

Spring boot



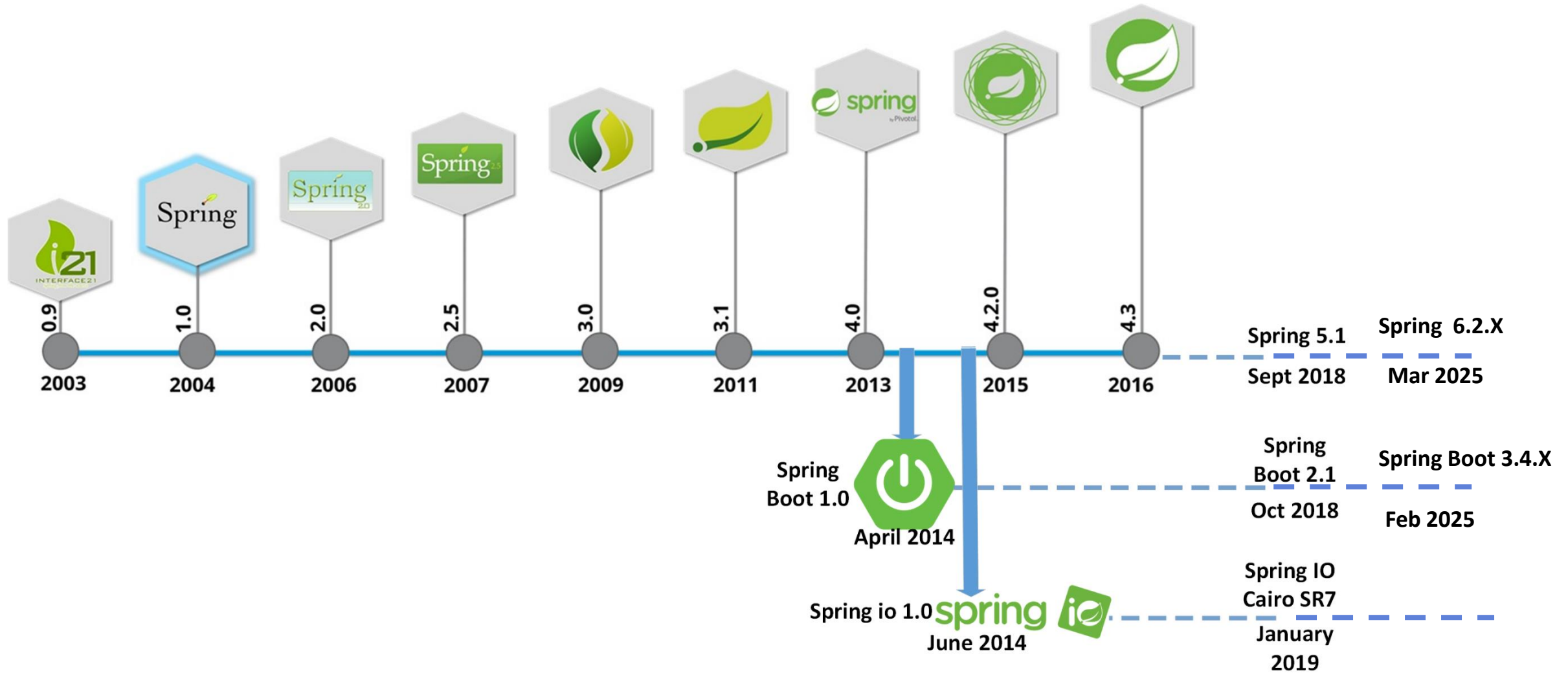
전제 목차

- ❖ Spring Boot 개요 및 환경설정
- ❖ Spring Boot Mybatis 및 jsp 사용
- ❖ Thymeleaf
- ❖ Spring Data JPA
- ❖ JPA 연관관계 설정
- ❖ Rest 방식 데이터 교환
- ❖ AOP와 트랜잭션
- ❖ 파일 업로드
- ❖ Spring Boot Security
- ❖ API 서버와 JWT

1. Spring Boot 개요 및 환경설정

1. 스프링 부트 소개
2. Spring vs Spring boot
3. Spring boot 프로젝트 생성

1. Spring Boot 소개



1. Spring Boot 소개

❖ Spring Boot란?

