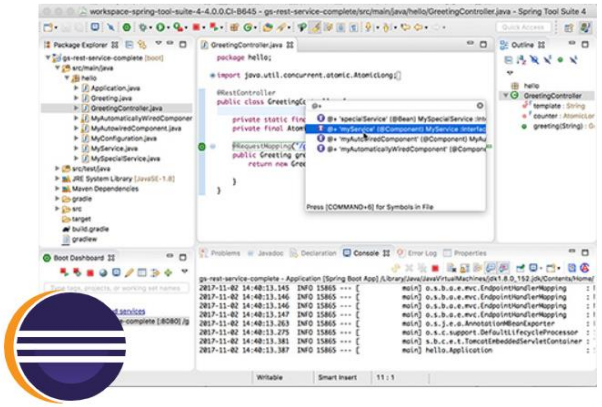
Spring Tools 4 for Eclipse

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

4.8.0 - LINUX 64-BIT

4.8.0 - MACOS 64-BIT

4.8.0 - WINDOWS 64-BIT

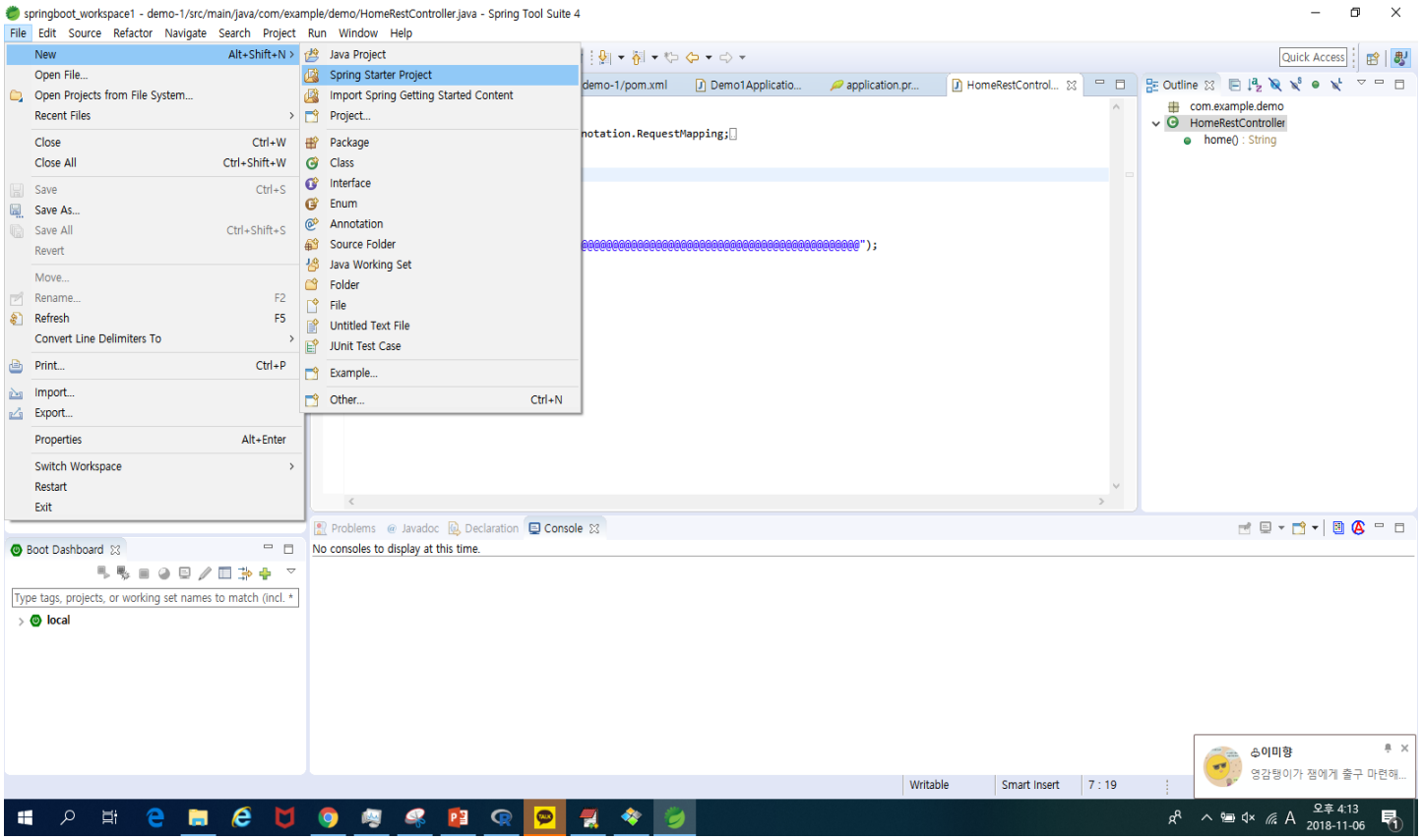


스프링 설치

- 파일을 더블클릭하면 jar파일의 압축이 풀립니다
- 만일 알집이 설치되어 있으면 jar파일이 실행되는 것이 아니고 압축이 풀려서 제대로 동작하지 않으므로 알집을 삭제후 다시 실행해 주세요 안될경우 (jarfix.exe) 를 다운받아 실행 후 다시 실행한다
- 스프링 3으로 개발할 목적이 아니면 별도의 톰캣을 설치 하지 않아도 됩니다.

