



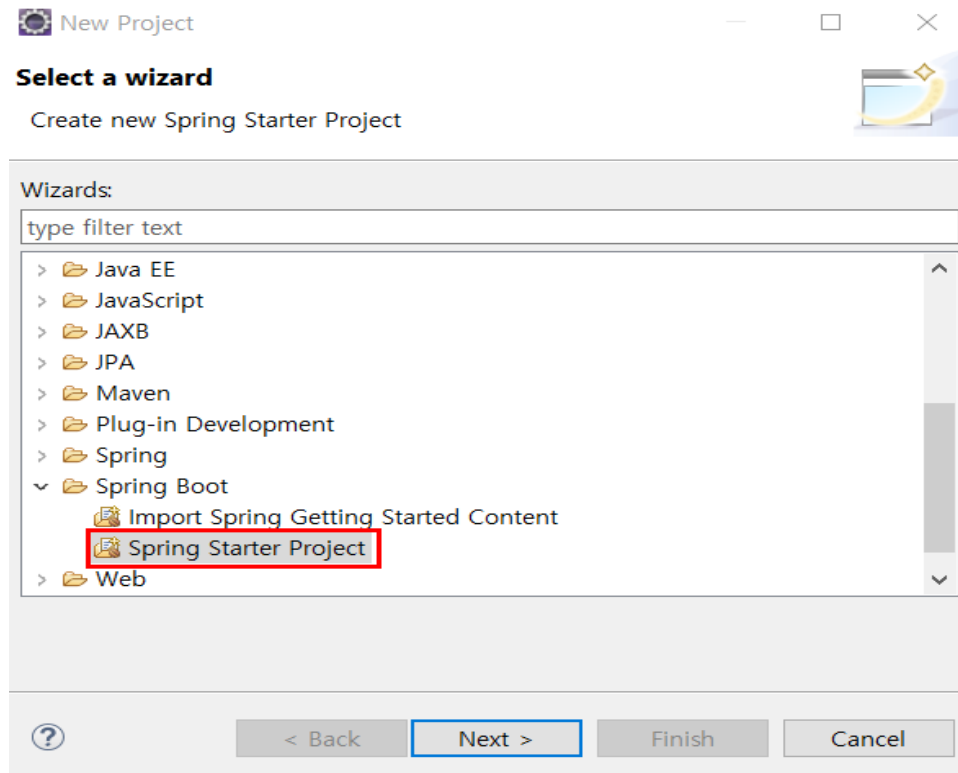
spring boot예제

안 화 수

spring boot 프로젝트

❖ boot01 project 생성

[File] – New - Project



spring boot 프로젝트