- 독립형 및 프로덕션 등급 Spring 기반 애플리케이션 구현에 사용되는 가벼운 오픈 소스 Java 기반 프레임워크
- 2002년에 처음 출시, Spring 플랫폼과 타사 라이브러리를 활용하여 개발자가 최소한의 번거로움으로 애플리케이션 구현을 시작할 수 있도록 지원

❖ Spring boot의 장점

- 복잡한 XML 환경설정 구성을 피하는데 도움
- 마이크로서비스 준비 프레임워크
- 강력한 일괄 처리 제공
- REST 엔드 포인트 관리에 도움
- 종속성 관리가 용이

<https://spring.io/projects/spring-boot>

2. Spring vs Spring boot

❖ 스프링과 스프링 부트

■ 스프링

- 자바 기반의 엔터프라이즈 애플리케이션
- 엔터프라이즈 애플리케이션 : 대규모 서비스를 뜻함
- 서버 성능, 안정성, 보안 등을 높은 수준으로 제공

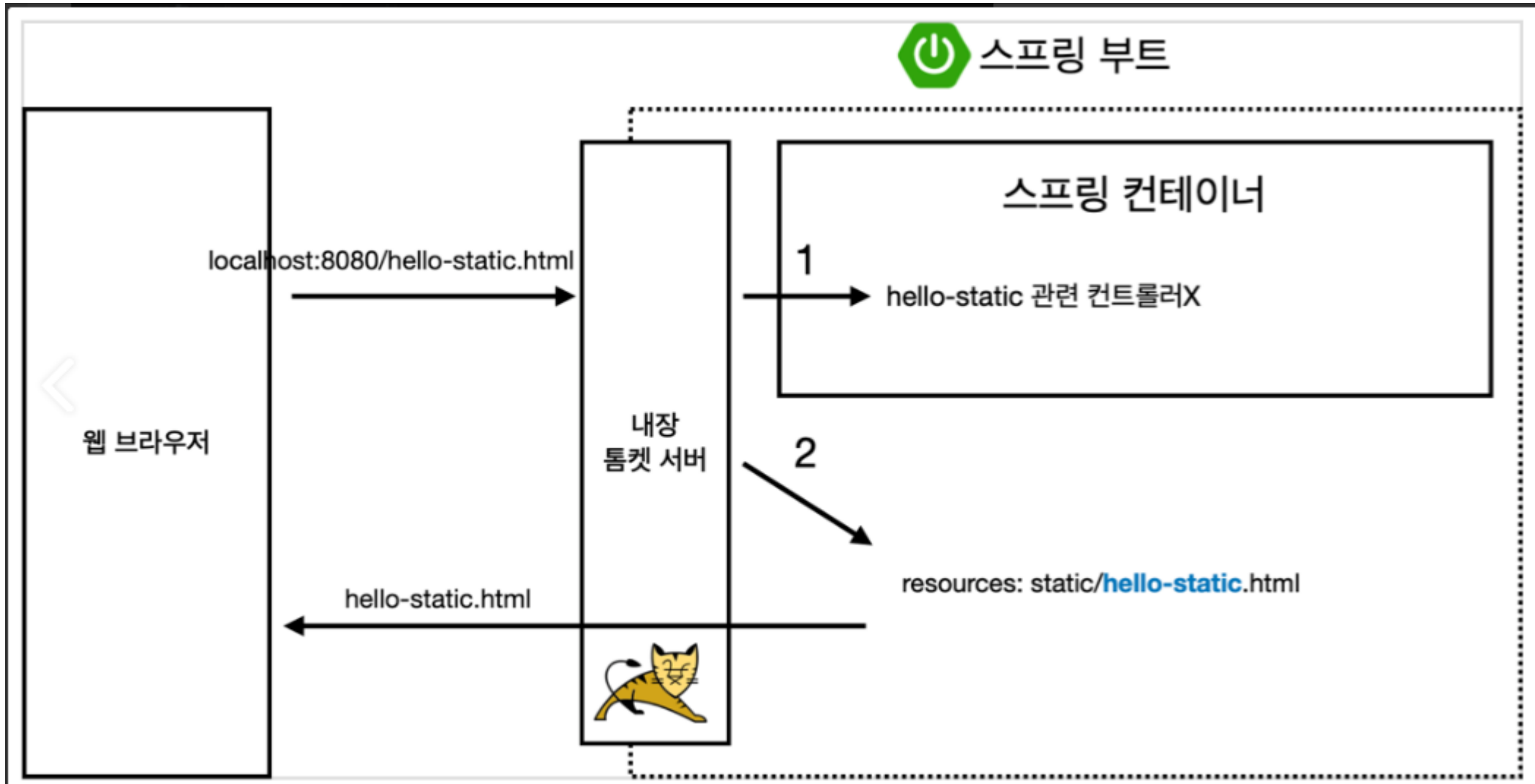
■ 스프링 부트

- 스프링의 복잡한 설정을 쉽게 만들어 출시한 일종의 스펀오프
 - 빠르게 스프링 프로젝트 설정 가능
 - 스타터를 사용하면 의존성 사용, 관리 용이

■ 차이점

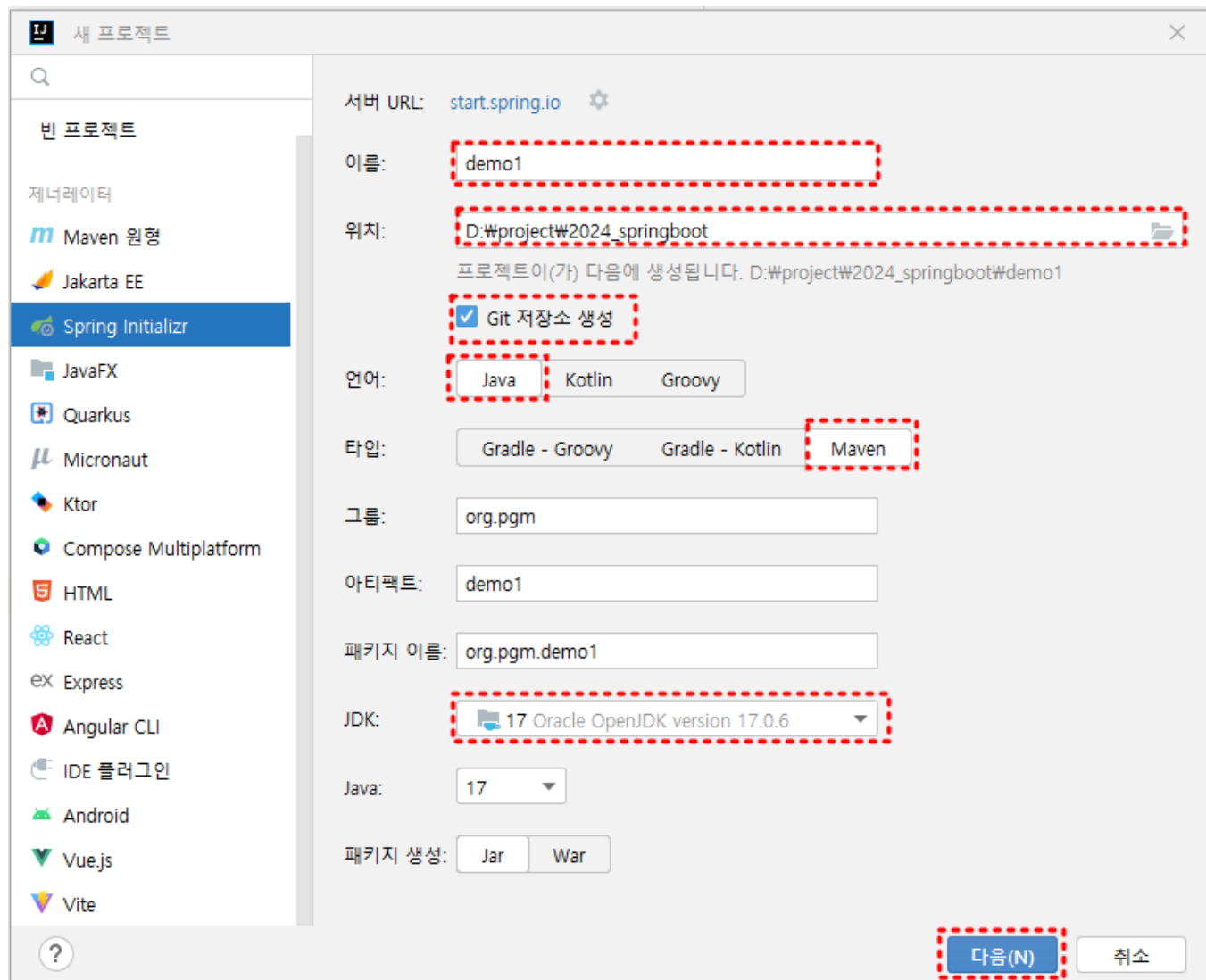
- 구성의 차이 : 스프링은 개발에 필요한 환경을 수동 구성과 스프링 부트는 자동 구성
- 내장 WAS의 유무 : 스프링 부트는 처음부터 WAS를 가지고 있음(tomcat)
 - WAS : 웹 애플리케이션을 실행하기 위한 장치

