

삼성 청년 SW 아카데미

Spring Framework

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

목차

1. SpringBoot
2. 방명록 생성
3. Swagger

SpringBoot

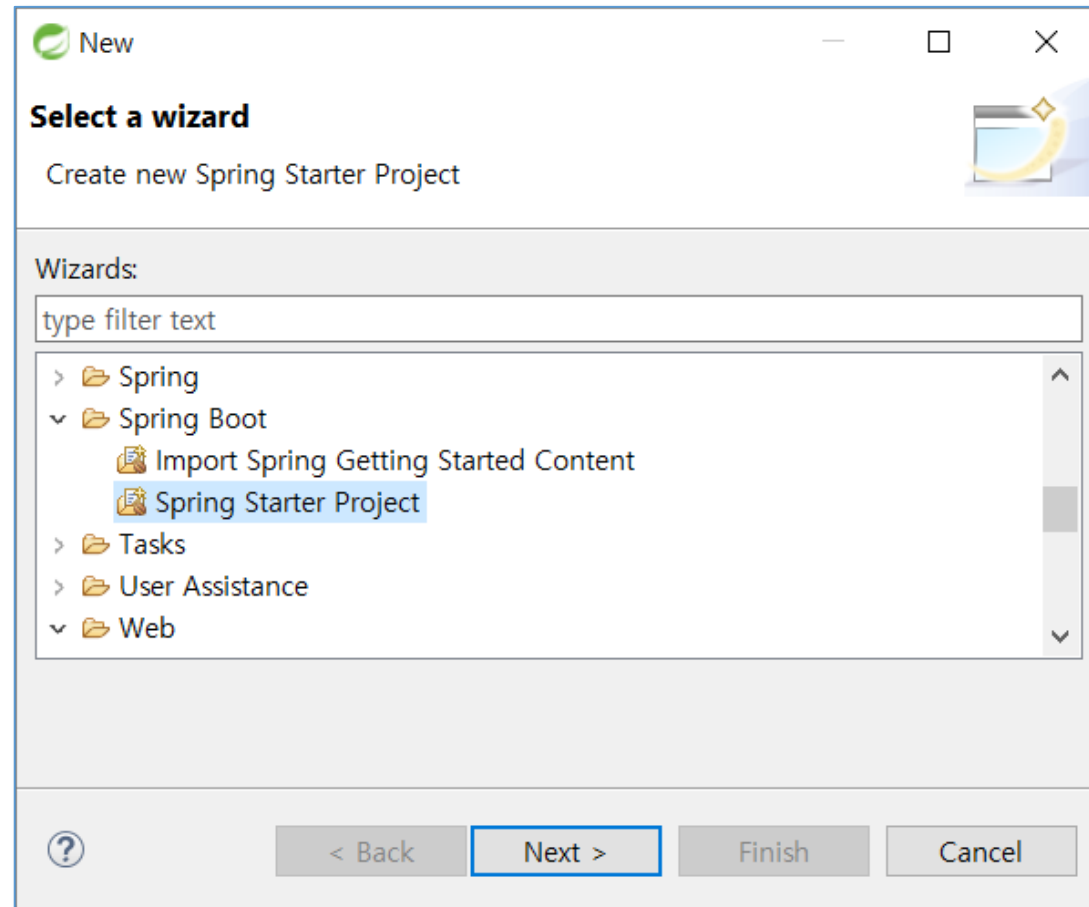
삼성 청년 SW 아카데미

✓ SpringBoot의 특징.

- Spring의 경우 Application을 개발하려면 사전에 많은 작업(library 추가, dependency 설정, SpringFramework가 처리해야 하는 여러 가지 구성 및 설정파일 등)을 해야 했다.
- SpringBoot의 장점.
 - project에 따라 자주 사용되는 library들이 미리 조합되어 있다.
 - 복잡한 설정을 자동으로 처리.
 - 내장 서버를 포함해서 tomcat과 같은 WAS를 추가로 설치하지 않아도 개발 가능.
 - WAS에 배포하지 않고도 실행할 수 있는 JAR파일로 Web Application을 개발 할 수 있다.

✓ SpringBoot Project 생성.

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



✓ SpringBoot Project 생성.

- 그림과 같이 설정 후 Next.

New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

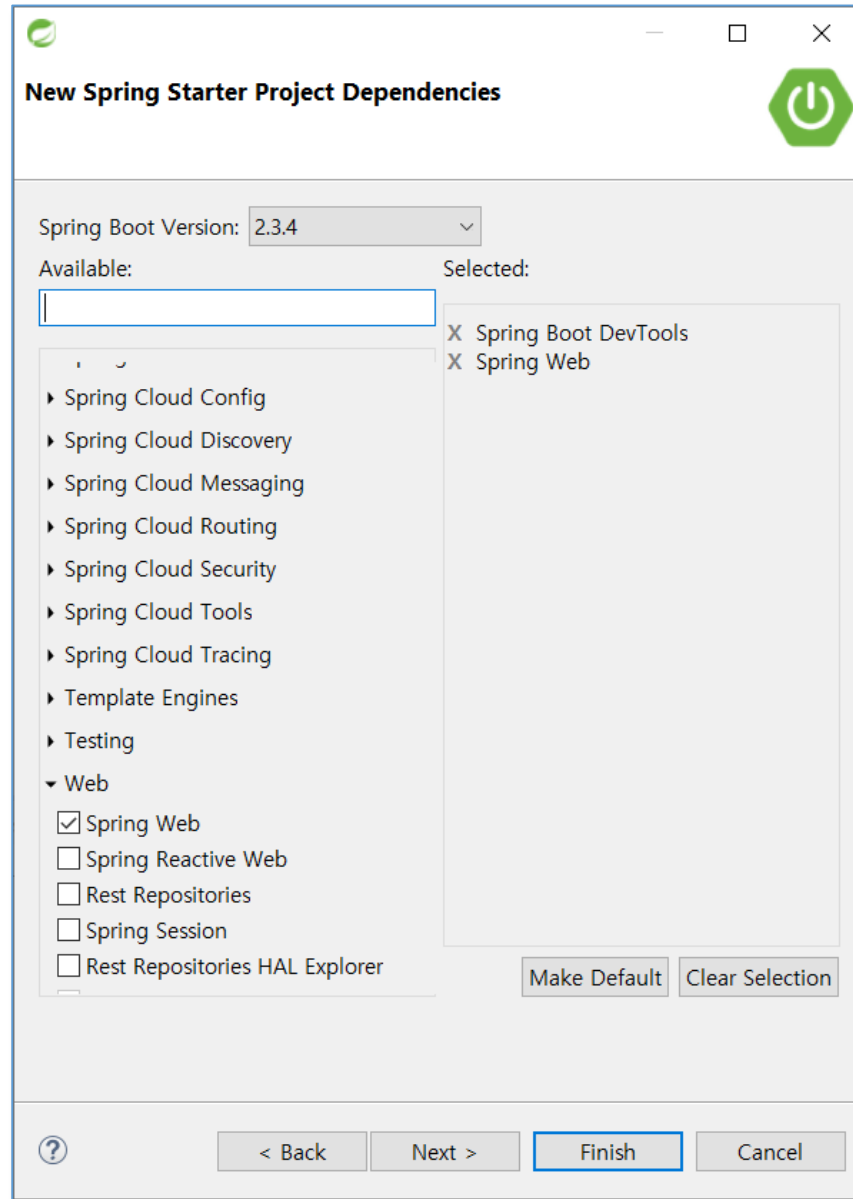
Working sets

☐ Add project to working sets

Working sets:

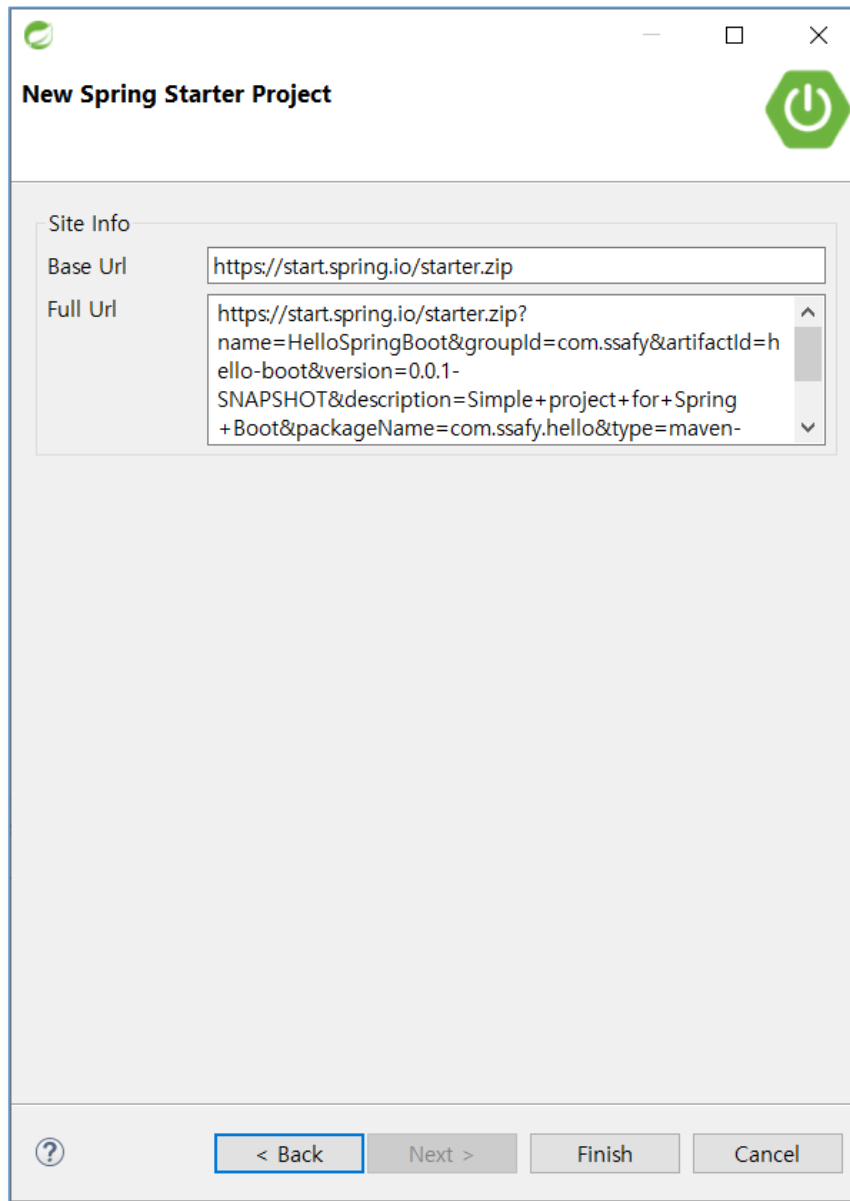
✓ SpringBoot Project 생성.

- SpringBoot의 버전 및 Dependency 설정.



✓ SpringBoot Project 생성.

- 확인 후 Finish.



The image shows a 'New Spring Starter Project' dialog box. It has a title bar with a green icon, a minus button, a maximize button, and a close button. Below the title bar is a green power button icon. The main area is divided into two sections: 'Site Info' and 'Full Url'. The 'Site Info' section has a 'Base Url' field with the value 'https://start.spring.io/starter.zip'. The 'Full Url' section has a text area with the value 'https://start.spring.io/starter.zip?name=HelloSpringBoot&groupId=com.ssafy&artifactId=hello-boot&version=0.0.1-SNAPSHOT&description=Simple+project+for+Spring+Boot&packageName=com.ssafy.hello&type=maven-'. At the bottom, there are four buttons: a help button (question mark), a '< Back' button, a 'Next >' button, and a 'Finish' button. The 'Finish' button is highlighted.

New Spring Starter Project

Site Info

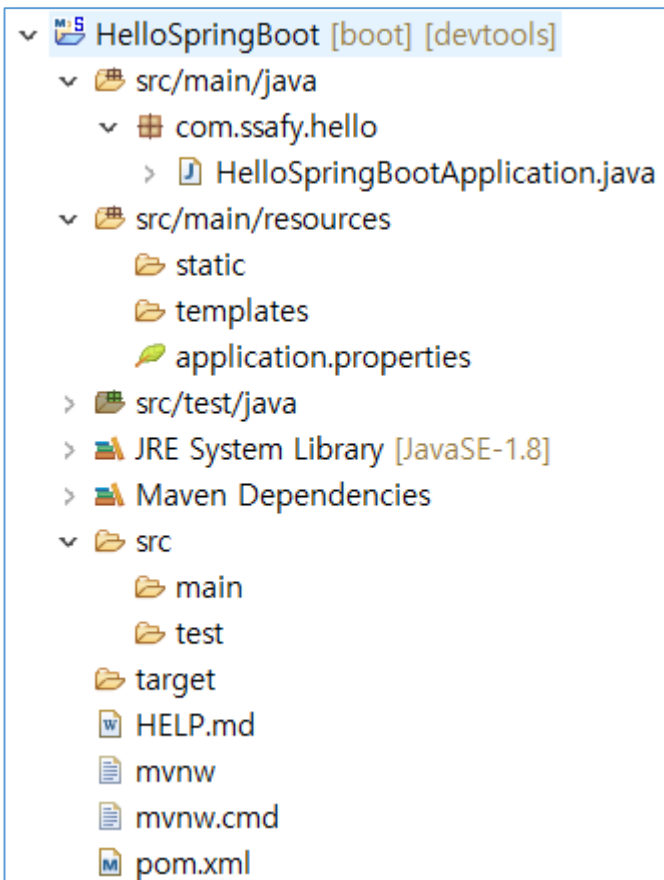
Base Url:

Full Url:

< Back Next > Finish Cancel

✓ SpringBoot Project 생성.

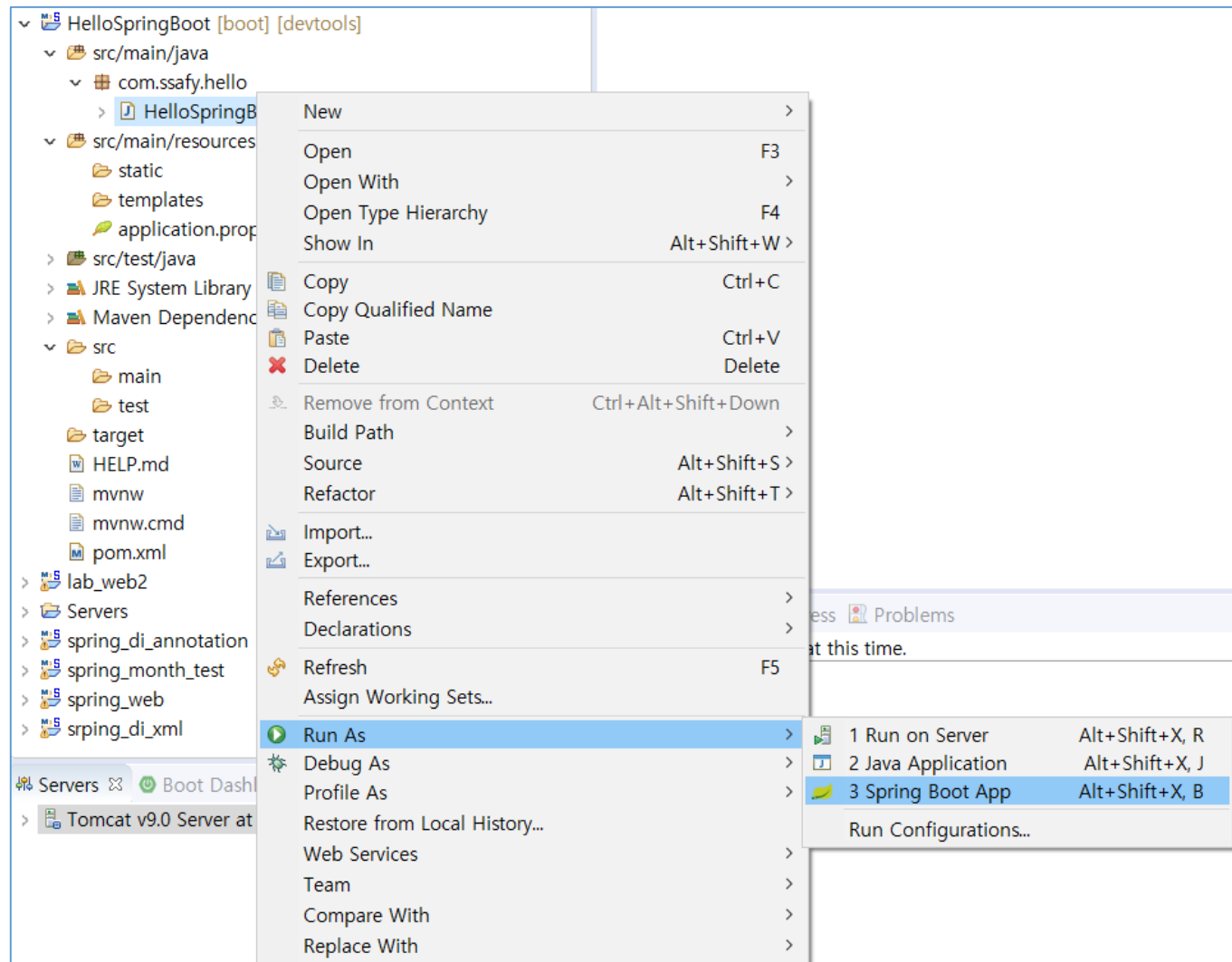
- project 생성 구조 및 주요 구성 폴더/파일.