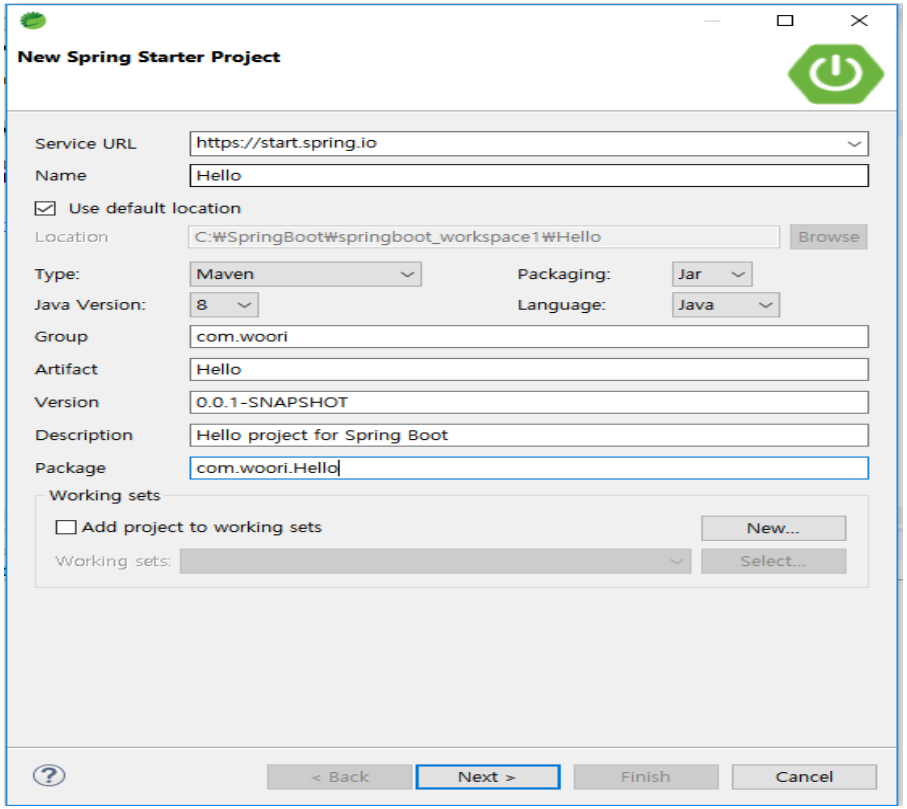
Hello, Web Application

- **file – new – spring starter project**

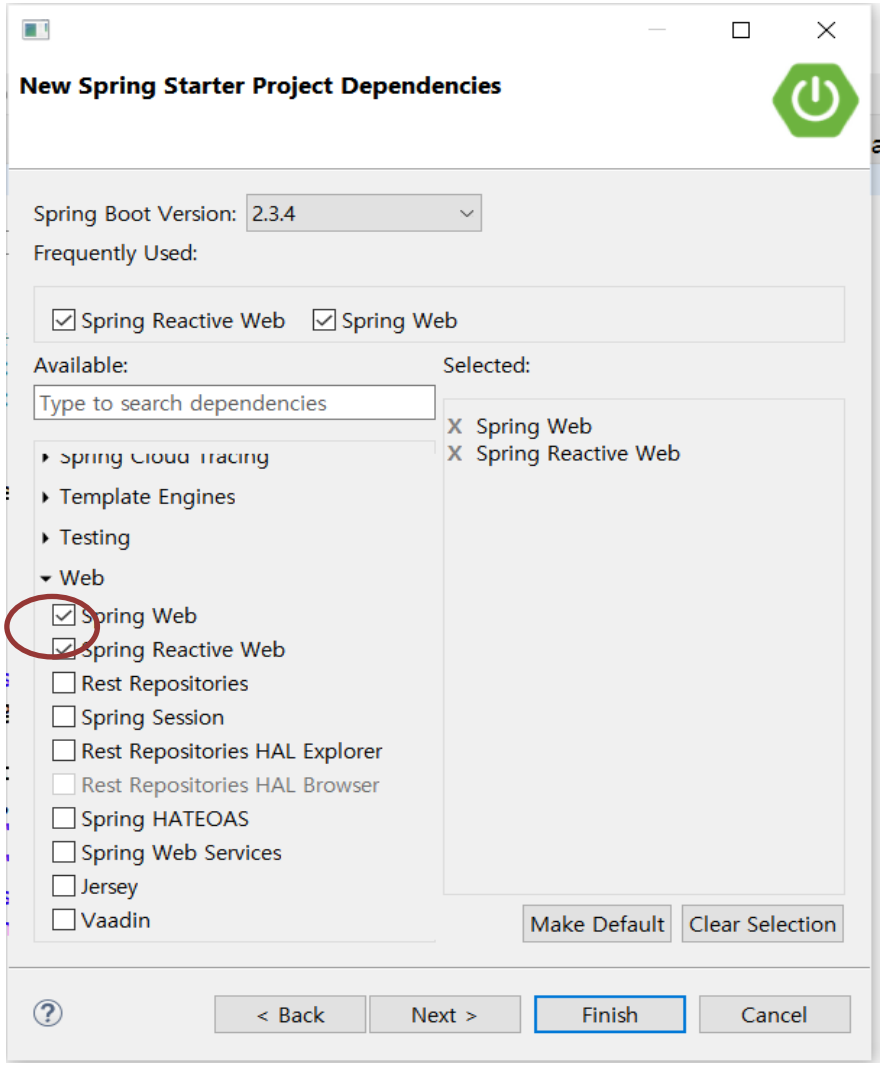


Hello, Web Application

- 프로젝트명 : Hello
- Group에 패키지명을 입력하세요, 최소 2depth입니다

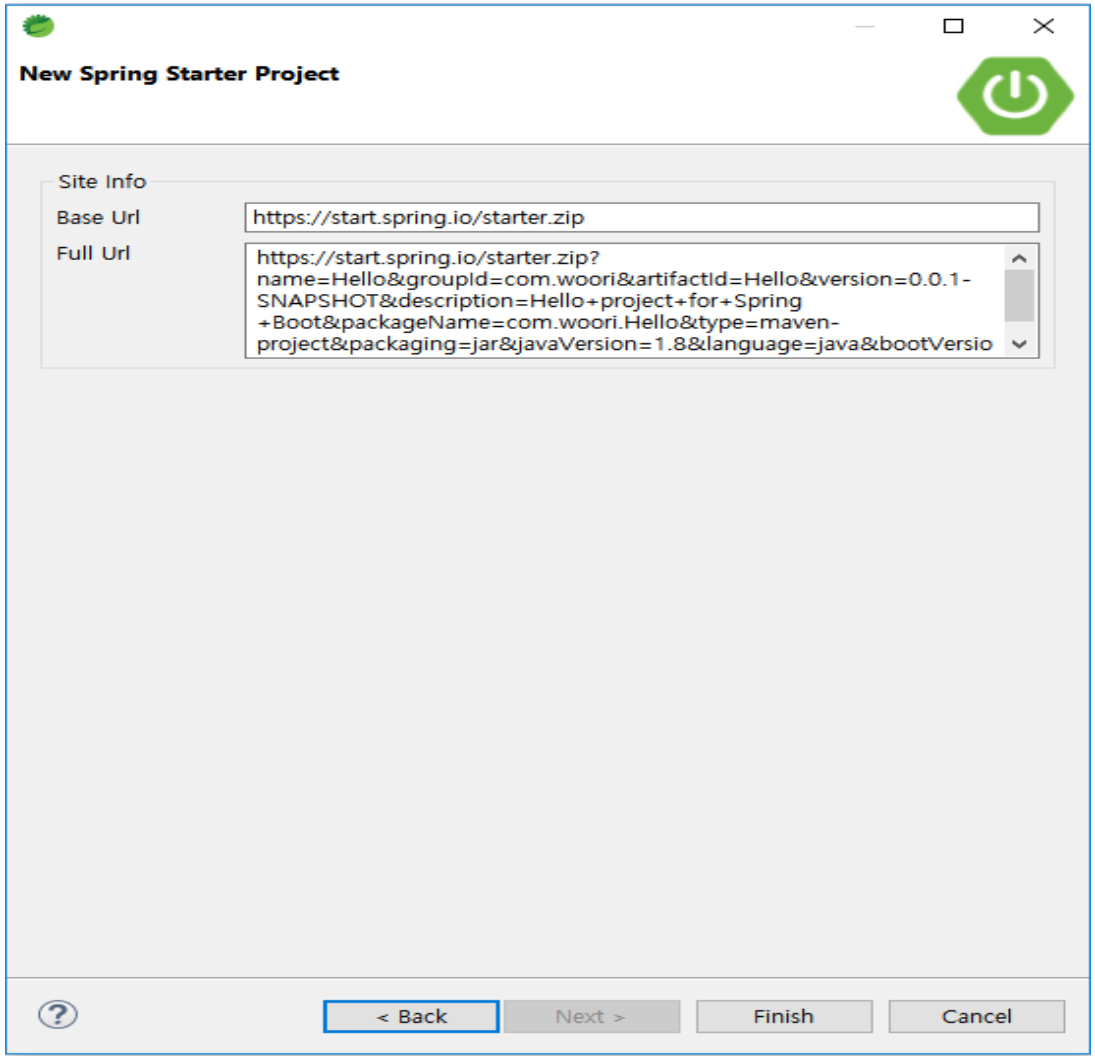


Hello, Web Application

- 

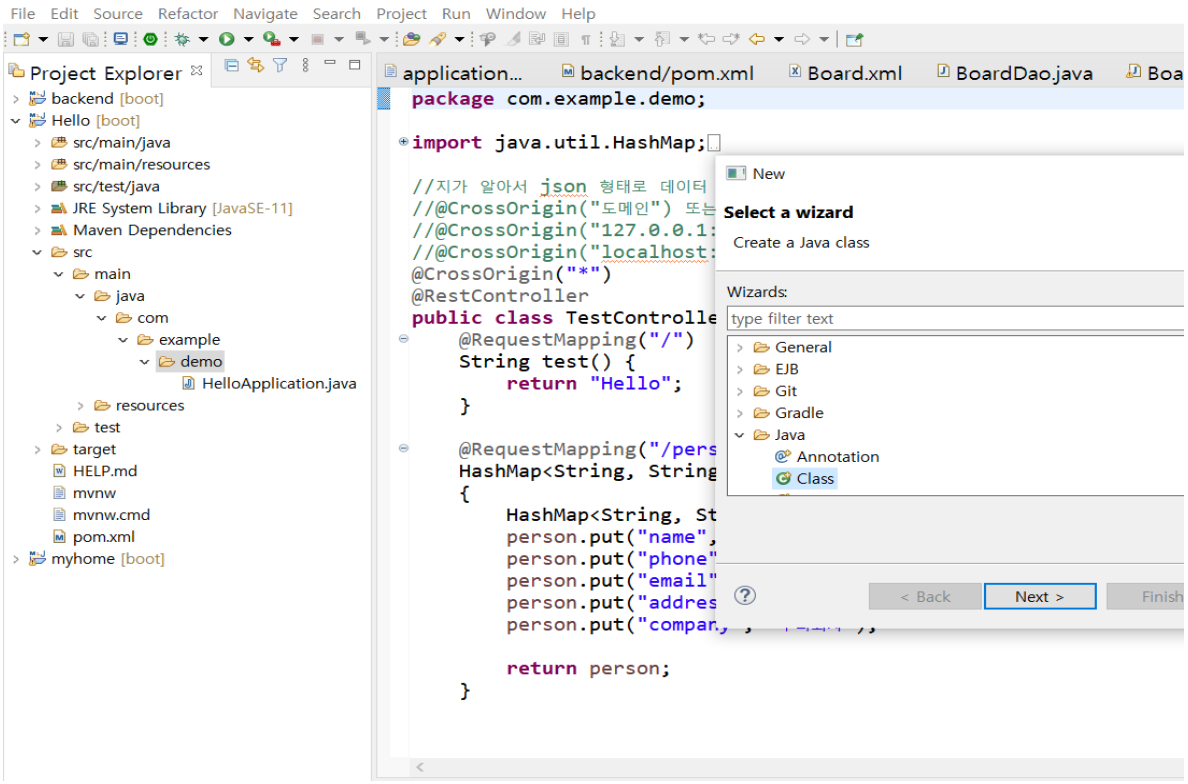
web 선택하기

Hello, Web Application

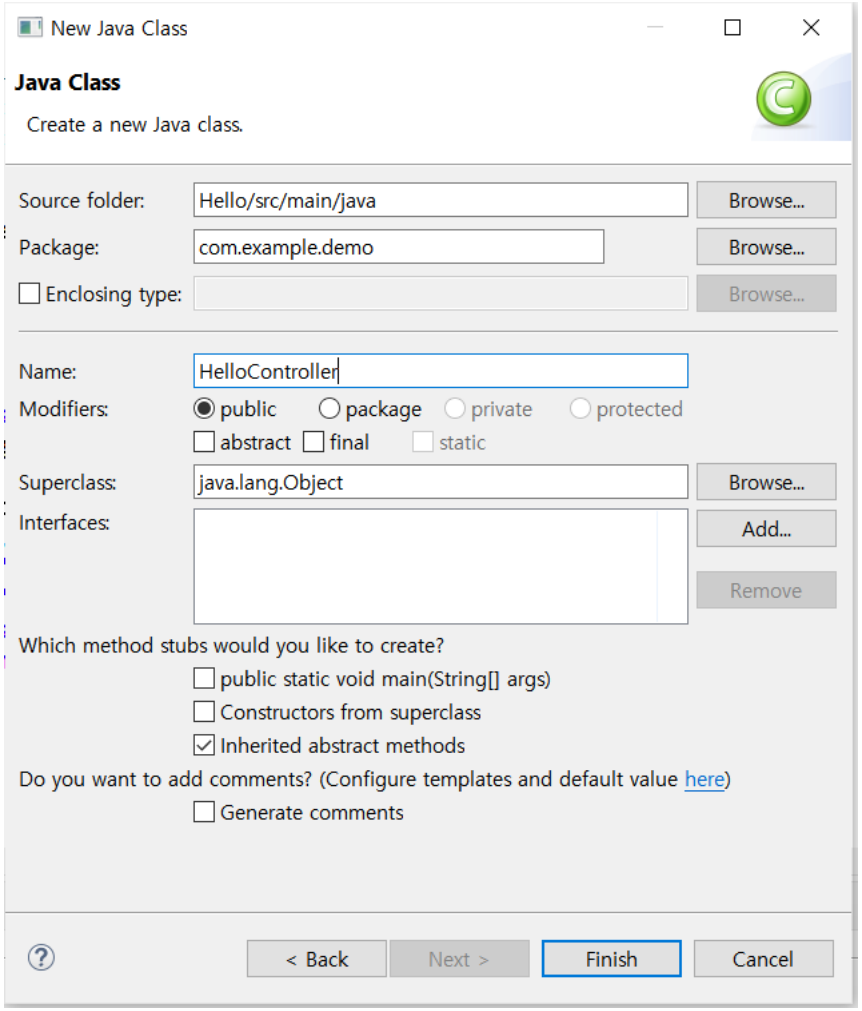
- 

HelloController 추가

- src/main/java/com/example/demo 최소 이 아래에 추가해야 동작한다
- 마우스오른쪽 other - class



HelloController 추가

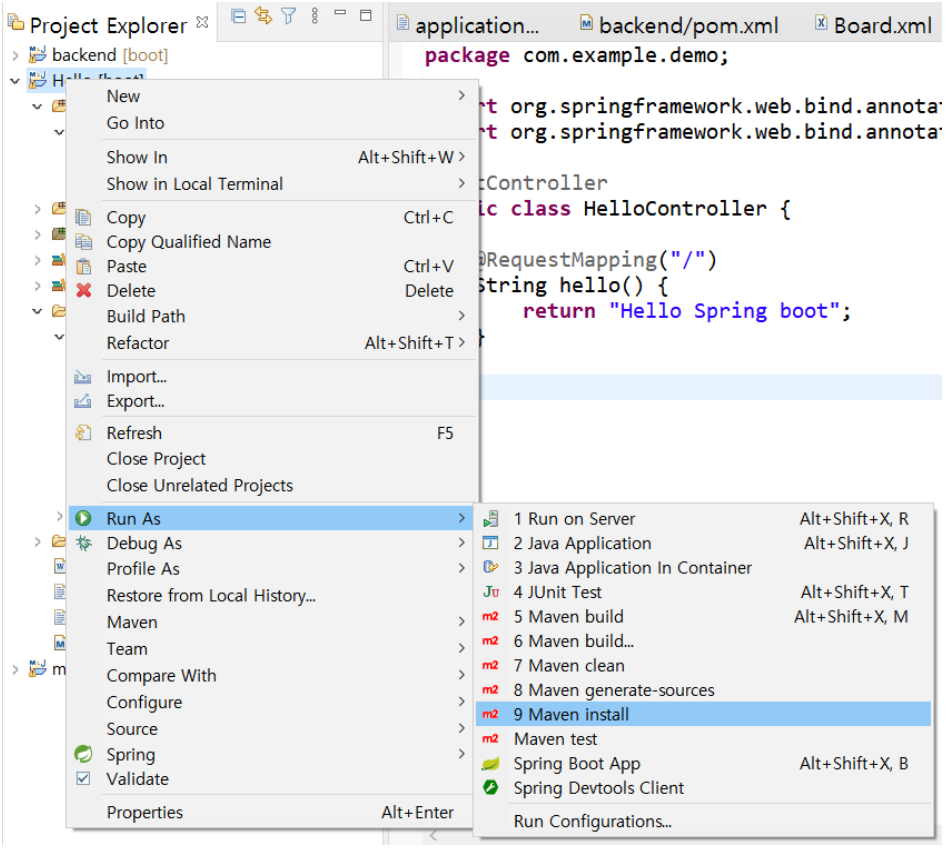


HelloController 작성

```java
package com.example.demo;

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

@RestController
public class HelloController {

    @RequestMapping("/")
    String hello() {
        return "Hello Spring boot";
    }
}
```

maven install

- project - 마우스오른쪽 - run as - maven install