2. Spring vs Spring boot : SPRING BOOT 동작



3. Spring Boot 프로젝트 생성

- [file]-> [새로만들기] -> [Project] -[Spring Initializr] 선택 후 아래 그림과 같이 지정



The image shows the 'Spring Initializr' dialog box in an IDE. The left sidebar lists various project types, with 'Spring Initializr' selected. The main area contains fields for project configuration. Red dashed boxes highlight the following fields: '이름' (Name) set to 'demo1', '위치' (Location) set to 'D:\project\2024_springboot', the 'Git 저장소 생성' (Create Git repository) checkbox, '언어' (Language) set to 'Java', '타입' (Type) set to 'Maven', '그룹' (Group) set to 'org.pgm', '아티팩트' (Artifact) set to 'demo1', '패키지 이름' (Package name) set to 'org.pgm.demo1', 'JDK' set to '17 Oracle OpenJDK version 17.0.6', and the '다음(N)' (Next) button at the bottom right.

새 프로젝트

빈 프로젝트

제너레이터

Maven 원형

Jakarta EE

Spring Initializr

JavaFX

Quarkus

Micronaut

Ktor

Compose Multiplatform

HTML

React

Express

Angular CLI

IDE 플러그인

Android

Vue.js

Vite

서버 URL: start.spring.io

이름: demo1

위치: D:\project\2024_springboot

프로젝트 이름(가) 다음에 생성됩니다. D:\project\2024_springboot\demo1

☒ Git 저장소 생성

언어: Java Kotlin Groovy

타입: Gradle - Groovy Gradle - Kotlin Maven

그룹: org.pgm

아티팩트: demo1

패키지 이름: org.pgm.demo1

JDK: 17 Oracle OpenJDK version 17.0.6

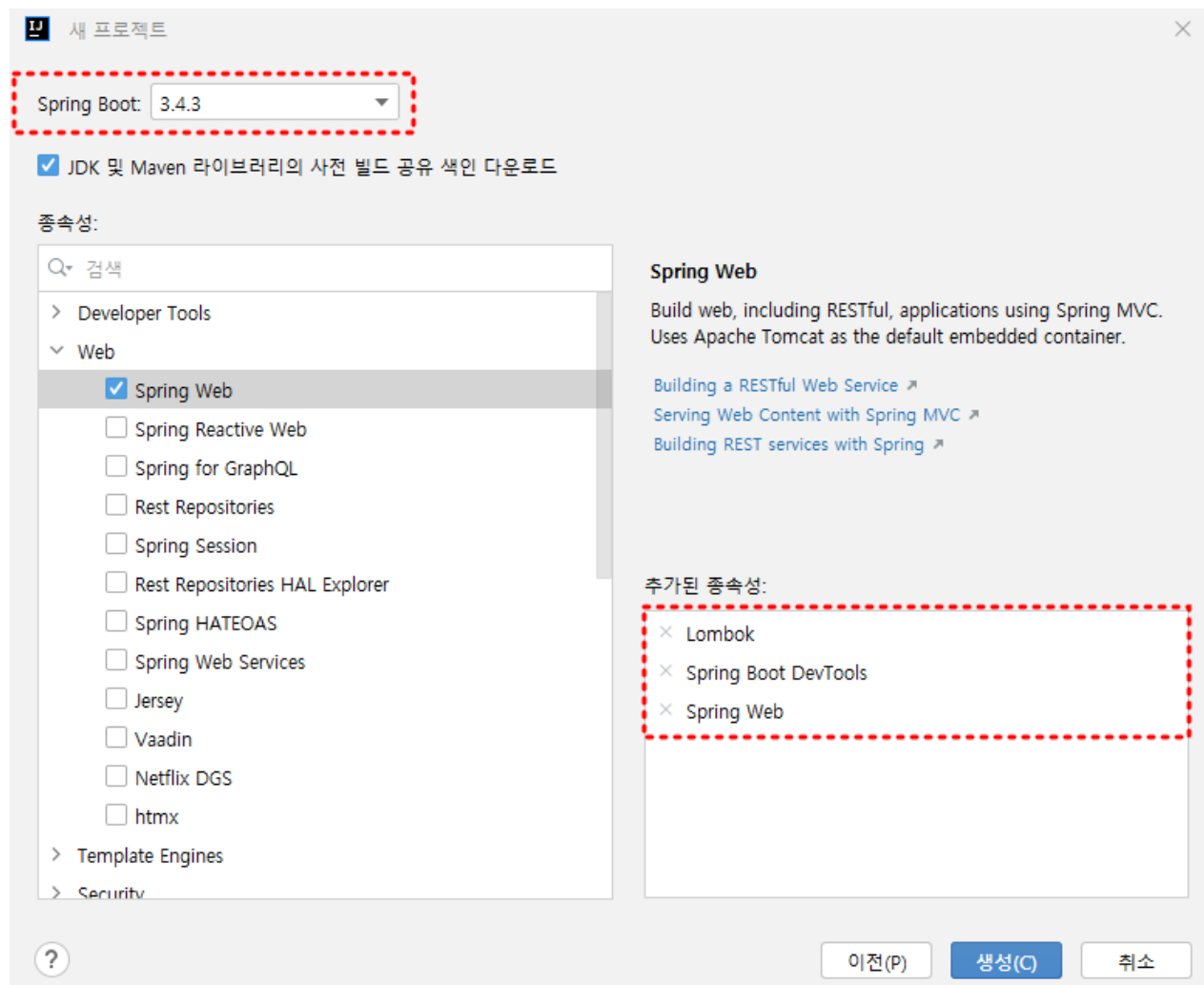
Java: 17

패키지 생성: Jar War

다음(N) 취소

3. Spring Boot 프로젝트 생성

❖ 버전 선택 및 종속성 추가