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 여러가지 property를 정의한 file
src/main	jsp등의 리소드 directory

✓ SpringBoot Project 생성.

■ 실행.



✓ SpringBoot Project 생성.

▪ 실행 콘솔 창.

```

  ____ _
 / ___ \| | | |
/ /   \| |_| |
\ \   /| | | |
 \___/\|_|_|_|
:: Spring Boot ::      (v2.3.4.RELEASE)

```

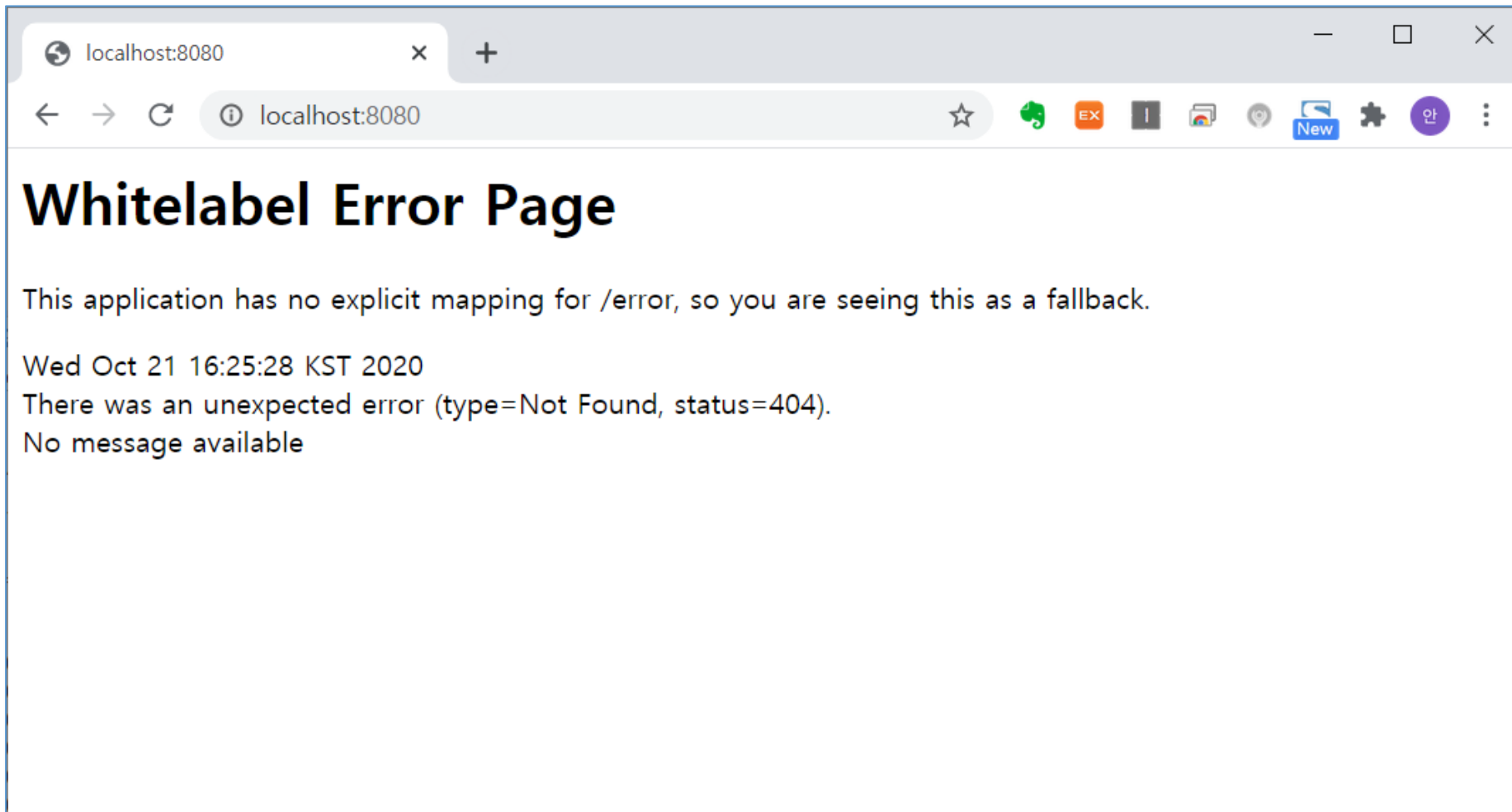
```

2020-10-21 16:22:11.924 INFO 12192 --- [ restartedMain] c.s.hello.HelloSpringBootApplication : Starting HelloSpringBootApplication on DESKTOP-OA2I3IE with PID 12192
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-properties' to 'true'
2020-10-21 16:22:12.000 INFO 12192 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.org.springframework.web' property
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 for 3.11s)

```

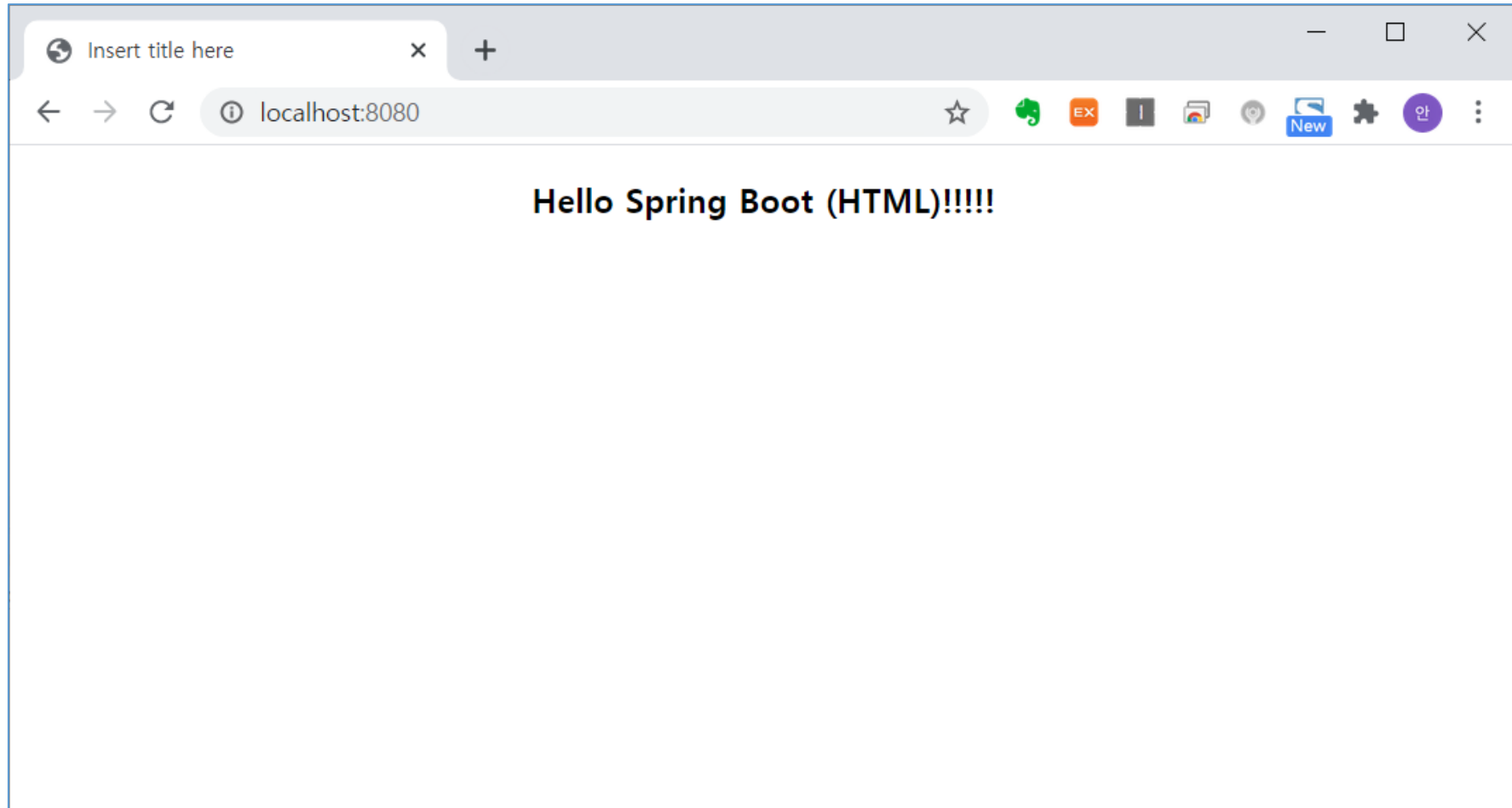
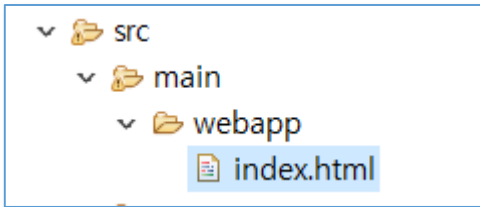
✓ SpringBoot Project 생성.

