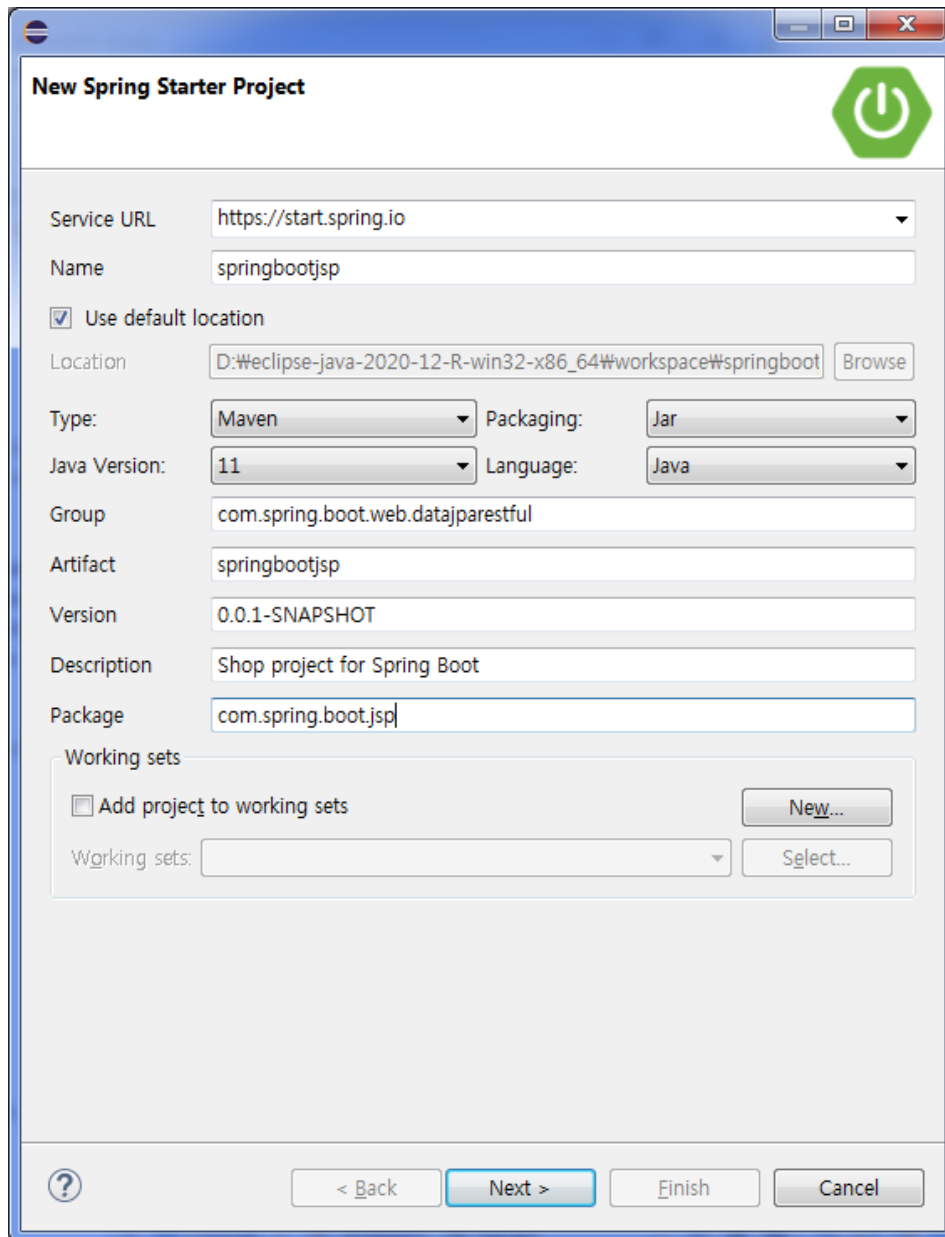


1. springbootjsp

가. 프로젝트 생성




The screenshot shows the 'New Spring Starter Project' dialog box in the Eclipse IDE. The dialog has a title bar with standard window controls and a green power button icon in the top right corner. The main content area contains several fields and options for configuring a new project:

- Service URL:** A dropdown menu with 'https://start.spring.io' selected.
- Name:** A text field containing 'springbootjsp'.
- Use default location:** A checked checkbox.
- Location:** A text field showing 'D:\Eclipse-java-2020-12-R-win32-x86_64\workspace\springboot' with a 'Browse' button to its right.
- Type:** A dropdown menu with 'Maven' selected.
- Packaging:** A dropdown menu with 'Jar' selected.
- Java Version:** A dropdown menu with '11' selected.
- Language:** A dropdown menu with 'Java' selected.
- Group:** A text field containing 'com.spring.boot.web.datajparestful'.
- Artifact:** A text field containing 'springbootjsp'.
- Version:** A text field containing '0.0.1-SNAPSHOT'.
- Description:** A text field containing 'Shop project for Spring Boot'.
- Package:** A text field containing 'com.spring.boot.jsp'.
- Working sets:** A section with a checkbox 'Add project to working sets' (unchecked), a 'New...' button, a 'Working sets:' dropdown menu, and a 'Select...' button.