The screenshot shows the 'New Project' (새 프로젝트) window in an IDE. At the top, the 'Spring Boot' version is set to '3.4.3', which is highlighted with a red dashed box. Below this, the checkbox for 'JDK 및 Maven 라이브러리의 사전 빌드 공유 색인 다운로드' is checked. The 'Dependencies' (종속성) section on the left has a search bar and a list of options. 'Spring Web' is selected with a checkmark, while other options like 'Spring Reactive Web', 'Spring for GraphQL', 'Rest Repositories', 'Spring Session', 'Rest Repositories HAL Explorer', 'Spring HATEOAS', 'Spring Web Services', 'Jersey', 'Vaadin', 'Netflix DGS', 'htmx', 'Template Engines', and 'Security' are unchecked. On the right, the 'Spring Web' description is shown, along with links for 'Building a RESTful Web Service', 'Serving Web Content with Spring MVC', and 'Building REST services with Spring'. Below this, the 'Added Dependencies' (추가된 종속성) section contains a list with 'Lombok', 'Spring Boot DevTools', and 'Spring Web', all of which are highlighted with a red dashed box. At the bottom, there are three buttons: '이전(P)' (Previous), '생성(C)' (Generate), and '취소' (Cancel).

새 프로젝트

Spring Boot: 3.4.3

☒ JDK 및 Maven 라이브러리의 사전 빌드 공유 색인 다운로드

종속성:

Q 검색

- > Developer Tools
- ▼ Web
 - ☒ Spring Web
 - ☐ Spring Reactive Web
 - ☐ Spring for GraphQL
 - ☐ Rest Repositories
 - ☐ Spring Session
 - ☐ Rest Repositories HAL Explorer
 - ☐ Spring HATEOAS
 - ☐ Spring Web Services
 - ☐ Jersey
 - ☐ Vaadin
 - ☐ Netflix DGS
 - ☐ htmx
- > Template Engines
- > Security

Spring Web

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

[Building a RESTful Web Service](#)

[Serving Web Content with Spring MVC](#)

[Building REST services with Spring](#)