- 브라우저를 열고 <http://localhost:8080> 실행.



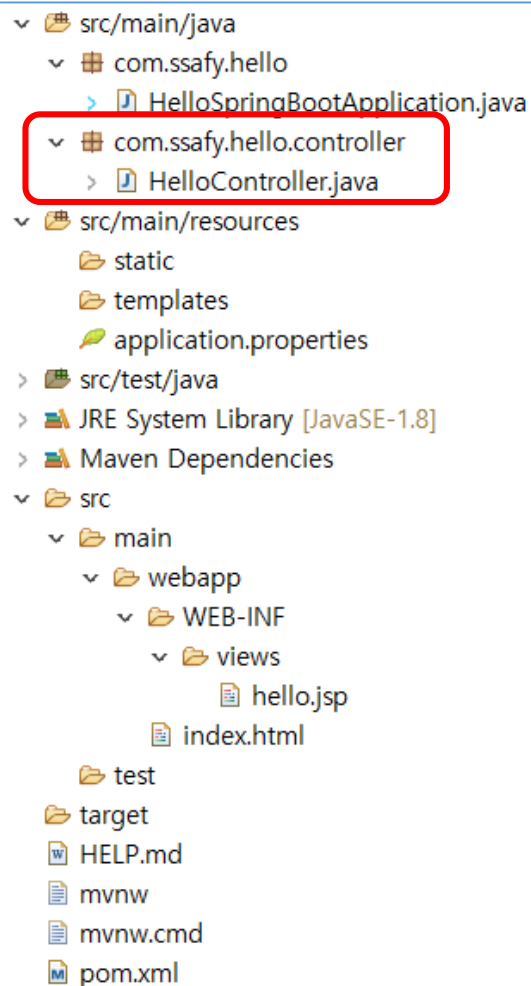
✓ SpringBoot Project 생성.

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



✓ SpringBoot Project 생성 : jsp test.

- com/ssafy/hello/controller/HelloController 생성.



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

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

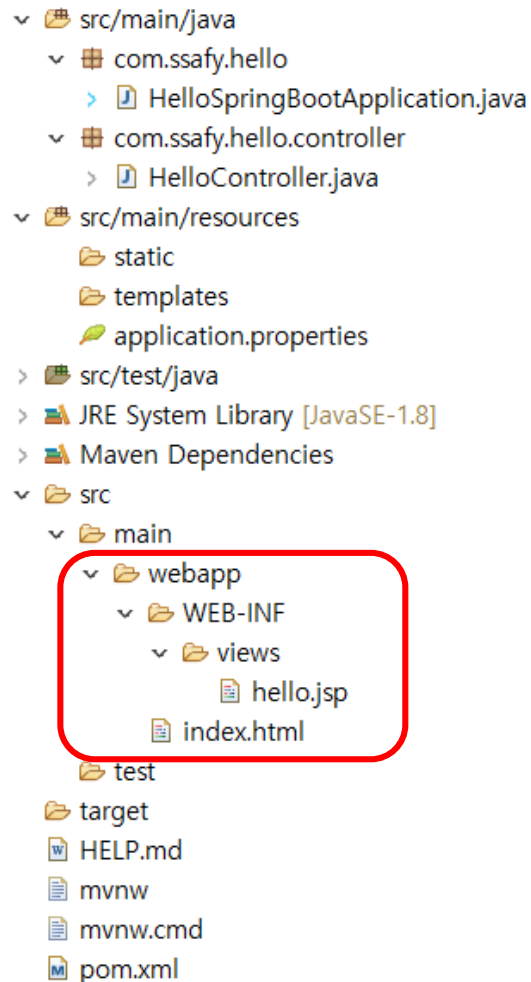
@Controller
public class HelloController {

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

}
```

✓ SpringBoot Project 생성 : jsp test.

- webapp/WEB-INF/views/hello.jsp 생성.

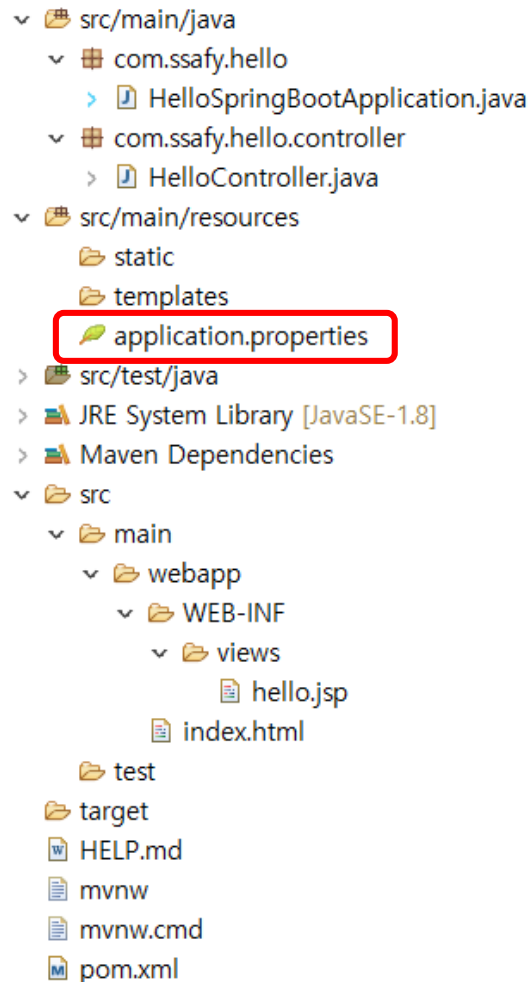


```
src/main/java
├── com.ssafy.hello
│   ├── HelloSpringBootApplication.java
│   └── com.ssafy.hello.controller
│       └── HelloController.java
├── src/main/resources
│   ├── static
│   ├── templates
│   └── application.properties
├── src/test/java
├── JRE System Library [JavaSE-1.8]
├── Maven Dependencies
├── 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 생성 : jsp test.

- application.properties file에 추가.



Web ContextRootPath and PortNumber Settings

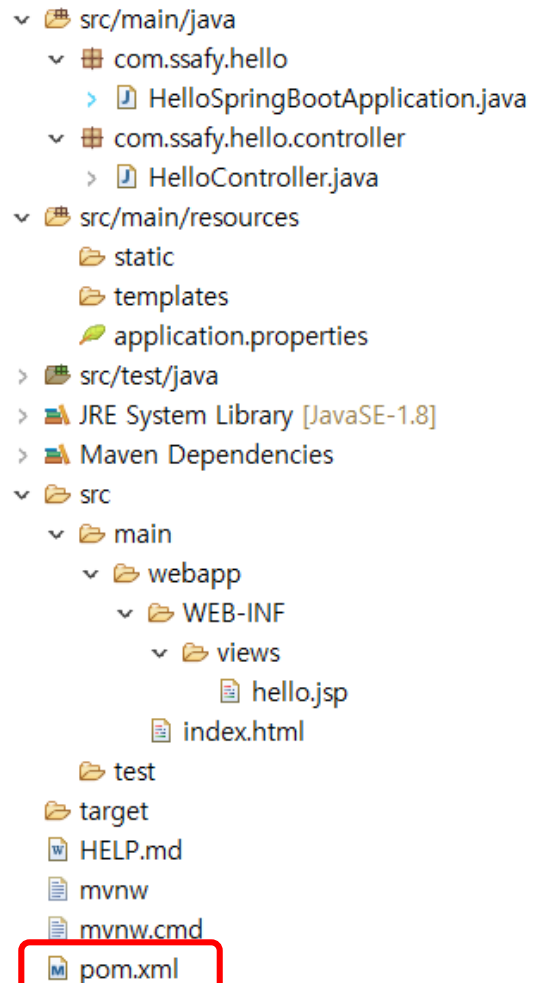
```
server.servlet.context-path=/ssafy  
server.port=80
```

JSP Path (ViewResolver)

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

✓ SpringBoot Project 생성 : jsp test.