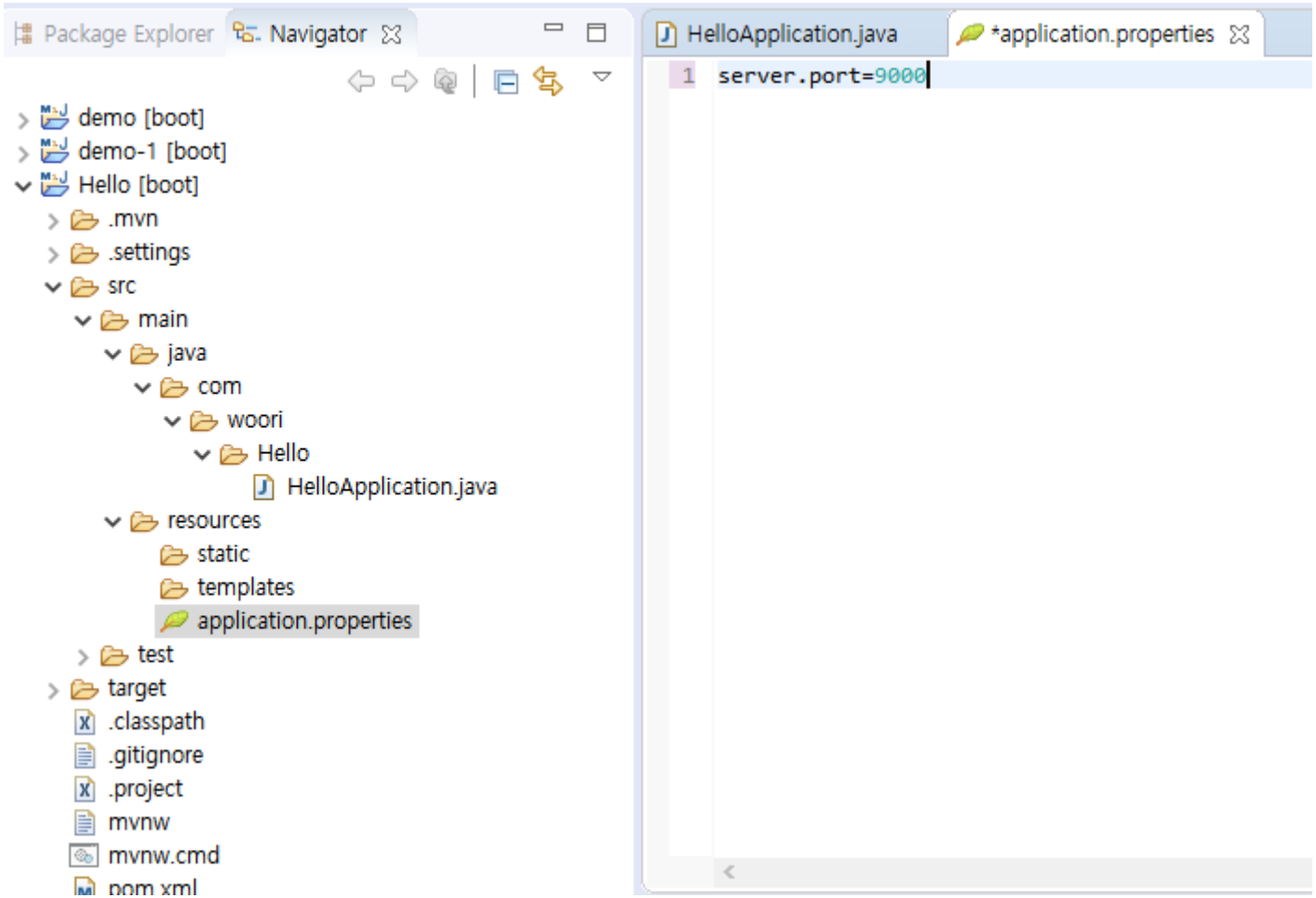
maven install

- maven install 시 필요로하는 파일들을 다운로드 한다

```
[INFO]
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ Hello ---
[INFO] Building jar: C:\Users\user\백현숙\스프링자료\스프링부트\spring_workspace\Hello\target\Hello-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.4.RELEASE:repackage (repackage) @ Hello ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ Hello ---
[INFO] Installing C:\Users\user\백현숙\스프링자료\스프링부트\spring_workspace\Hello\target\Hello-0.0.1-SNAPSHOT.jar to
[INFO] Installing C:\Users\user\백현숙\스프링자료\스프링부트\spring_workspace\Hello\pom.xml to C:\Users\user\.m2\repos:
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  18.464 s
[INFO] Finished at: 2020-10-12T22:23:41+09:00
[INFO] ------------------------------------------------------------------------
[WARNING] The requested profile "pom.xml" could not be activated because it does not exist.
```