❖ boot01 project 생성

➤ Name : **boot01**

➤ 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

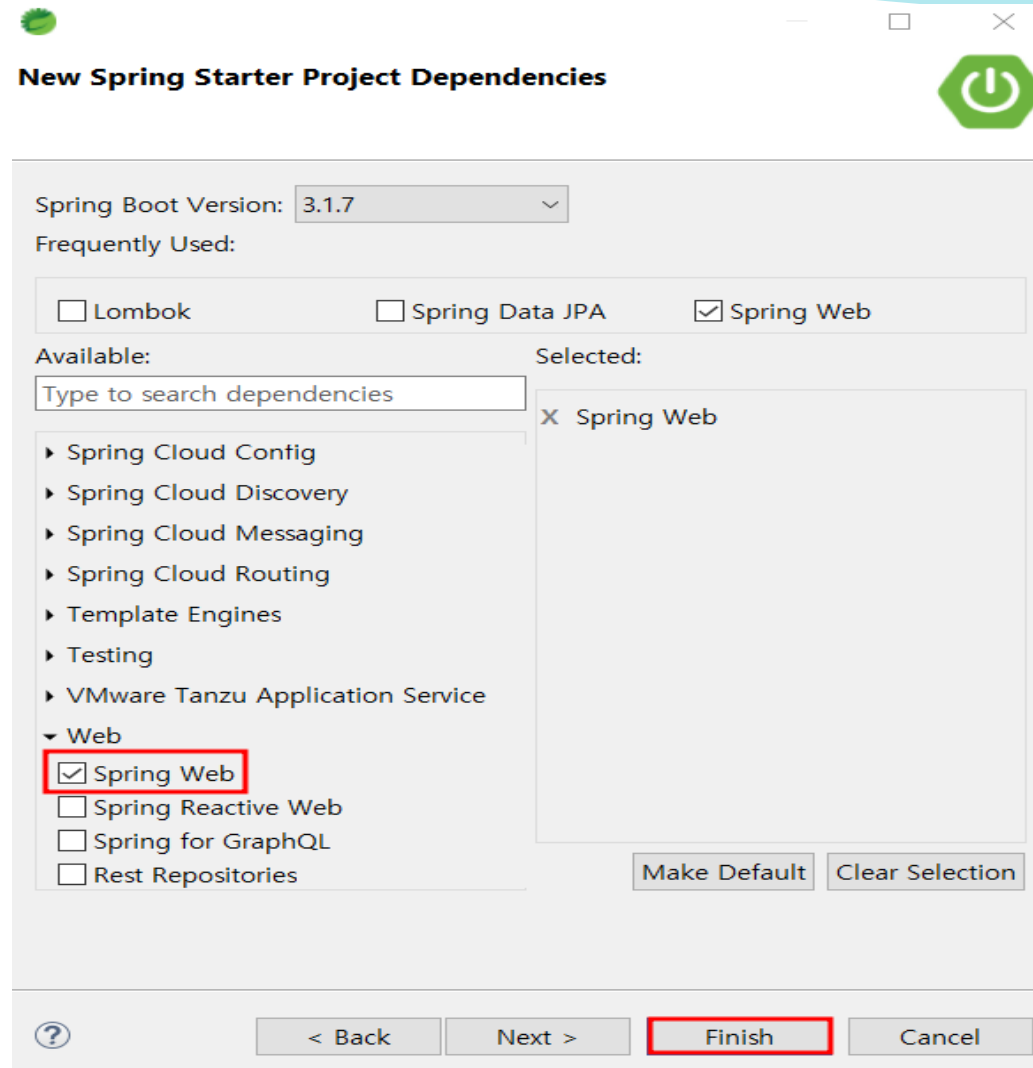
Working sets:

spring boot 프로젝트

❖ boot01 project 생성

➤ Web

Spring Web 체크



spring boot 서버 실행

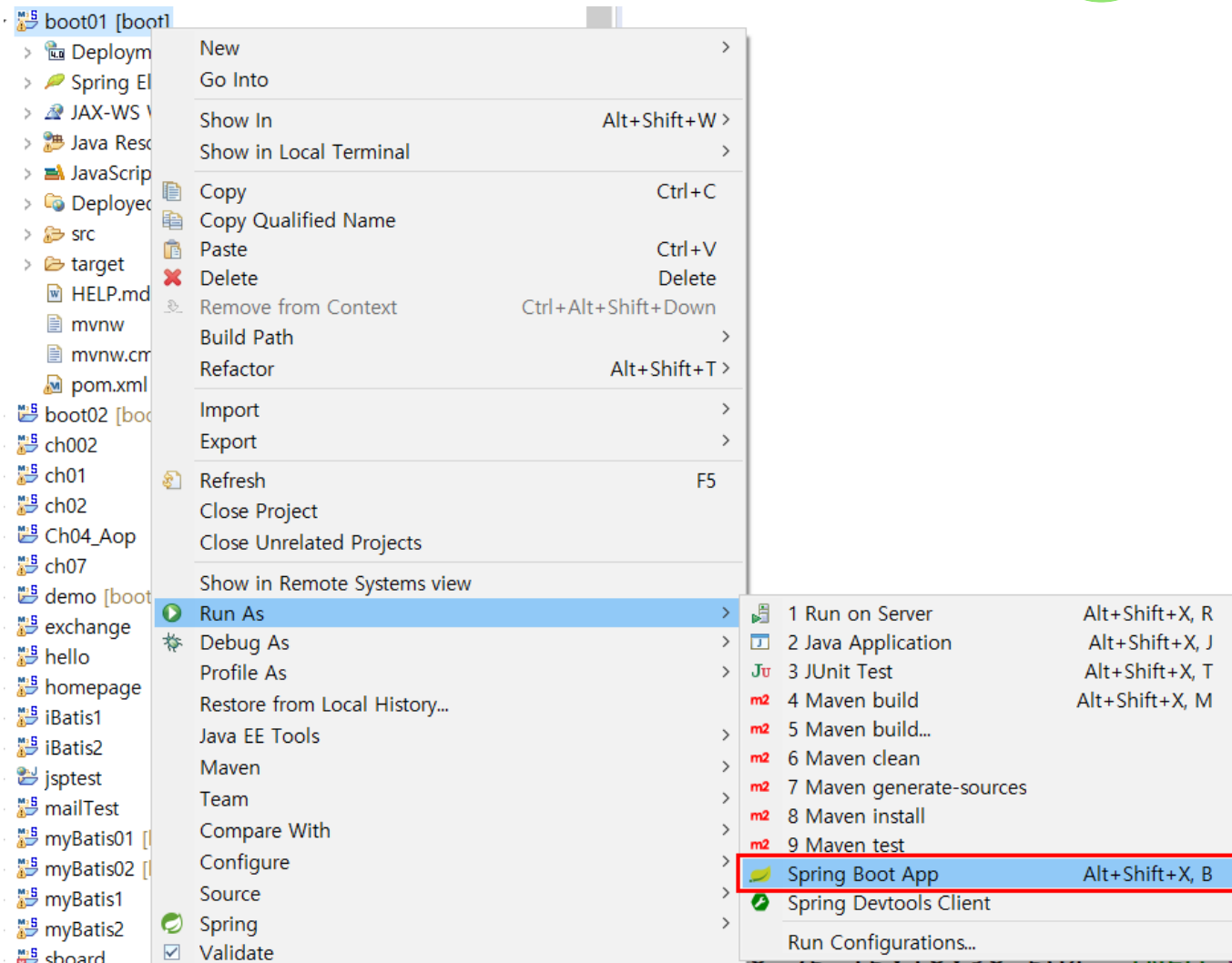
❖ boot01 project 실행

내장 Tomcat 실행

boot01 - 오른마우스 클릭

- Run As

- Spring Boot App



spring boot 서버 실행

boot01 project 실행 결과

다음과 같이 출력되면 성공~!!

```

  .
 / \ / \
( ( ) \
 \ \   )
  '   |
=====|_=====
:: Spring Boot ::      (v2.2.9.RELEASE)

2020-08-25 15:26:50.480 INFO 10868 --- [
2020-08-25 15:26:50.483 INFO 10868 --- [
2020-08-25 15:26:51.444 INFO 10868 --- [
2020-08-25 15:26:51.455 INFO 10868 --- [
2020-08-25 15:26:51.456 INFO 10868 --- [
2020-08-25 15:26:51.723 INFO 10868 --- [
2020-08-25 15:26:51.727 INFO 10868 --- [
2020-08-25 15:26:51.727 INFO 10868 --- [
2020-08-25 15:26:51.889 INFO 10868 --- [
2020-08-25 15:26:52.051 INFO 10868 --- [
2020-08-25 15:26:52.054 INFO 10868 --- [

main] com.example.demo.Boot01Application
main] com.example.demo.Boot01Application
main] o.s.b.w.embedded.tomcat.TomcatWebServer
main] o.apache.catalina.core.StandardService
main] org.apache.catalina.core.StandardEngine
main] org.apache.jasper.servlet.TldScanner
main] o.a.c.c.C.[Tomcat].[localhost].[/]
main] w.s.c.ServletWebServerApplicationContext
main] o.s.s.concurrent.ThreadPoolTaskExecutor
main] o.s.b.w.embedded.tomcat.TomcatWebServer
main] com.example.demo.Boot01Application