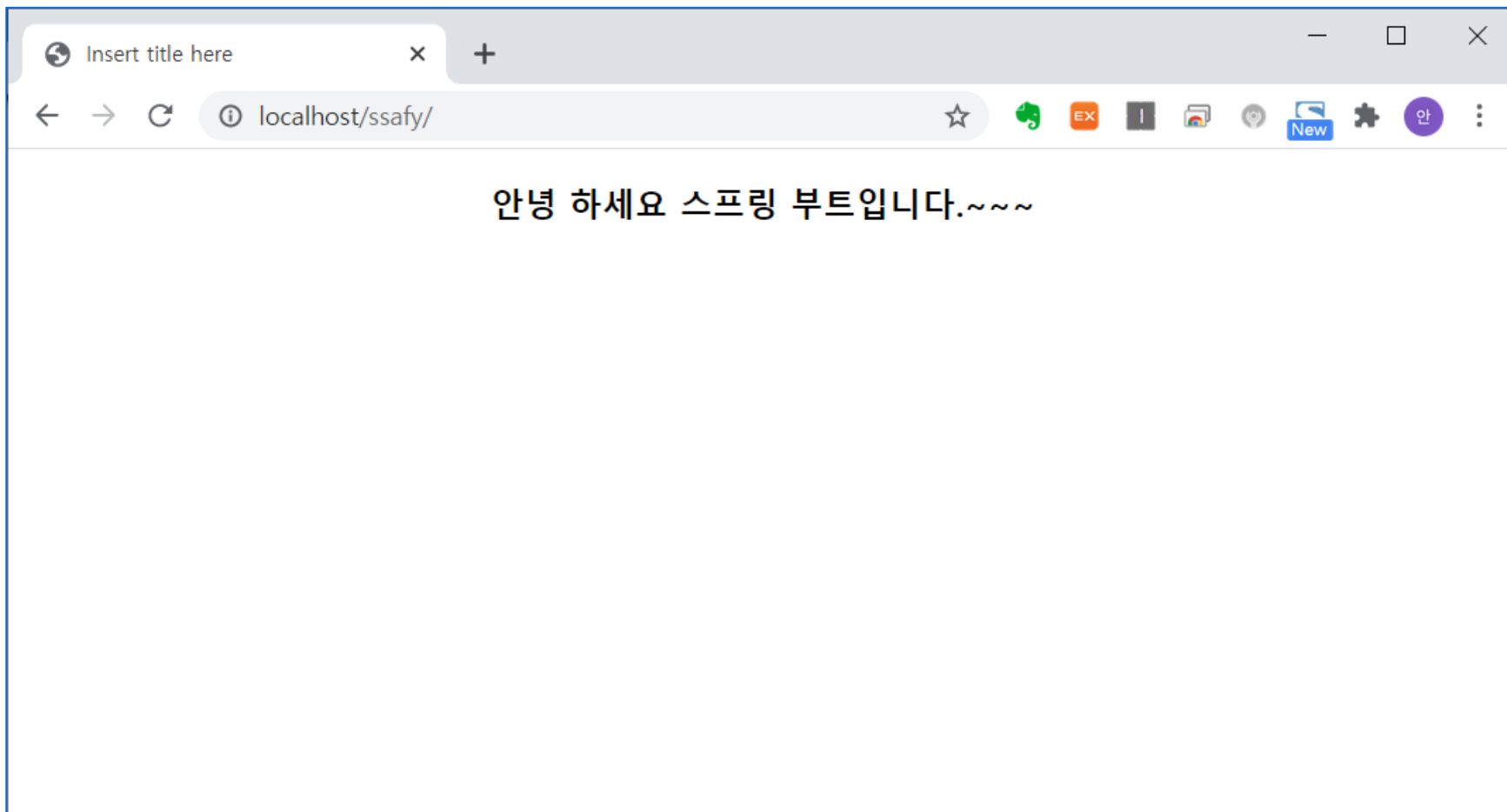
- pom.xml에 dependency 추가.



```
<!-- 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 생성.

- HelloSpringBootApplication 실행.
- 브라우저 : <http://localhost/ssafy>



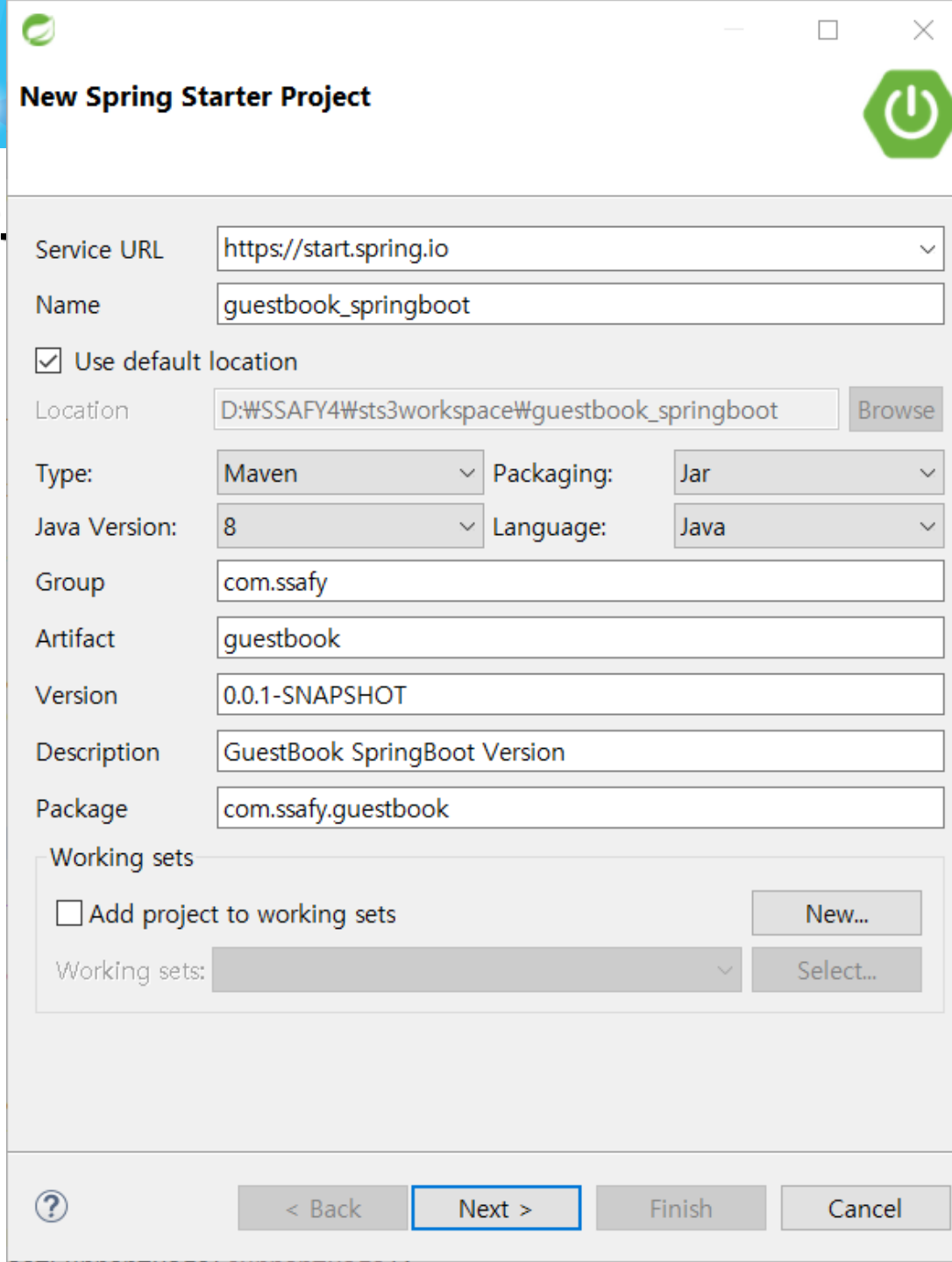
이어서..

삼성 청년 SW 아카데미

방명록 생성

삼성 청년 SW 아카데미

✓ guestbook_springboot.



The image shows a 'New Spring Starter Project' dialog box from an IDE. It contains various input fields for project configuration. The 'Service URL' is set to 'https://start.spring.io'. The 'Name' is 'guestbook_springboot'. The 'Location' is 'D:\SSAFY4\sts3\workspace\guestbook_springboot'. The 'Type' is 'Maven', 'Packaging' is 'Jar', 'Java Version' is '8', and 'Language' is 'Java'. The 'Group' is 'com.ssafy', 'Artifact' is 'guestbook', 'Version' is '0.0.1-SNAPSHOT', 'Description' is 'GuestBook SpringBoot Version', and 'Package' is 'com.ssafy.guestbook'. There is a 'Working sets' section with an unchecked checkbox 'Add project to working sets' and a 'New...' button. Below it is a 'Working sets:' dropdown menu and a 'Select...' button. At the bottom, there are navigation buttons: '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

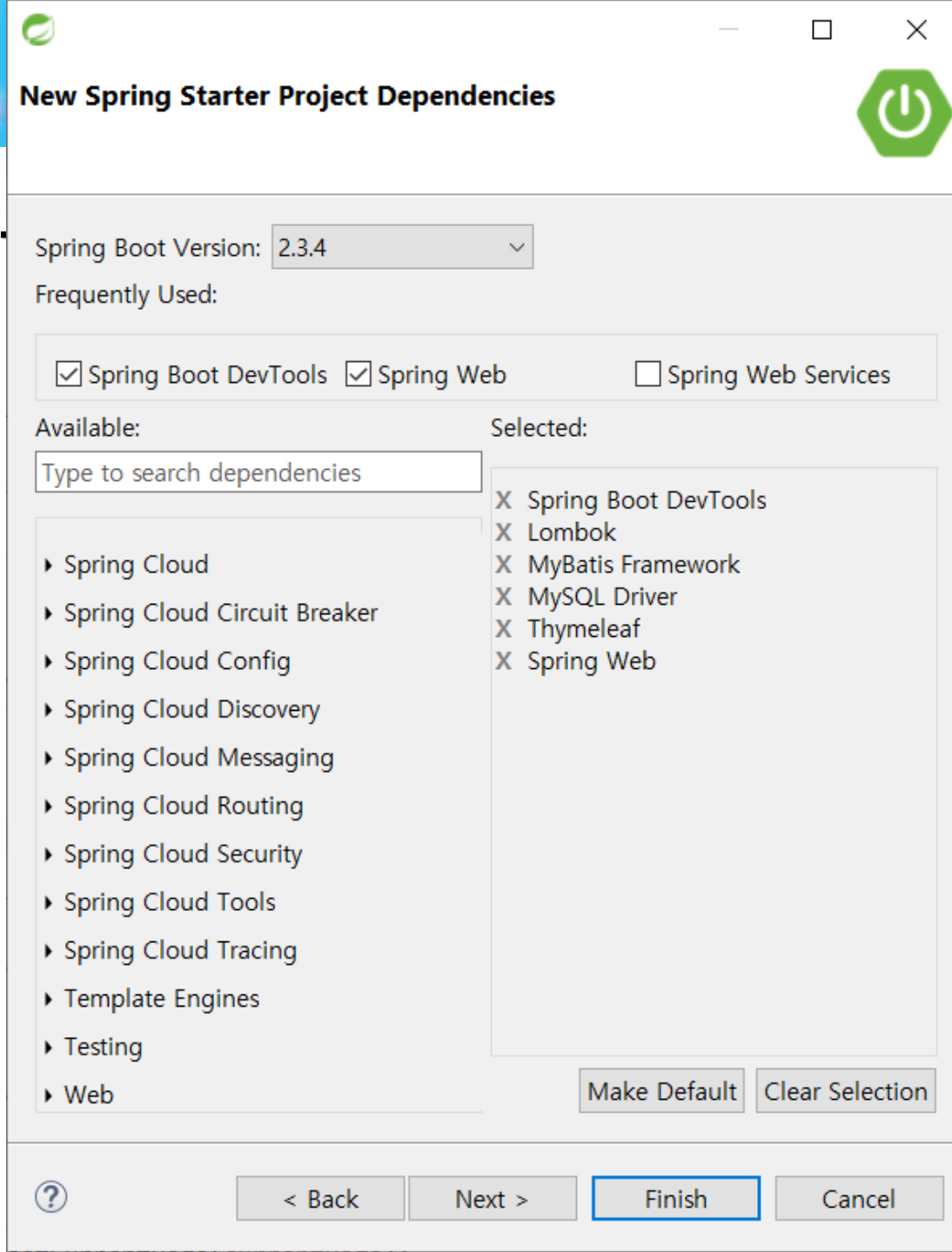
Package:

Working sets

☐ Add project to working sets

Working sets:

✓ guestbook_springboot.



The image shows a 'New Spring Starter Project Dependencies' dialog box. At the top, it has a title bar with a green Spring logo, a maximize button, and a close button. Below the title bar is a green power button icon. The main content area is divided into two sections: 'Available:' and 'Selected:'. The 'Available:' section has a search bar with the placeholder text 'Type to search dependencies' and a list of dependencies with expandable arrows. The 'Selected:' section has a list of dependencies with 'X' marks next to them. At the bottom of the dialog, there are buttons for 'Make Default', 'Clear Selection', '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'. A help icon (?) is also present in the bottom left corner.