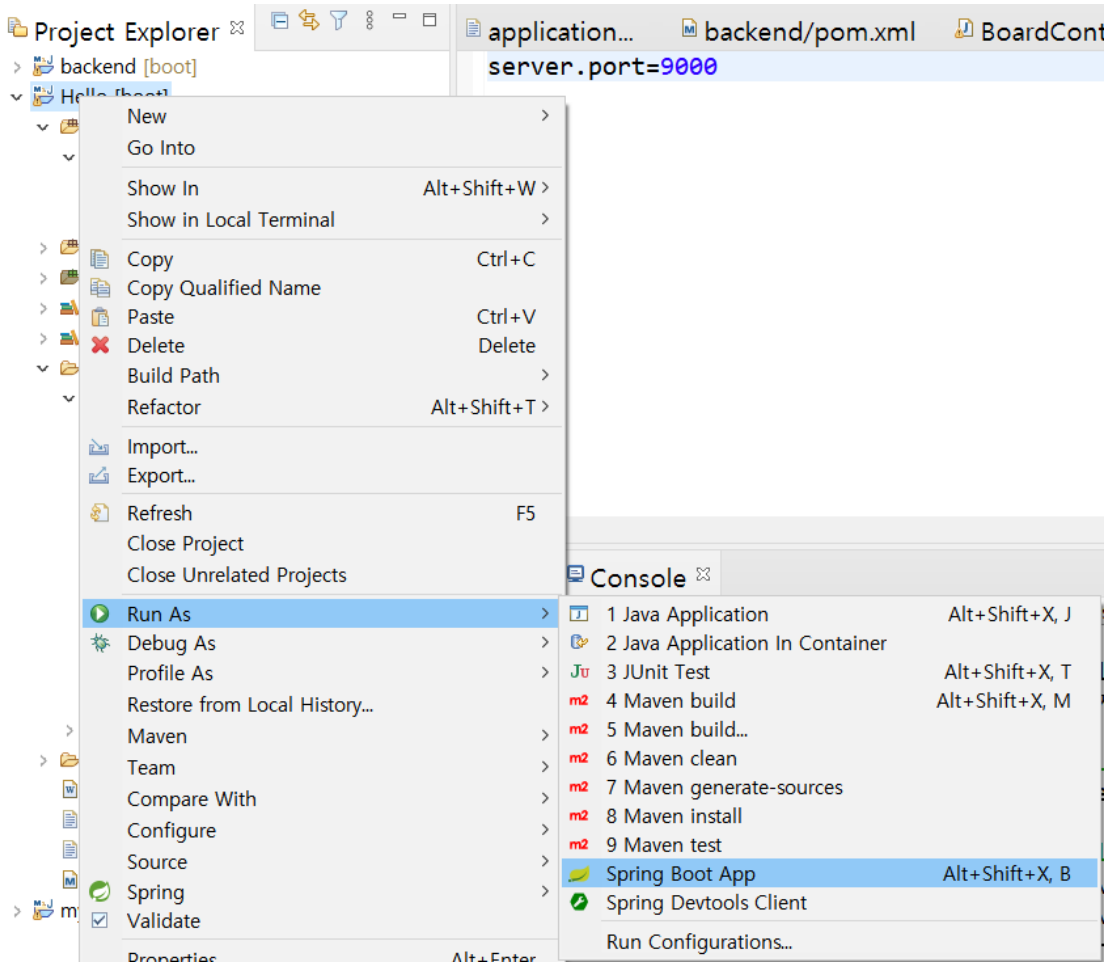
Hello, Web Application

- application.properties 에 포트번호를 지정합니다



서버 실행

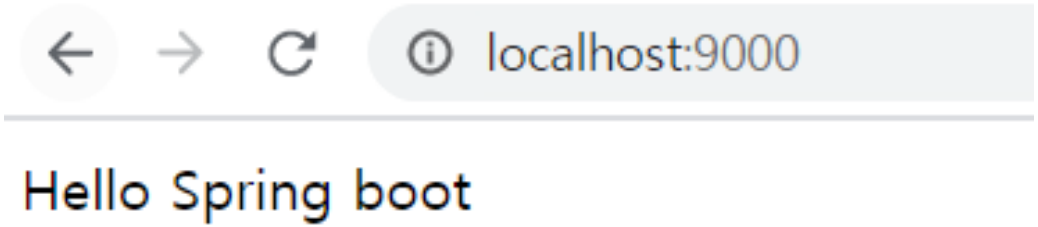
- 프로젝트명 - 마우스 오른쪽 - run as - spring boot app



Hello, Web Application

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.0.RELEASE)

2018-11-06 16:21:11.625  INFO 16200 --- [           main] com.woori.Hello.HelloApplication         : Starting HelloApplication on LAPTOP-8IK90LES with PID 16200 (C:\SpringBoot'
2018-11-06 16:21:11.629  INFO 16200 --- [           main] com.woori.Hello.HelloApplication         : No active profile set, falling back to default profiles: default
2018-11-06 16:21:12.611  INFO 16200 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 9000 (http)
2018-11-06 16:21:12.628  INFO 16200 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2018-11-06 16:21:12.628  INFO 16200 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/9.0.12
2018-11-06 16:21:12.634  INFO 16200 --- [           main] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance
2018-11-06 16:21:12.763  INFO 16200 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2018-11-06 16:21:12.763  INFO 16200 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 1083 ms
2018-11-06 16:21:12.806  INFO 16200 --- [           main] o.s.b.w.servlet.ServletRegistrationBean  : Servlet dispatcherServlet mapped to [/]
2018-11-06 16:21:12.810  INFO 16200 --- [           main] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'characterEncodingFilter' to: [/*]
2018-11-06 16:21:12.810  INFO 16200 --- [           main] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]
2018-11-06 16:21:12.810  INFO 16200 --- [           main] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'formContentFilter' to: [/*]
2018-11-06 16:21:12.810  INFO 16200 --- [           main] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'requestContextFilter' to: [/*]
2018-11-06 16:21:13.006  INFO 16200 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2018-11-06 16:21:13.224  INFO 16200 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 9000 (http) with context path ''
2018-11-06 16:21:13.227  INFO 16200 --- [           main] com.woori.Hello.HelloApplication         : Started HelloApplication in 1.944 seconds (JVM running for 2.921)
```

Hello, Web Application

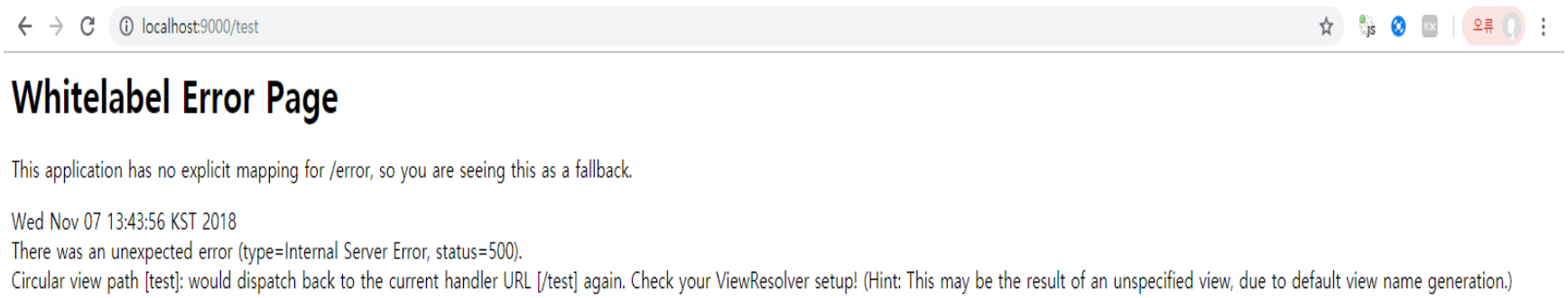


Hello, Web Application

- @RestController : 별도의 html 페이지가 필요 없다.
- @Controller : 별도의 html 페이지가 필요하다

```java
package com.woori.Hello;

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

@Controller
public class TestController {

    @RequestMapping("/test")
    public String test()
    {
        return "test";
    }
}
```

Hello, Web Application



Hello, Web Application

- html 뷰 지원하라면 반드시 아래 코드를 pom.xml에 넣어야 한다

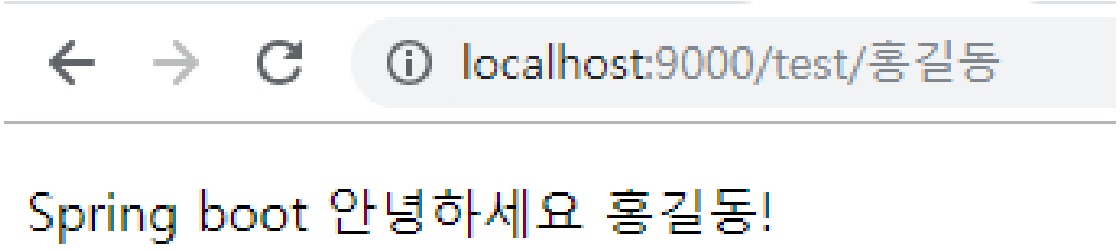
```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-mustache</artifactId>
</dependency>
```

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-mustache</artifactId>
</dependency>
```