```

의존 라이브러리 추가

❖ pom.xml

pom.xml 파일에 필요한 의존 라이브러리를 추가한다.

```
<dependencies>
  <!-- 내장 Tomcat실행시 jsp 파일을 사용하기 위한 의존 라이브러리 -->
  <dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
  </dependency>
  <!-- jakarta jstl -->
  <dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>3.0.0</version>
  </dependency>
</dependencies>
```

환경 설정파일 수정

❖ 환경 설정 파일 수정

main / resources / **application.properties**

port

server.port=80

prefix and suffix

spring.mvc.view.prefix=/WEB-INF/views/

spring.mvc.view.suffix=.jsp

❖ webapp / **WEB-INF** / **views** 폴더 생성

Controller 생성

❖ Controller 생성 (1/2)

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

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

```
import java.io.IOException;
```

```
import java.util.Random;
```

```
import javax.servlet.http.HttpServletResponse;
```

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

```
import org.springframework.ui.Model;
```

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

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

Controller 생성

❖ Controller 생성 (2/2)

@Controller

```
public class SampleController {  
    @RequestMapping("/hi")  
    @ResponseBody  
    public void hello(HttpServletResponse response) throws IOException {  
        response.getWriter().print("Hello world~!!!");  
    }  
    @RequestMapping("/abc")  
    @ResponseBody  
    public String abc() {  
        return "hi abc";  
    }  
    @RequestMapping("/hello")  
    public String hello() {  
        return "hello";  
    }  
    @RequestMapping("/gugu")  
    public String gugu(Model model) {  
        Random r = new Random();  
        int dan = r.nextInt(8) + 2;  
        model.addAttribute("dan", dan);  
        return "gugu";  
    }  
}
```

// 2 ~ 9단

Controller 생성

❖ Controller 생성

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

```
package com.example.demo.controller;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;
```

@RestController

```
public class HelloController {  
    @RequestMapping("/welcome")  
    public String welcome() {  
        return "welcome to spring boot";  
    }  
}
```

View 생성

❖ View 생성

webapp / WEB-INF / views - hello.jsp 생성

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>hello</title>
</head>
<body>
```

hello.jsp page ~!!

```
</body>
</html>
```

View 생성

❖ View 생성

webapp / WEB-INF / views - gugu.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>
<html>
<head>
<meta charset="UTF-8">
<title>구구단 출력</title>
</head>
<body>

    [ ${dan} 단 ] <br><br>
    <c:forEach var="i" begin="1" end="9">
        ${dan} * ${i} = ${dan * i} <br>
    </c:forEach>

</body>
</html>
```

View 생성

❖ View 생성 : webapp - index.jsp 생성

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
    (document).ready(function(){
        $("#sel").change(function(){
            var sel = $("#sel").val();
            location.href=sel;
        });
    });
</script>
</head>
<body>
선택 :
<select id="sel">
    <option value="">메뉴</option>
    <option value="hi">hi</option>
    <option value="welcome">welcome</option>
    <option value="abc">abc</option>
    <option value="hello">hello</option>
    <option value="gugu">구구단</option>
</select>
</body>
</html>
```

boot01 프로젝트 접속

❖ boot01 프로젝트 접속 : 외부 Tomcat 실행할 경우

외부 Tomcat 실행후에 boot01 프로젝트 접속해보자~!!

<http://localhost/boot01/index.jsp> : webapp/index.jsp 파일이 자동으로 실행됨

<http://localhost/boot01/hi>

<http://localhost/boot01/welcome>

<http://localhost/boot01/abc>

<http://localhost/boot01/hello>

<http://localhost/boot01/gugu>

boot01 프로젝트 접속

❖ boot01 프로젝트 접속 : 내장 Tomcat 실행할 경우

내장 Tomcat 실행후에 boot01 프로젝트 접속해보자~!! (boot01 프로젝트명이 경로상에서 빠진다.)

<http://localhost/index.html> : resources/static/index.html 파일이 자동으로 실행됨

<http://localhost/hi>

<http://localhost/welcome>

<http://localhost/abc>

<http://localhost/hello>

<http://localhost/gugu>