New Spring Starter Project Dependencies

Spring Boot Version: 2.3.4

Frequently Used:

☒ Spring Boot DevTools ☒ Spring Web ☐ Spring Web Services

Available:

Type to search dependencies

- Spring Cloud
- Spring Cloud Circuit Breaker
- 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

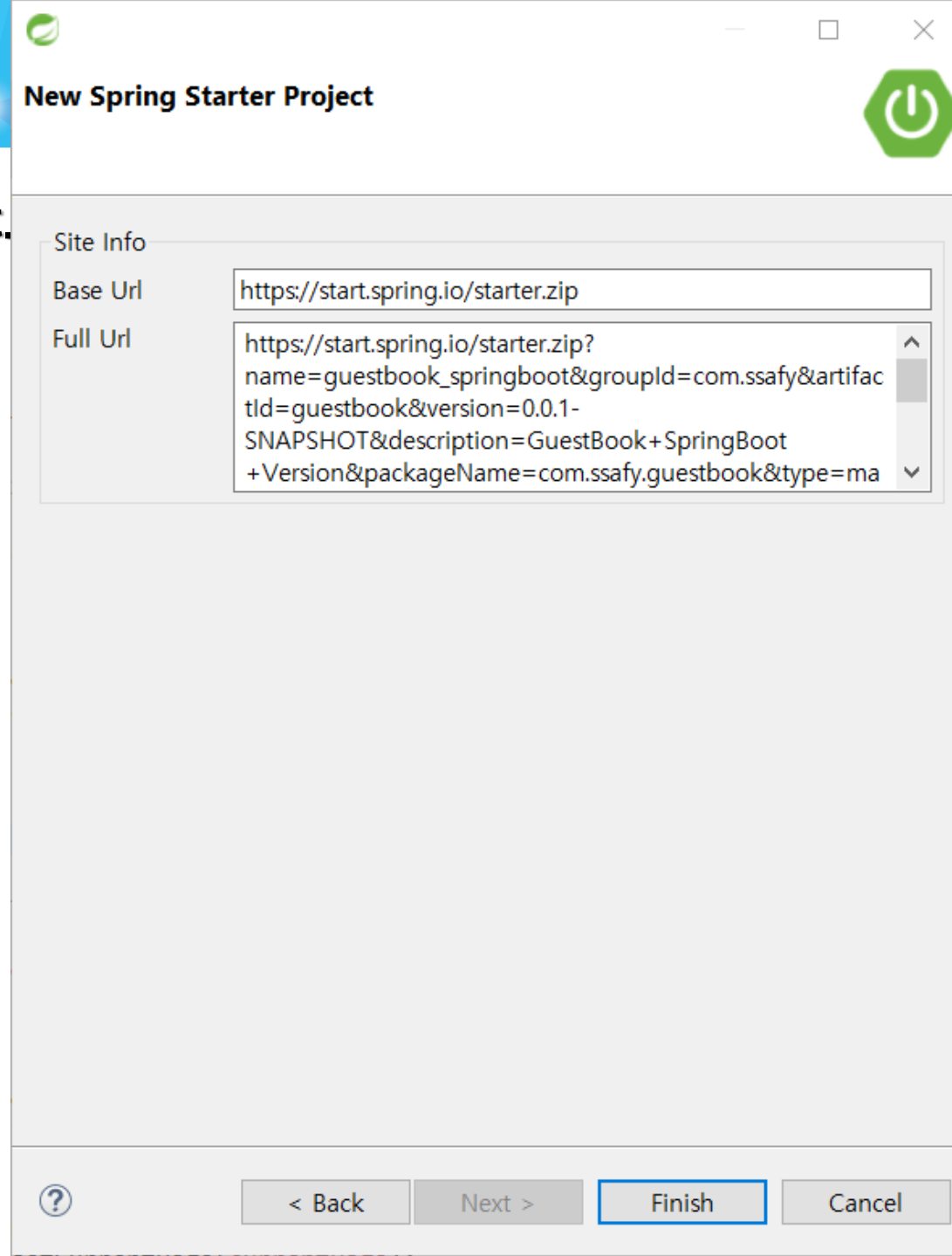
Selected:

- X Spring Boot DevTools
- X Lombok
- X MyBatis Framework
- X MySQL Driver
- X Thymeleaf
- X Spring Web

Make Default Clear Selection

? < Back Next > Finish Cancel

✓ guestbook_springboot.



The image shows a 'New Spring Starter Project' dialog box. It has a title bar with a green Spring logo, a maximize button, a close button, and a green power button icon. The main area is titled 'Site Info' and contains two text input fields. The 'Base Url' field contains 'https://start.spring.io/starter.zip'. The 'Full Url' field contains a longer URL: 'https://start.spring.io/starter.zip?name=guestbook_springboot&groupId=com.ssafy&artifactId=guestbook&version=0.0.1-SNAPSHOT&description=GuestBook+SpringBoot+Version&packageName=com.ssafy.guestbook&type=ma'. At the bottom, there are four buttons: a help button (question mark in a circle), a '< Back' button, a 'Next >' button, and a 'Finish' button (which is highlighted with a blue border). A 'Cancel' button is also present to the right of the 'Finish' button.

New Spring Starter Project

Site Info

Base Url:

Full Url:

? < Back Next > Finish Cancel

이어서..

삼성 청년 SW 아카데미

Swagger

삼성 청년 SW 아카데미

✓ Swagger를 이용한 REST API 문서화.

- 프로젝트 개발 시 일반적으로 FrontEnd 개발자와 BackEnd 개발자가 분리.
- FrontEnd 개발자의 경우 화면과 로직에 집중을 하고 BackEnd 개발자가 만든 문서 API를 보며 데이터 처리를 하게 된다.
- 이때 개발 상황의 변화에 따른 API의 추가 또는 변경할 때 마다 문서에 적용하는 불편함 발생.
- 이 문제를 해결하기 위해 Swagger를 사용.

✓ Swagger?

- 간단한 설정으로 프로젝트의 API 목록을 웹에서 확인 및 테스트 할 수 있게 해주는 Library.
- Swagger를 사용하면 Controller에 정의되어 있는 모든 URL을 바로 확인할 수 있다.
- API 목록 뿐 아니라 API의 명세 및 설명도 볼 수 있으며, 또한 API를 직접 테스트해 볼 수도 있다.

✓ Swagger 적용.

- pom.xml에 swagger2 dependency 추가.