Hello, Web Application

- src/main/resources/application.properties 파일에
- 아래처럼 기술해야 한다
- spring.mustache.suffix: .html

실행 결과



참고

- 자동 import 구문 안보일때 pom.xm 파일에 다음 추가

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```


```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

jsp 사용하려면

- eclipse –help – market place
- Eclipse EE Developer Tool 설치하기

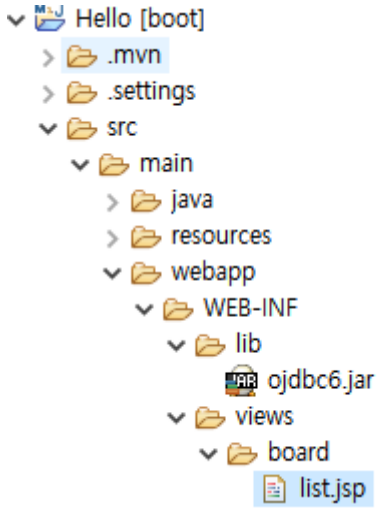
pom.xml에 추가

```
<!--  jsp 지원하기  -->
<dependency>
 <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>

<!--  jstl 지원하기  -->
<dependency>
    <groupId>javax.servlet.jsp.jstl</groupId>
    <artifactId>javax.servlet.jsp.jstl-api</artifactId>
    <version>1.2.1</version>
</dependency>
```

application.properties

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

jsp 추가


Hello [boot]
.mvn
.settings
src
main
java
resources
webapp
WEB-INF
lib
ojdbc6.jar
views
board
list.jsp

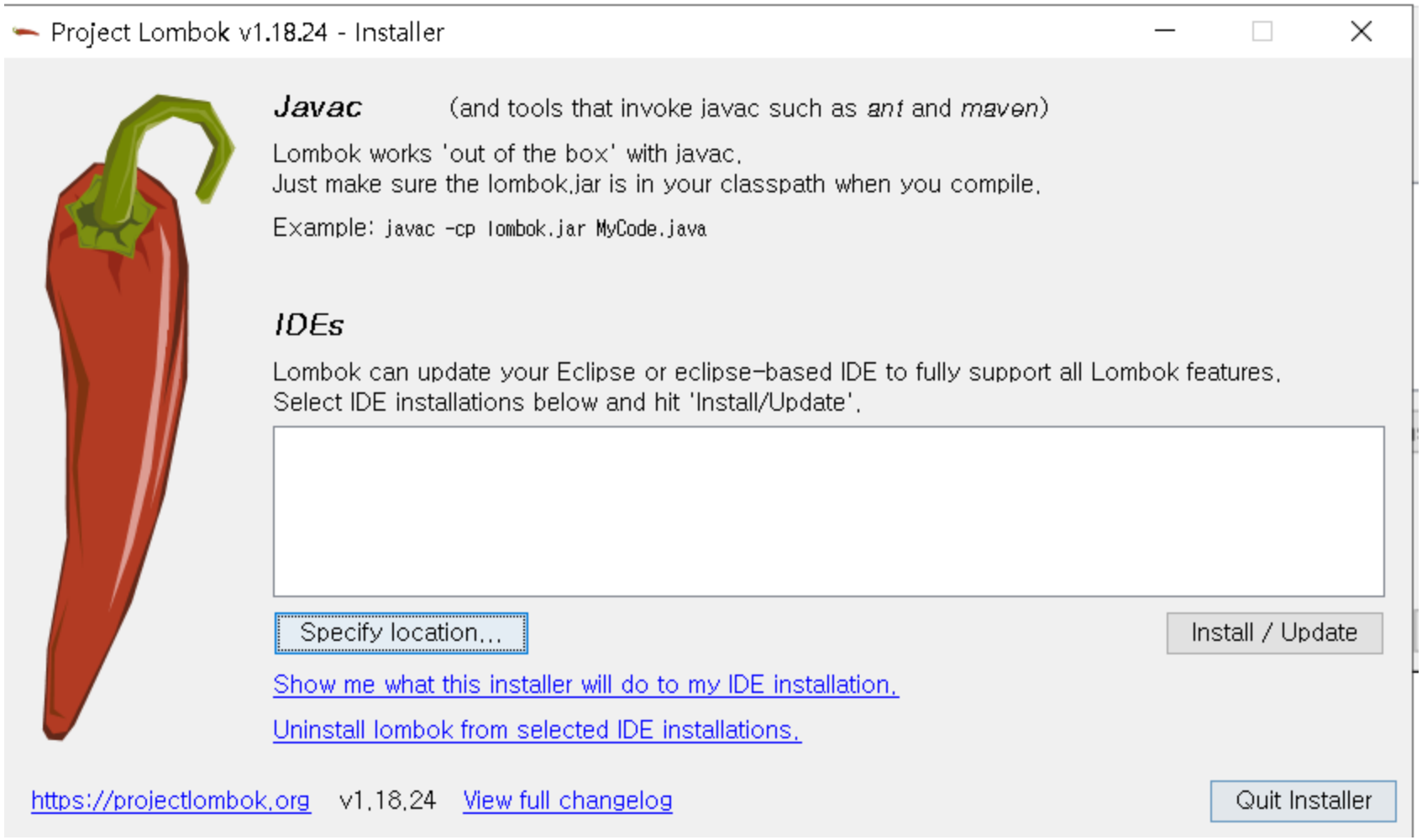
lombok설치하기

- **롬복 다운로드**
- **https://projectlombok.org/download**

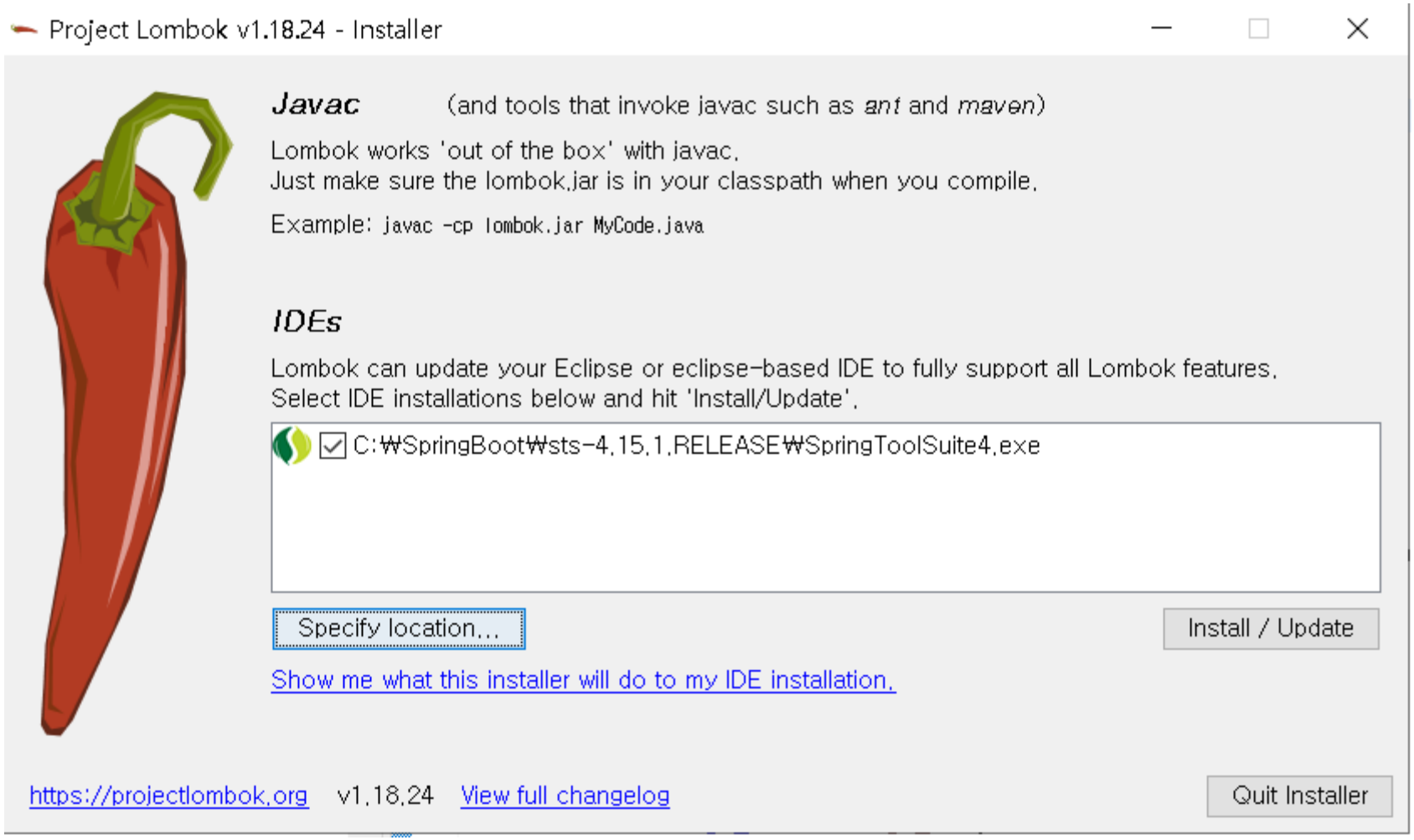
- 이클립스(Eclipse)가 설치된 경로에 **lombok.jar** 파일을 추가하고, jar를 실행해 주세요.

- cmd 창을 관리자 권한으로 켠 후
-
- java -jar lombok.jar 로 실행하기

lombok설치하기


Project Lombok v1.18.24 - Installer
Javac (and tools that invoke javac such as ant and maven)
Lombok works 'out of the box' with javac.
Just make sure the lombok.jar is in your classpath when you compile.
Example: javac -cp lombok.jar MyCode.java
IDEs
Lombok can update your Eclipse or eclipse-based IDE to fully support all Lombok features.