At the bottom of the dialog, there is a row of buttons: a help icon (?), '< Back', 'Next >', 'Finish', and 'Cancel'.

New Spring Starter Project Dependencies



Spring Boot Version: 2.7.17

Frequently Used:

☐ H2 Database

☐ Lombok

☐ MySQL Driver

☐ Spring Boot DevTools

☐ Spring Configuration Processor

☐ Spring Data JPA

☒ Spring Web

☐ Thymeleaf

☐ Validation

Available:Selected:

web

▼ Messaging

☐ WebSocket

▼ Template Engines

☐ Thymeleaf

☐ Apache Freemarker

☐ Mustache

▼ Testing

☐ Testcontainers

▼ Web

☒ Spring Web

☐ Spring Reactive Web

☐ Spring Web Services

☐ Jersey

☐ Vaadin

X Spring Web

Make Default

Clear Selection

?

< Back

Next >

Finish

Cancel

나. pom.xml 수정

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

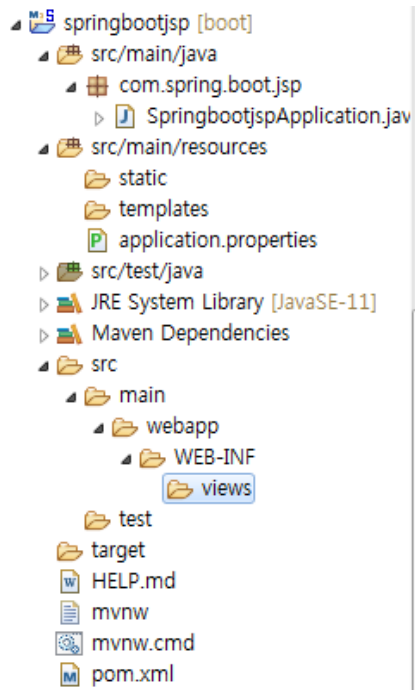
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-tomcat</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.tomcat.embed</groupId>
        <artifactId>tomcat-embed-jasper</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
```

다. application.properties 수정

```
server.port=8082
server.servlet.context-path=/jsp
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.j네
```

라. 프로젝트 모습



마. 서블릿 초기화

```
package com.spring.boot.jsp;
```

```
import org.springframework.boot.builder.SpringApplicationBuilder;
```

```
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
```

```
public class ServletInitializer extends SpringBootServletInitializer {
```

```
    @Override
```

```
    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {
```

```
        return builder.sources(SpringbootdynamicApplication.class);
```

```
    }
```

```
}
```

바. 컨트롤러 생성

```
package com.spring.boot.jsp;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.springframework.stereotype.Controller;
```

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

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
@Controller
```

```
public class TestController {
```

```
    @RequestMapping(value="/test", method = RequestMethod.GET)
```

```
    public ModelAndView getScoreInfo(@RequestParam String score1, @RequestParam  
String score2) {
```

```
        int sum = Integer.valueOf(score1) + Integer.valueOf(score2);
```

```
        System.out.println("합계점수: " + sum);
```

```
        ModelAndView mv = new ModelAndView();
```

```
        List<String> items = new ArrayList<String>();
```

```
        items.add("김형태1");
```

```
        items.add("김형태2");
```

```
        items.add("김형태3");
```

```
        mv.addObject("list", items);
```

```
        mv.addObject("sum", sum);
```

```
        mv.setViewName("list");
```

```
        return mv;
```

```
    }
```

```
}
```

사. HTML 만들기: src/main/에 webapp 폴더를 만들고 index.html 생성

```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>TEST</title>
</head>
<body>
<h1>TEST</h1>
</body>
</html>
```

아. JSP 만들기: src/main/webapp에 WEB-INF/views 폴더를 만들고 list.jsp 생성

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>User List</title>
</head>
<body>
```

```
<table border="1">
<tr>
<th>성명</th>
</tr>
<c:forEach var="item" items="${list}">
<tr><!-- 첫번째 줄 시작 -->
<td>${item}</td>
</tr><!-- 첫번째 줄 끝 -->
</c:forEach>
<tr>
<th>${sum}</th>
</tr>
</table>
</body>
</html>
```

자. 실행: 쿼리 문자열을 주지 않으면 에러(400: Bad Request)

<http://localhost:8082/jsp/test?score1=101&score2=301>