```
<!-- Swagger2 추가 -->
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
  <version>2.9.2</version>
</dependency>
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger-ui</artifactId>
  <version>2.9.2</version>
</dependency>
```

✓ Swagger 적용.

- SwaggerConfiguration.java

```
@Configuration
@EnableSwagger2
public class SwaggerConfiguration {

    @Bean
    public Docket postsApi() {
        return new Docket(DocumentationType.SWAGGER_2).groupName("public-api").apiInfo(apiInfo())
            .select().paths(postPaths()).build();
    }

    private Predicate<String> postPaths() {
        return or(regex("/api/posts.*"), regex("/api.*"));
    }

    private ApiInfo apiInfo() {
        return new ApiInfoBuilder().title("SSAFY API")
            .description("<h3>SSAFY API Reference for Developers</h3>Swagger를 이용한 HRM API")
            .contact(new Contact("SSAFY", "https://edu.ssafy.com", "ssafy@ssafy.com")).license("SSAFY License")
            .licenseUrl("https://www.ssafy.com/ksp/jsp/swp/etc/swpPrivacy.jsp").version("1.0").build();
    }
}
```

✓ Swagger 적용.

- Swagger 적용 될 Controller.

```
@CrossOrigin(origins = {"*"}, maxAge = 6000)
@RestController
@RequestMapping("/api")
@Api(value="SSAFY")
public class EmployeeController {
    @Autowired
    private EmployeeService employeeService;

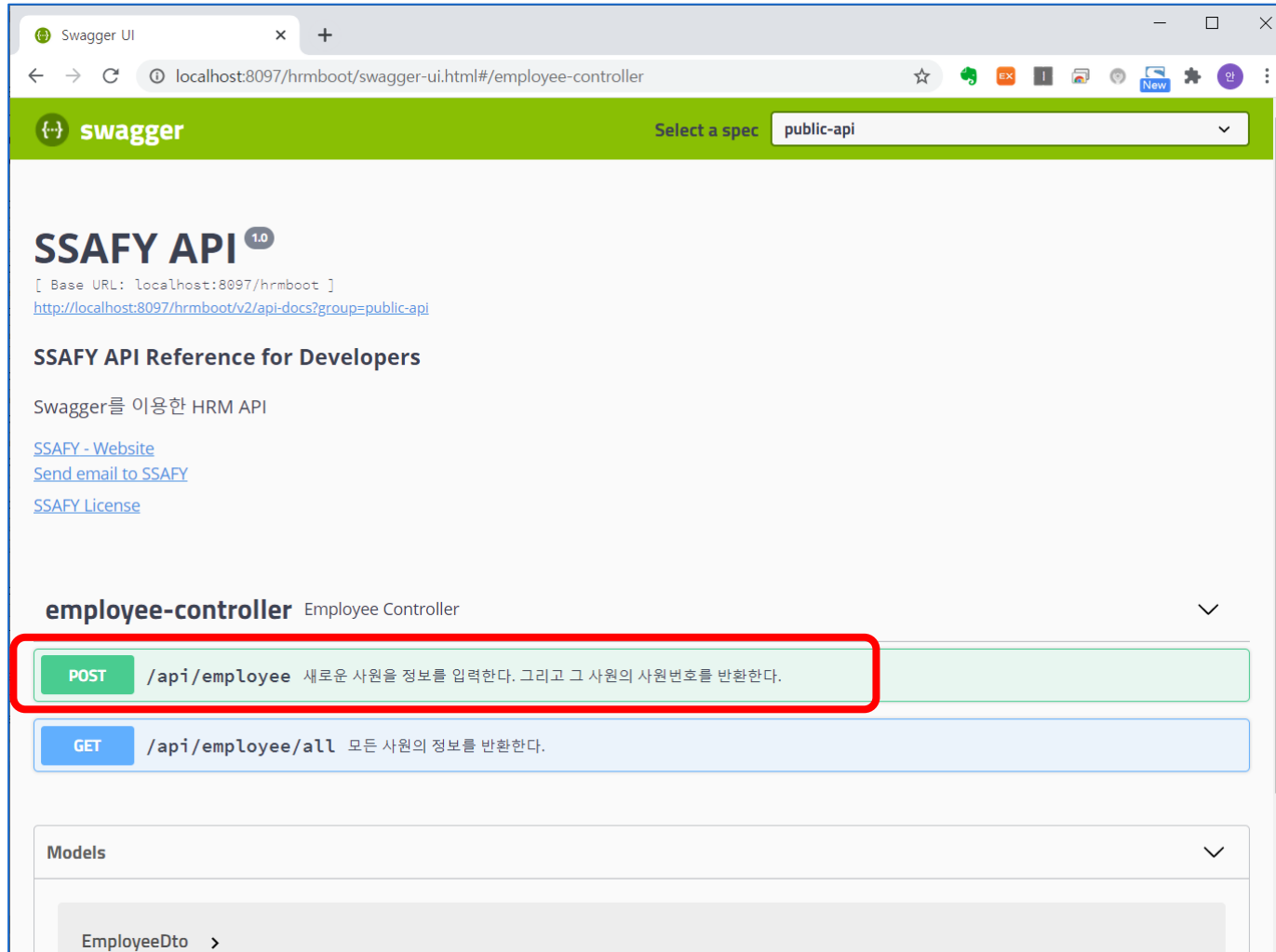
    @ApiOperation(value = "모든 사원의 정보를 반환한다.", response = List.class)
    @RequestMapping(value = "/employee/all", method = RequestMethod.GET)
    public ResponseEntity<List<EmployeeDto>> findAllEmployees() throws Exception {
        List<EmployeeDto> emps = employeeService.findAllEmployees();
        if (emps.isEmpty()) {
            return new ResponseEntity(HttpStatus.NO_CONTENT);
        }
        return new ResponseEntity<List<EmployeeDto>>(emps, HttpStatus.OK);
    }