Select IDE installations below and hit 'Install/Update'.
Specify location...
Install / Update
Show me what this installer will do to my IDE installation.
Uninstall lombok from selected IDE installations.
https://projectlombok.org
v1.18.24
View full changelog
Quit Installer

lombok설치하기


Project Lombok v1.18.24 - Installer
Javac (and tools that invoke javac such as ant and maven)
Lombok works 'out of the box' with javac.
Just make sure the lombok.jar is in your classpath when you compile.
Example: javac -cp lombok.jar MyCode.java
IDEs
Lombok can update your Eclipse or eclipse-based IDE to fully support all Lombok features.
Select IDE installations below and hit 'Install/Update'.
C:₩SpringBoot₩sts-4.15.1.RELEASE₩SpringToolSuite4.exe
Specify location...
Install / Update
Show me what this installer will do to my IDE installation.
https://projectlombok.org
v1.18.24
View full changelog
Quit Installer

lombok 설치하기

- 이클립스를 다시 켠다
- 기존 프로젝트일 경우에 파일들을 다시 컴파일 한다.

backend 프로젝트 생성

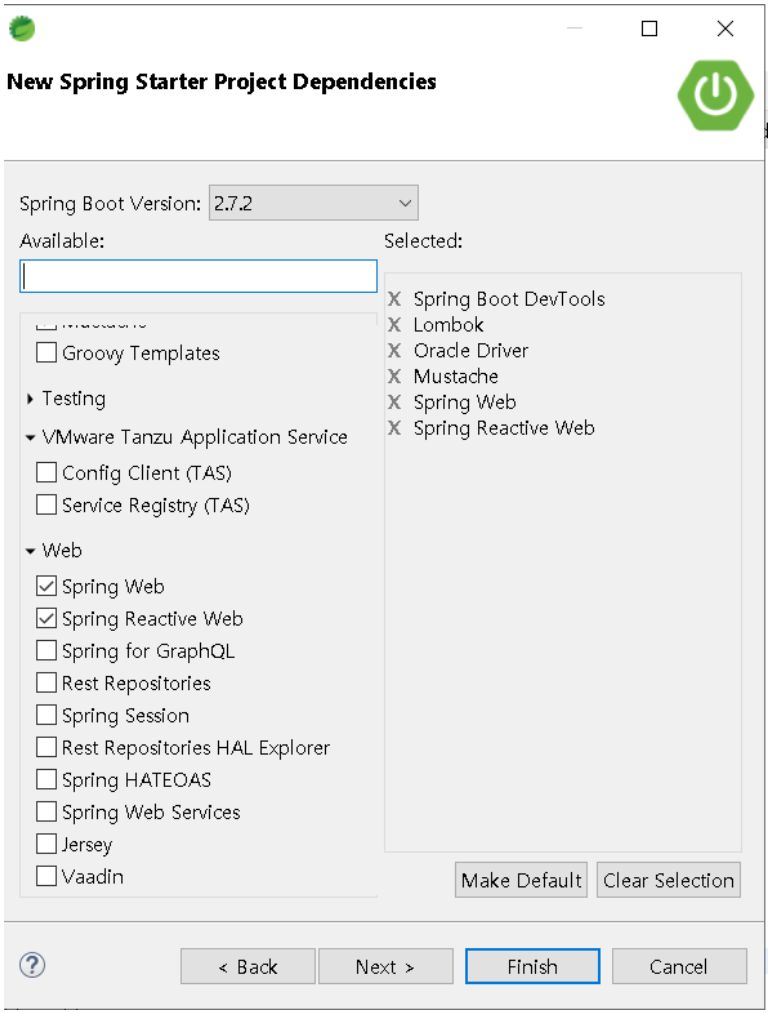
- File - new - Spring Starter Project

- 프로젝트명 : myhome
- 프로젝트 : Gradule (Maven은 배포시 문제 발생)
- java : 11 (17은 배포시 문제 발생)

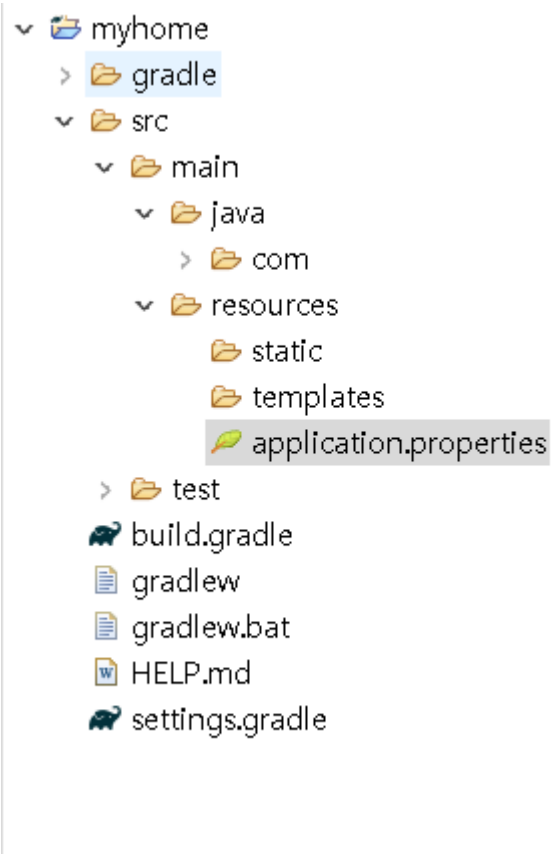
필요라이브러리

- Devloper tool
  - Lombok
  - Sprint Boot DevTools
- SQL
  - oracle driver
- Template engine
  - mustache
- Web
  - Spring Web
  - Spring Rective Web

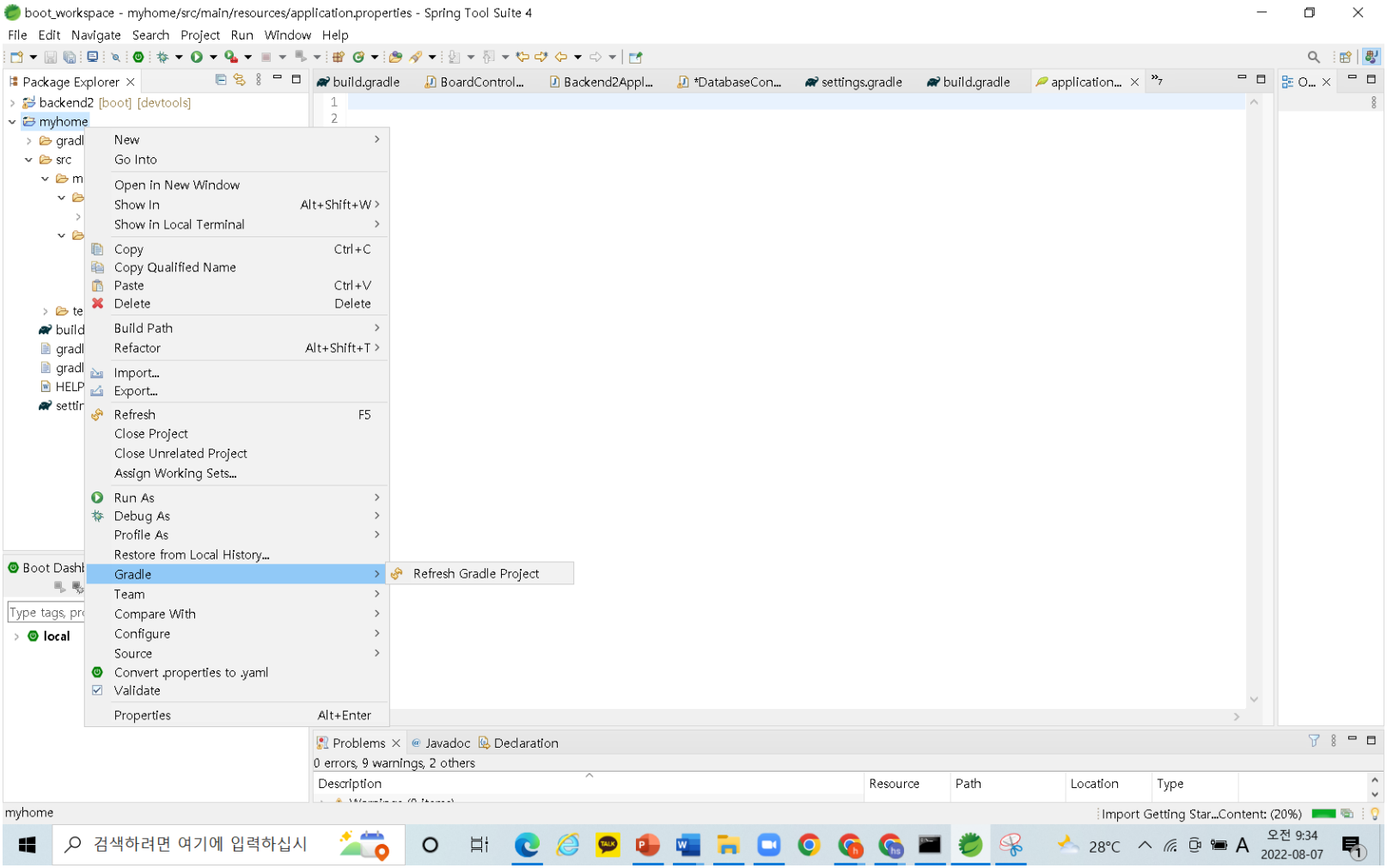
backend 프로젝트 생성



backend 프로젝트 생성



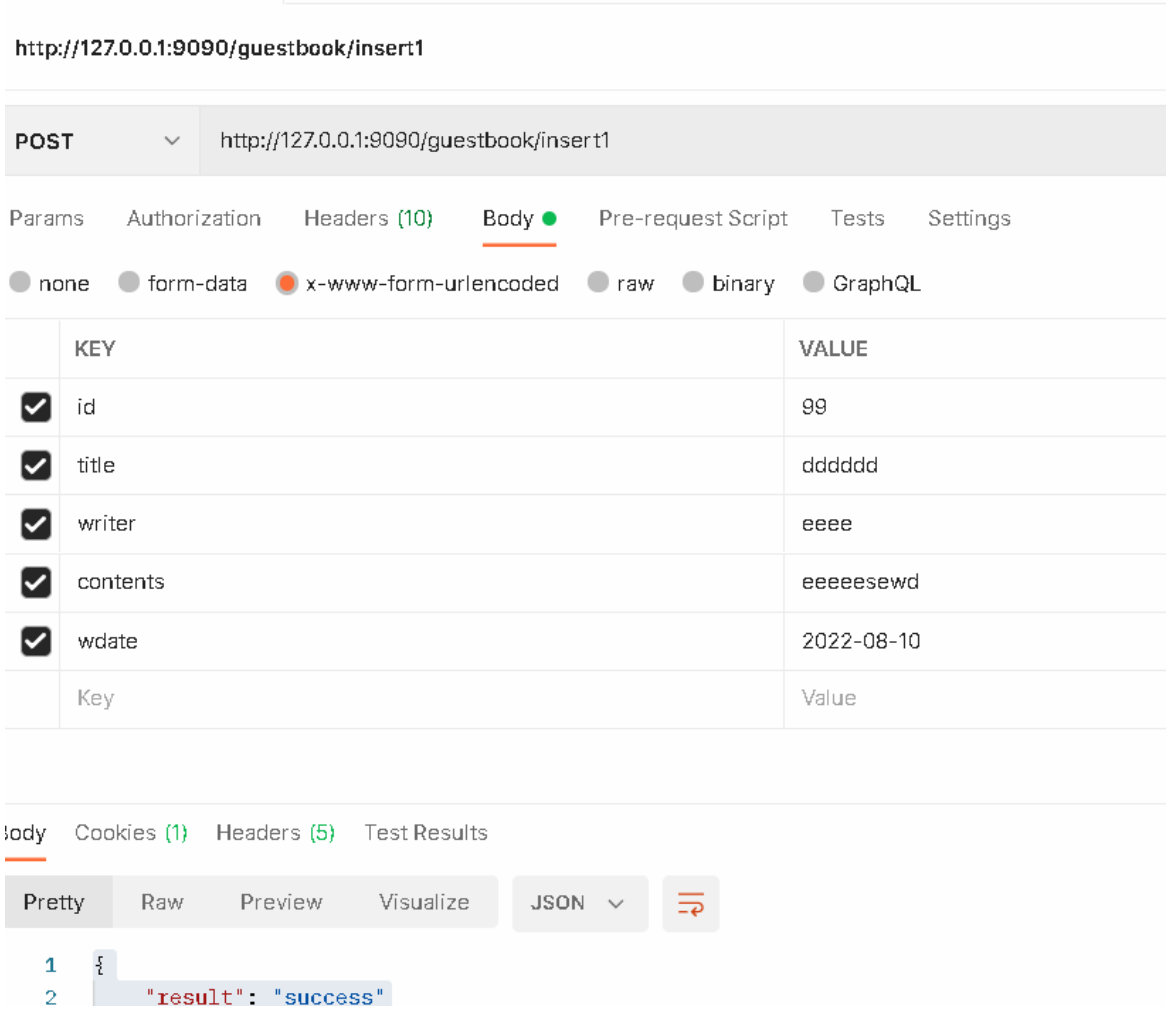
Gradule - Refresh Gradule Project



POST 방식(X-www-form-urlencoded)