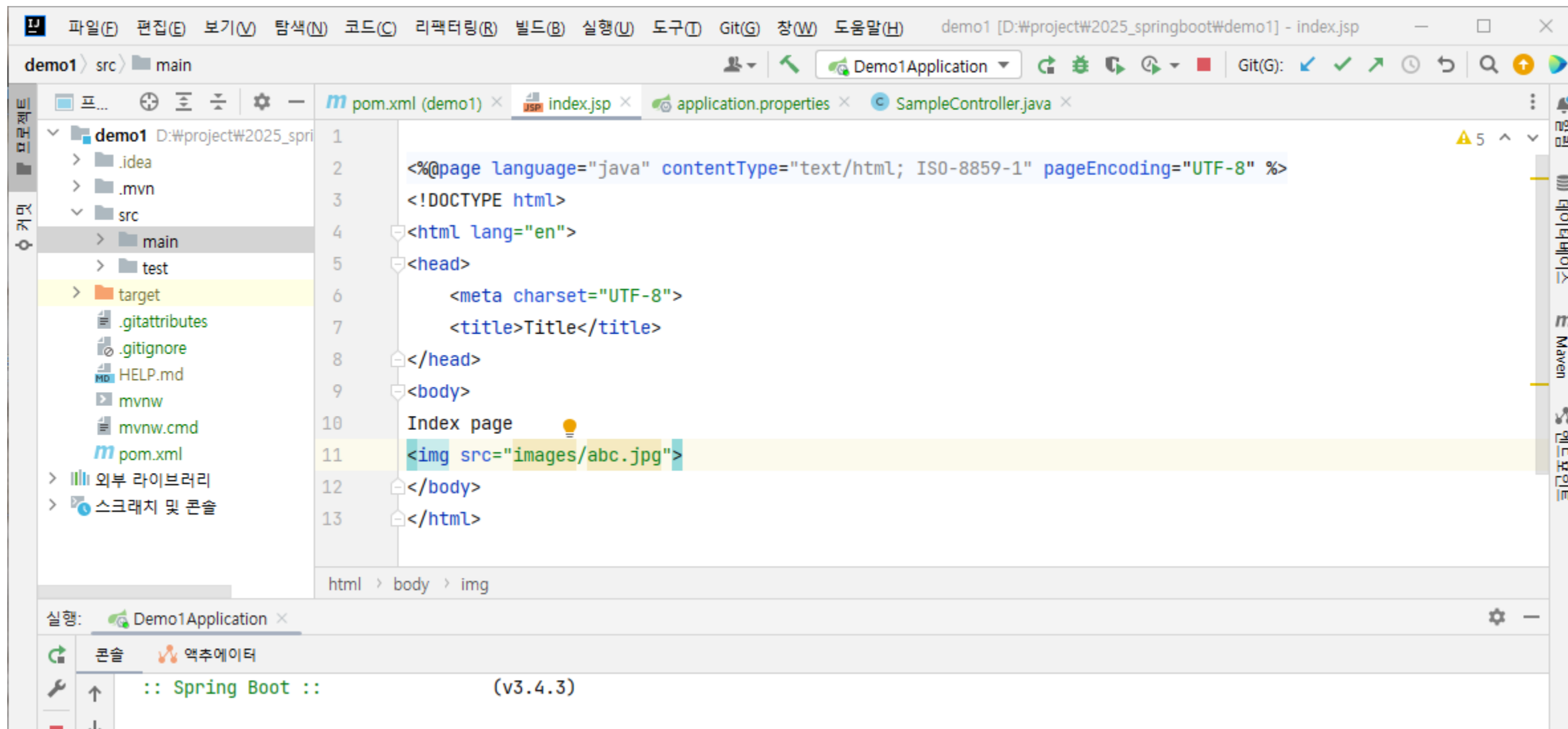
추가된 종속성:

- × Lombok
- × Spring Boot DevTools
- × Spring Web

? 이전(P) 생성(C) 취소

3. Spring Boot 프로젝트 생성

❖ 프로젝트 구조



The screenshot displays an IDE window for a Spring Boot project named 'demo1'. The left sidebar shows the project structure with folders like '.idea', '.mvn', 'src', 'main', 'test', and 'target'. The main editor area shows the 'index.jsp' file with the following content:

```
1 <%@page language="java" contentType="text/html; ISO-8859-1" pageEncoding="UTF-8" %>
2 <!--DOCTYPE html-->
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <title>Title</title>
7 </head>
8 <body>
9     Index page
10    
11 </body>
12 </html>
```

The bottom status bar indicates the application is running: '실행: Demo1Application x' and '콘솔 :: Spring Boot :: (v3.4.3)'.