    @ApiOperation(value = "새로운 사원을 정보를 입력한다. 그리고 그 사원의 사원번호를 반환한다.", response = NumberResult.class)
    @RequestMapping(value = "/employee", method = RequestMethod.POST)
    public ResponseEntity<NumberResult> addEmployee(@RequestBody EmployeeDto dto) throws Exception {
```

✓ Swagger 적용.

- Swagger 적용 API 실행화면.

<http://localhost:8097/hrmboot/swagger-ui.html>



✓ Swagger 적용.

- Swagger 적용 될 Model(Dto).

```
@ApiModelProperty(value = "EmployeeDto : 사원정보", description = "사원의 상세 정보를 나타낸다.")  
public class EmployeeDto implements Serializable {  
    @ApiModelProperty(value = "사원 번호")  
    private int id;  
    @ApiModelProperty(value = "사원 이름")  
    private String name;  
    private String mailid;  
    private String start_date;
```

✓ Swagger 적용.

- Swagger 적용 될 Model(Dto) 적용 화면.

employee-controller Employee Controller

Models

EmployeeDto : 직원정보 ▼ {

description:

사원의 상세 정보를 나타낸다.

cname

string

commission_pct

number(\$double)

dept_id

integer(\$int32)

dept name

string

id

integer(\$int32)

직원 번호

mailid

string

manager_id

integer(\$int32)

name

string

직원 이름

salary

number(\$double)

start_date

string

title

string

}

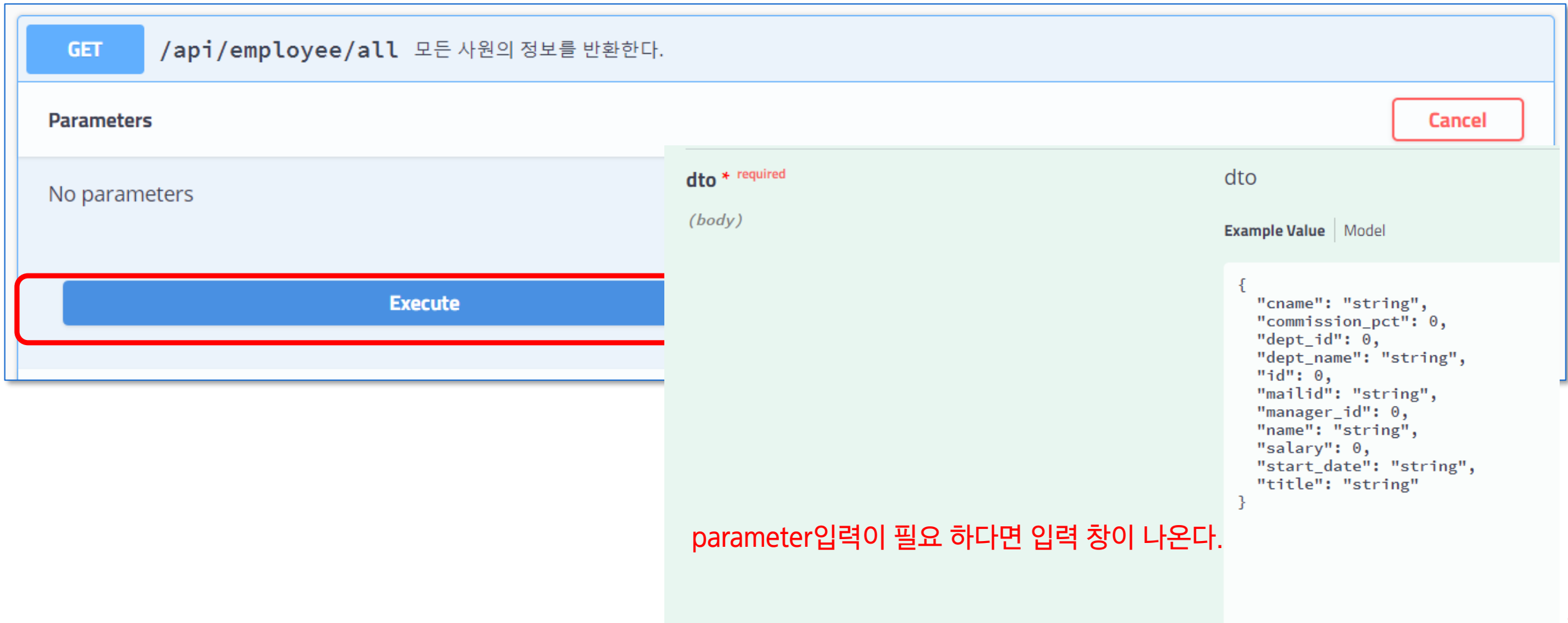
✓ Swagger 적용.

- Swagger API를 이용한 테스트

The image shows a Swagger UI interface for a REST API. At the top, there is a blue button labeled 'GET' followed by the endpoint path `/api/employee/all` and a description in Korean: '모든 사원의 정보를 반환한다.' (Returns information of all employees). Below this, there is a section titled 'Parameters' which contains the text 'No parameters'. On the right side of the 'Parameters' section, there is a button labeled 'Try it out' which is highlighted with a red rectangular border.

✓ Swagger 적용.

- Swagger API를 이용한 테스트



GET /api/employee/all 모든 사원의 정보를 반환한다.

Parameters

No parameters

Execute

Cancel

dto * required
(body)

dto

Example Value | Model

```
{
  "cname": "string",
  "commission_pct": 0,
  "dept_id": 0,
  "dept_name": "string",
  "id": 0,
  "mailid": "string",
  "manager_id": 0,
  "name": "string",
  "salary": 0,
  "start_date": "string",
  "title": "string"
}
```

parameter입력이 필요 하다면 입력 창이 나온다.

✓ Swagger 적용.

- Swagger API를 이용한 테스트

The screenshot displays the Swagger UI interface. At the top, the 'Request URL' is set to `http://localhost:8097/hrmboot/api/employee/all`. Below this, the 'Server response' section shows a 'Code' of 200 and 'Details' for the 'Response body'. The response body is a JSON array containing two employee objects. The first object has an id of 1, name '박구곤', mailid 'sky98', start_date '1990-03-03', manager_id 0, title '사장', dept_id 118, salary 5000, commission_pct 0, dept_name '인사부', and cname '박구곤'. The second object has an id of 2, name '손명섭', mailid '2xxsig', start_date '1990-03-08', manager_id 1, title '기획부장', dept_id 110, salary 3000, commission_pct 0, dept_name '기획부', and cname '손명섭'. A 'Download' button is visible next to the response body. At the bottom, the 'Response headers' are listed: `connection: keep-alive`, `content-type: application/json`, `date: Wed, 21 Oct 2020 11:38:48 GMT`, `keep-alive: timeout=60`, and `transfer-encoding: chunked`.

```
Request URL
http://localhost:8097/hrmboot/api/employee/all

Server response
Code    Details
200     Response body
[
  {
    "id": 1,
    "name": "박구곤",
    "mailid": "sky98",
    "start_date": "1990-03-03",
    "manager_id": 0,
    "title": "사장",
    "dept_id": 118,
    "salary": 5000,
    "commission_pct": 0,
    "dept_name": "인사부",
    "cname": "박구곤"
  },
  {
    "id": 2,
    "name": "손명섭",
    "mailid": "2xxsig",
    "start_date": "1990-03-08",
    "manager_id": 1,
    "title": "기획부장",
    "dept_id": 110,
    "salary": 3000,
    "commission_pct": 0,
    "dept_name": "기획부",
    "cname": "손명섭"
  }
]

Response headers
connection: keep-alive
content-type: application/json
date: Wed, 21 Oct 2020 11:38:48 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

내일 방송에서 만나요!

삼성 청년 SW 아카데미