```
//GET방식은 데이터 전송시 header만 보낸다. 비교적 간단한 정보만 보낸다.
//POST방식은 데이터를 전송할때 우선 header를 보내고 body를 보낸댜
//form-data : 파일 업로드 테스트할때 파일을 업로드 하면, form태그에
//            enctype="multipart/form-data"

//x-www-form-urlencoded : 일반적인 POST 방식
//raw : JSON방식으로 전송할때

//x-www-form-urlencoded
@RequestMapping("/guestbook/insert1")
public HashMap<String, String>insert1(GuestbookDto dto)
{
    HashMap<String, String> map = new HashMap<String, String>();
    System.out.println(dto.getTitle());
    service.insert(dto);
    map.put("result", "success");
    return map;
}
```

x-www-form-urlencoded

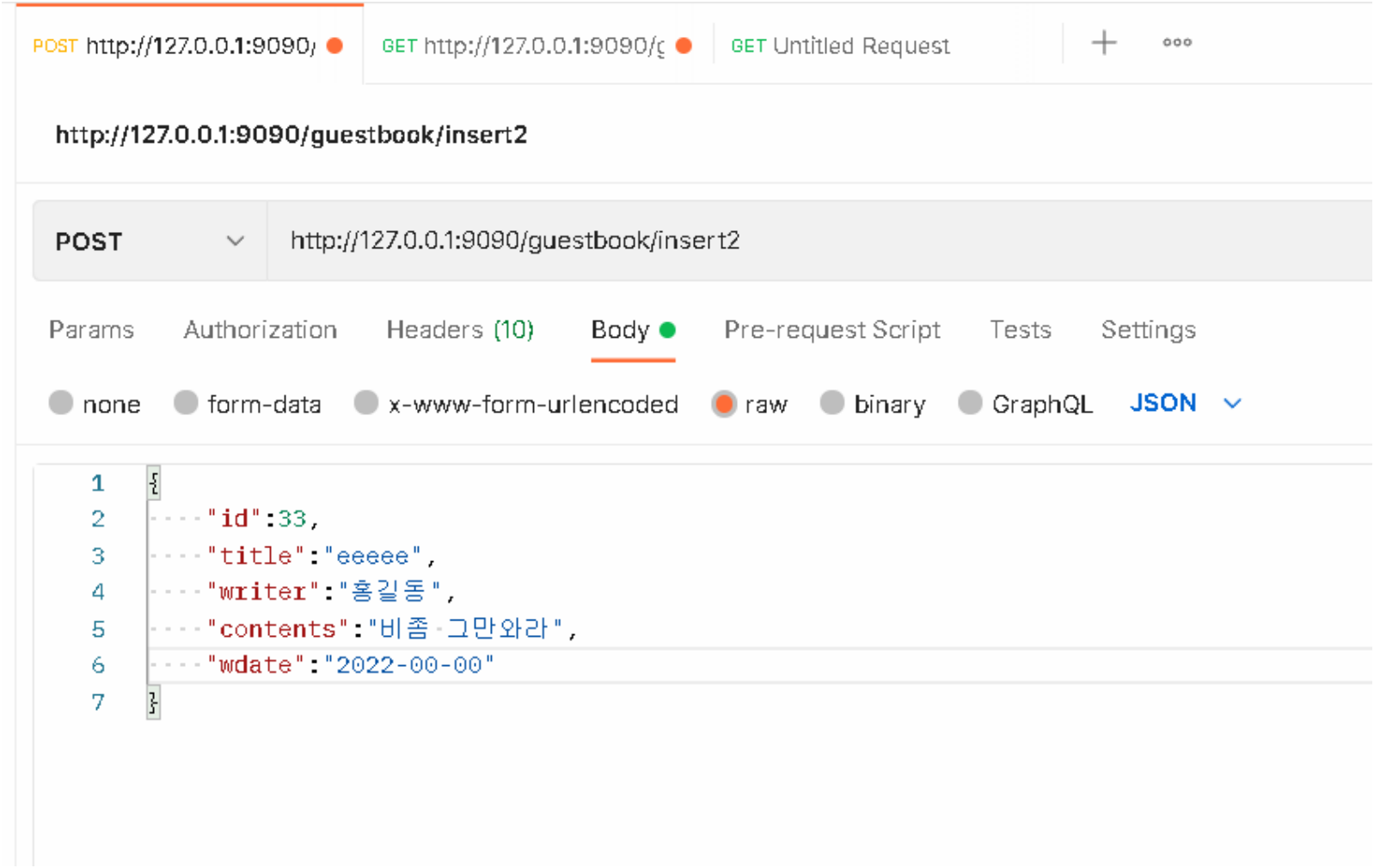


POST2(RAW-JSON)

- //RAW - JSON

