

## spring boot 와 legacy mvc project 비교

1. **pom.xml** 스프링 프레임워크 버전 설정 없다. 부트에서 자동 설정한다.

2. **src/main/java/xxxxApplication.java** 에는 메인 메소드 (**public static void main(String[] args)**) 가 있는 것을 확인할 수 있다.

부트에서는 **@SpringBootApplication** 이 설정되어 있고 **main** 메소드 있는 클래스 실행시 스프링 부트 시작한다.

3. **run – spring boot app** 로 실행한다.

4. 내장 **tomcat** 서버는 **Main** 실행시 자동 시작한다. 오류 발생시 포트 번호 확인한다.

5. 스프링 부트 프로젝트에서는 **web.xml** 파일이 존재하지 않는다.

스프링 부트에서 제공하는 **SpringBootServletInitializer** 클래스가 대신한다.

6. **tomcat** 시작되면 현재 부트 프로젝트 컨텍스트명은 / 이다

**http://ip:Port/**

5. **jsp** 를 기본 **view** 로 하지 않으므로 추가 설정이 필요하다.

(**application.properties** , **pom.xml**, **webapp/WEB-INF/views ....**)

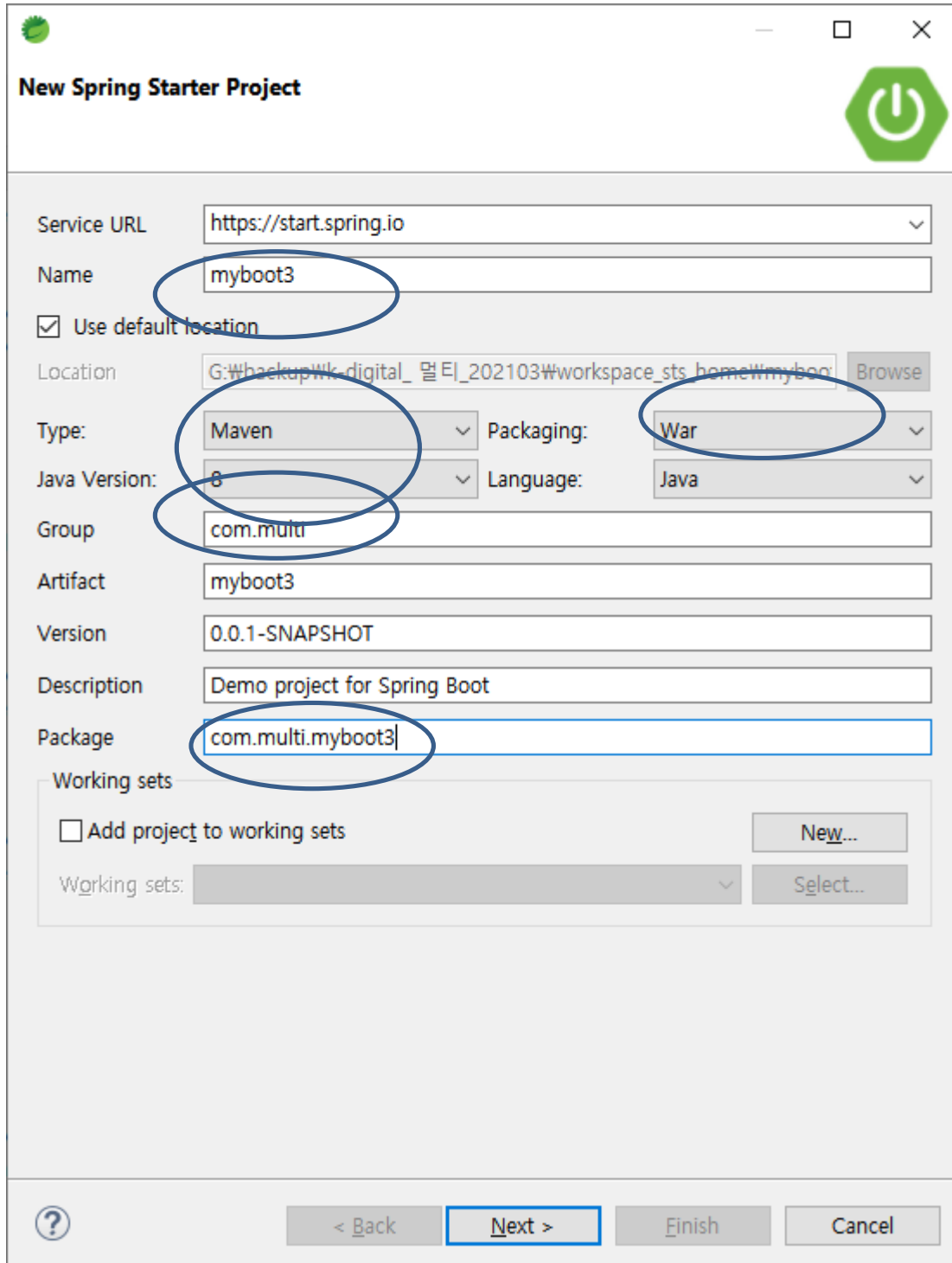
6. **src/main/resources/static** 폴더에는 리소스 파일 **html, gif, js, css** 등 저장했다.