3. Spring Boot 프로젝트 생성

❖ Jsp 파일 사용 설정: pom.xml 아래 내용을 추가

```
<!-- https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper -->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
</dependency>

<!-- https://mvnrepository.com/artifact/jakarta.servlet.jsp.jstl/jakarta.servlet.jsp.jstl-api -->
<dependency>
  <groupId>jakarta.servlet.jsp.jstl</groupId>
  <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
  <version>3.0.0</version>
</dependency>
```

3. Spring Boot 프로젝트 생성

❖ application.properties 작성

```
server.port=8081 # port 번호 설정
```

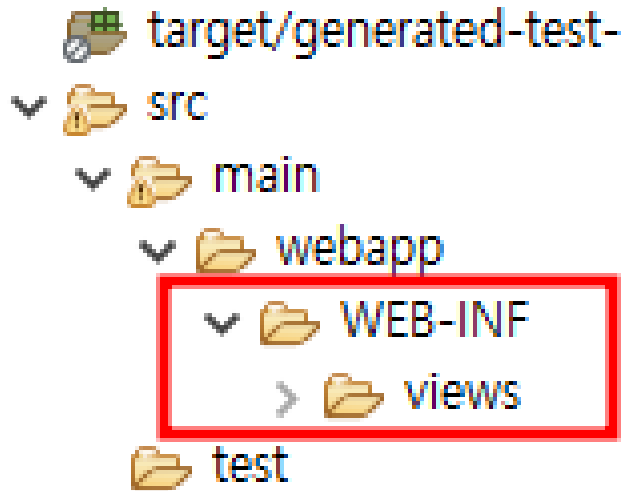
```
#view setting
```

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

```
spring.mvc.view.suffix=.jsp
```

3. Spring Boot 프로젝트 생성

❖ src/main/webapp에 WEB-INF/views 폴더 생성 후, index.jsp 파일 생성,

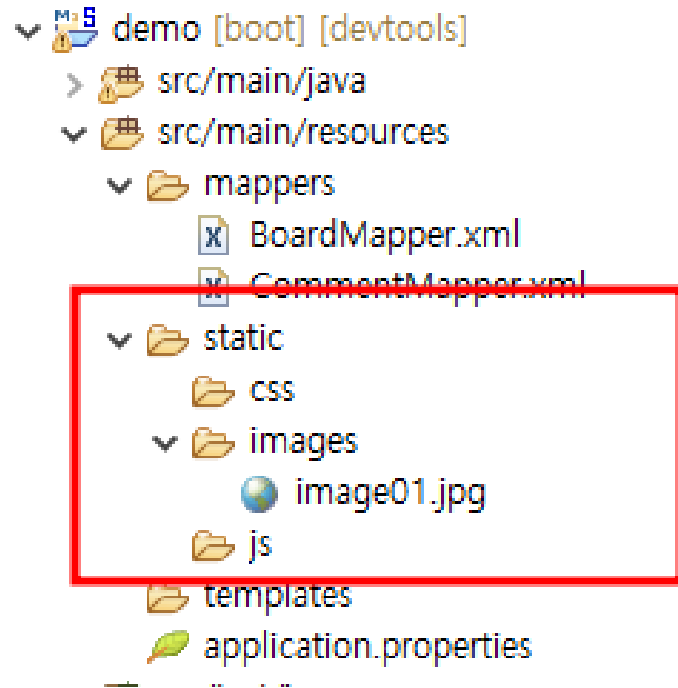


```
<%@page language="java" contentType="text/html; charset=utf-8"
pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<h1>Index page</h1>

</body>
</html>
```

3. Spring Boot 프로젝트 생성

- ❖ Mapper, Service, Controller, jsp View 스프링 프레임워크와 동일
- ❖ Static 소스 작성 위치 및 활용예