```
@RequestMapping("/guestbook/insert2")
public HashMap<String, String>insert2(@RequestBody GuestbookDto
dto)
{
    HashMap<String, String> map = new HashMap<String, String>();
    System.out.println(dto.getTitle());
    service.insert(dto);
    map.put("result", "success");
    return map;
}
```

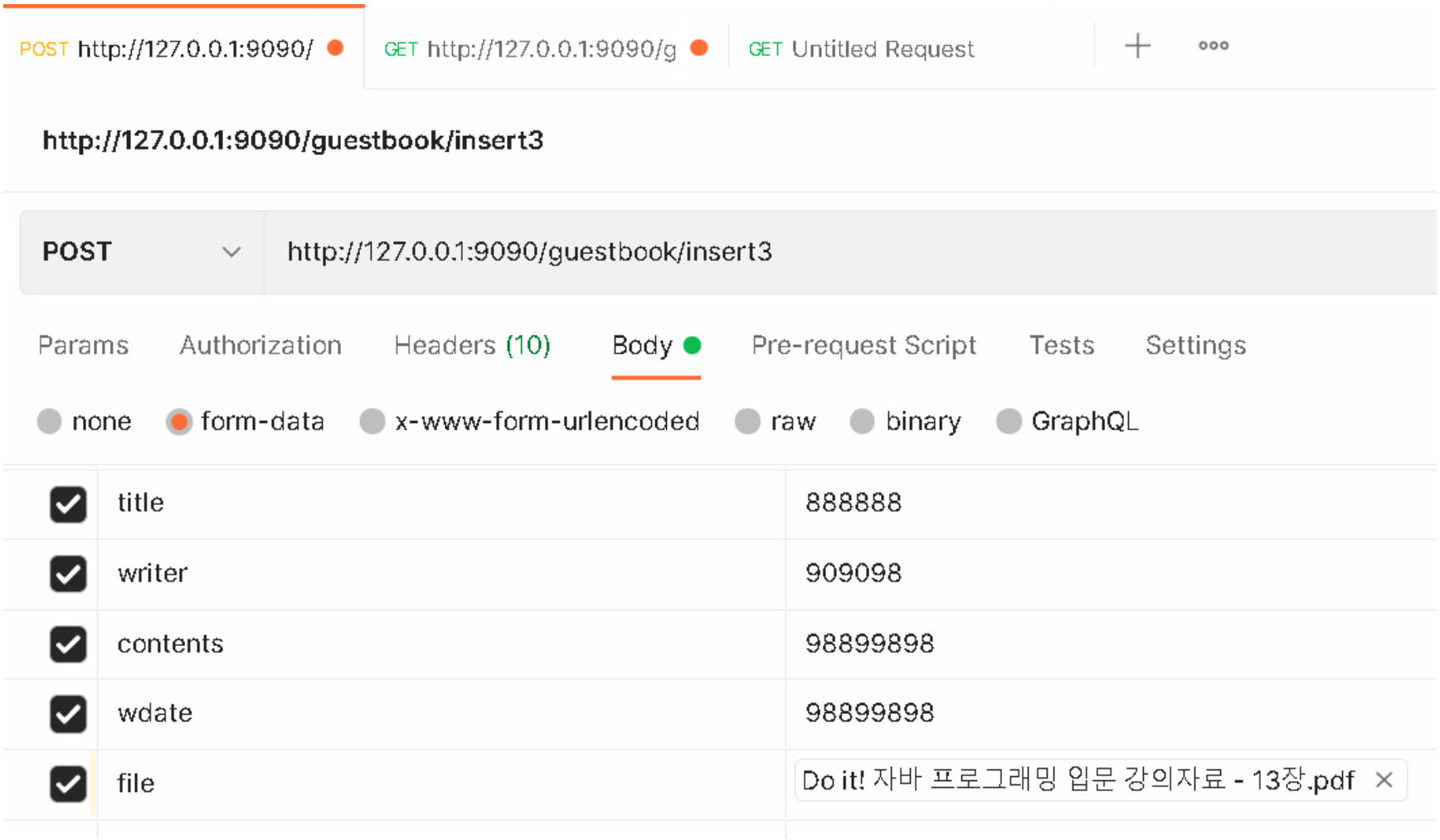
raw 방식


POST http://127.0.0.1:9090/
GET http://127.0.0.1:9090/
GET Untitled Request
http://127.0.0.1:9090/guestbook/insert2
POST
http://127.0.0.1:9090/guestbook/insert2
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"id":33,
"title":"eeeee",
"writer":"홍길동",
"contents":"비좀 그만와라",
"wdate":"2022-00-00"
}

POST3(form-data)

```
//form-data
@RequestMapping("/guestbook/insert3")
public HashMap<String, String>insert3(MultipartFile file,
GuestbookDto dto)
{
    HashMap<String, String> map = new HashMap<String, String>();
    System.out.println(dto.getTitle());
    service.insert(dto);
    map.put("result", "success");
    return map;
}
```

form-data


POST http://127.0.0.1:9090/
GET http://127.0.0.1:9090/g
GET Untitled Request
http://127.0.0.1:9090/guestbook/insert3
POST
http://127.0.0.1:9090/guestbook/insert3
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
title
888888
writer
909098
contents
98899898
wdate
98899898
file
Do it! 자바 프로그래밍 입문 강의자료 - 13장.pdf

myapp-router

- nodejs를 설치한다
- 공유폴더에서 프로젝트를 다운 받는다
- 압축을 푼다
- cd 폴더로 이동한다
- npm install 을 실행하여 필요한 라이브러리를 설치한다
- npm start 명령어를 입력하면 서버가 작동한다
- localhost:3000
- 게시판 메뉴가 서버와 통신한다

스프링 부트

```
@CrossOrigin("*") //도메인
@RestController
public class GuestbookController {
    @Autowired
    GuestbookService service;

    @RequestMapping("/guestbook/list")
    HashMap<String, Object>getList(){

        List<GuestbookDto>  list = service.getList();
        HashMap<String, Object> map = new HashMap<String, Object>();
        map.put("totalCnt", list.size());
        map.put("list", list);

        return map;
    }
```

DB설정(application.properties)

```
#oracle
spring.datasource.url=jdbc:oracle:thin:@127.0.0.1:1521:XE
spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
spring.datasource.username=user01
spring.datasource.password=1234

#file upload
spring.servlet.multipart.maxFileSize=20MB
spring.servlet.multipart.maxRequestSize=20MB

spring.http.multipart.max-file-size=20MB
spring.http.multipart.max-request-size=20MB

fileUploadPath=fileupload
domain=http://127.0.0.1:9090
```

기본 패키지 아래에 config 폴더를 만든다

- **DatabaseConfig.java 파일추가**

```
@Configuration
@PropertySource("classpath:/application.properties")
public class DatabaseConfig implements WebMvcConfigurer{

    @Value("${fileUploadPath}")
    String fileUploadPath;

     @Bean
     public SqlSessionFactory sqlSessionFactory(DataSource dataSource) throws Exception {
       final SqlSessionFactoryBean sessionFactory = new SqlSessionFactoryBean();
       sessionFactory.setDataSource(dataSource);
       PathMatchingResourcePatternResolver resolver = new PathMatchingResourcePatternResolver();
       Resource configLocation =
           (Resource)new PathMatchingResourcePatternResolver().getResource("classpath:mybatis-config.xml");

       sessionFactory.setConfigLocation(configLocation);
       //sessionFactory.setMapperLocations(resolver.getResources("classpath:mappers/*.xml"));
       return sessionFactory.getObject();
    }

    @Bean
    public SqlSessionTemplate sqlSession(SqlSessionFactory sqlSessionFactory) {
        return new SqlSessionTemplate(sqlSessionFactory);
    }
```

mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
    <typeAliases>
        <!-- package name="com.example.demo"/-->

        <typeAlias alias="BoardDto"
type="com.example.demo.board.domain.BoardDto"/>
    </typeAliases>

    <mappers>
        <mapper resource="mappers/Board.xml"/>
    </mappers>
</configuration>
```

Guestbook.xml

- filename : resources/mappers/Guestbook.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
                PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
                "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="Board">

                <!-- 쿼리를 xml로 작성하고 id:식별값
                                parameterType:우리가 전달할값
                    resultType : 받아오는값  -->

                <!-- if test="key=='1'"  - 이렇게 쓰면 안된다 인식안된다 -->

                <select id="Board_getList"  parameterType="BoardDto"  resultType="BoardDto">

                                select * from board

                </select>
</mapper>
```

GuestbookDao

```
@Autowired
SqlSessionTemplate sm;

@Override
public List<BoardDto> getList(BoardDto dto) {

        return sm.selectList("Board_getList", dto);
}

@Override
public BoardDto getView(long id) {
        return sm.selectOne("Board_getView",  id);
}

@Override
public void insert(BoardDto dto) {

        sm.insert("Board_insert", dto);
}
```