7. **src/main/resources/templates** 폴더에는 템플릿 이 들어간다.

8. **src/main/resources/application.properties** 는 스프링 설정 **xml** 파일들을 대신하는 설정 파일이다.

## 스프링 **boot** 프로젝트 생성

file – spring starter project 선택




The image shows the 'New Spring Starter Project' dialog box in an IDE. The dialog is titled 'New Spring Starter Project' and features a green power button icon in the top right corner. The fields are as follows:

- Service URL: `https://start.spring.io`
- Name: `myboot3`
- ☒ Use default location
- Location: `G:\#backup\Wk_digital_멀티_202103\workspace_sts_home\myboot3` (with a 'Browse' button)
- Type: `Maven`
- Packaging: `War`
- Java Version: `8`
- Language: `Java`
- Group: `com.multi`
- Artifact: `myboot3`
- Version: `0.0.1-SNAPSHOT`
- Description: `Demo project for Spring Boot`
- Package: `com.multi.myboot3`

Below these fields is the 'Working sets' section:

- ☐ Add project to working sets (with a 'New...' button)
- Working sets: (empty list with a 'Select...' button)


At the bottom, there are four buttons: '?', '< Back', 'Next >', and 'Finish'. The 'Next >' button is highlighted with a blue border.



—

□

×

New Spring Starter Project Dependencies

Spring Boot Version: 2.4.5

Frequently Used:

☒ JDBC API

☒ MyBatis Framework

☒ Oracle Driver

☒ Spring Web

Available:

Type to search dependencies

Spring Cloud Messaging

Spring Cloud Routing

Spring Cloud Security

Spring Cloud Tools

Template Engines

Testing

VMware Tanzu Application Service

Web

☒ Spring Web

☐ Spring Reactive Web

☐ Rest Repositories

☐ Spring Session

☐ Rest Repositories HAL Explorer

☐ Spring HATEOAS

☐ Spring Web Services

☐ Jersey

☐ Vaadin

Selected:

X JDBC API


X MyBatis Framework

X Oracle Driver

X Spring Web

Make Default

Clear Selection



< Back

Java framework for building rich client apps based on \ components.

Guides

## spring boot mvc 실행하기

src/main/webapp

src/main/webapp/WEB-INF

src/main/webapp/WEB-INF/views

폴더 수동으로 생성(스프링 부트에서는 존재하지 않는다)

pom.xml

```
<!-- for jsp -->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
<!-- for jsp jstl -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
</dependency>
<!--for tomcat restart -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <optional>true</optional>
</dependency>
```

application.properties

server.port=9002

#view resolver

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

spring.mvc.view.suffix=.jsp

#html,js,gif등은 static폴더에 저장이 기본설정

TestController 작성

```
import javax.servlet.http.HttpServletRequest;
```

```

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
public class TestController {
    @ResponseBody
    @RequestMapping("/")
    public String home(){
        System.out.println("Hello Boot!!");
        return "Hello Boot!!";
    }

    @RequestMapping("/hello.do")
    public String hello(Model model){
        System.out.println("안녕하세요");
        model.addAttribute("message", "hello.jsp 입니다.");
        System.out.println(model.getAttribute("message"));
        return "hello";
    }
}

```

### hello.jsp 작성

```

<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert title here</title>
<script src="/xxx/static/jquery-3.2.1.min.js"></script>
</head>
<body>
<%=request.getAttribute("message") %>
</body>
</html>

```

브라우저 수동으로 연다  
주소 입력한다(컨텍스트명이 없다)