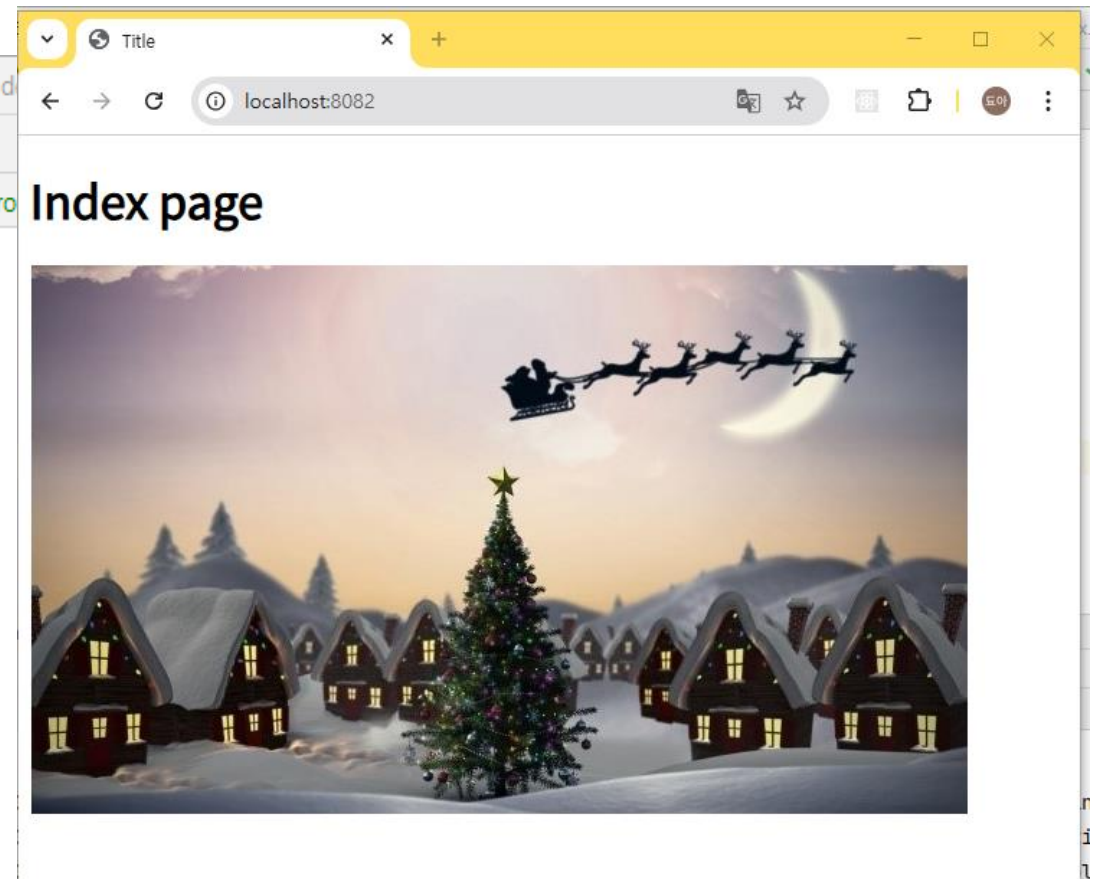
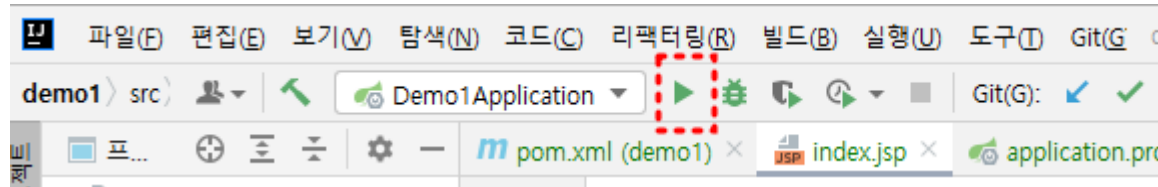


``

3. Spring Boot 프로젝트 생성

❖ 실행

- 도구에서 실행 버튼을 클릭하여 실행
- 브라우저를 열어서 url 창에 "localhost:8082" 입력하고 Enter 키를 누름



3. Spring Boot 프로젝트 생성

❖ 새프로젝트 만들기 실습

- 프로젝트명 : memo2
- 종속성 추가
 - Loombok
 - Spring Boot DevTools
 - Spring Web
- pom.xml
 - jsp와 jstl 라이브러리 추가
- jsp 파일 작성
 - src/main/webapp/WEB-INF/views/info.jsp를 작성
- controller 작성
 - controll 클래스 명 : InfoController
 - info.jsp 파일이 로드 되록록 getInto() 함수를 작성하라.

