



# 스프링 개발 환경 구축(1)

[Spring 환경]

# Spring 개발 환경 구축 절차

- ▶ 개발 도구(IDE) 설치
- ▶ JDK 설치
- ▶ 빌드 도구 설정 (Maven 또는 Gradle)
- ▶ 프로젝트 생성
- ▶ 디렉토리 구조 확인
- ▶ 서버 실행 및 확인
- ▶ 스프링 환경 구축

# Spring 개발 환경 구축 절차

## ▶ 개발 도구(IDE) 설치

도구	설명
IntelliJ IDEA	Spring Boot를 완벽 지원하는 대표 IDE (Community 또는 Ultimate)
Eclipse + Spring Tools	Eclipse 사용자라면 STS(Spring Tool Suite) 플러그인 설치 가능
VS Code	Java Extension Pack과 함께 사용 가능하지만 Spring에 특화 되진 않음

## ▶ JDK 설치

- ▶ Spring Boot 3.x 이상: Java 17 이상 필수
- ▶ 설치 후 환경 변수(JAVA\_HOME) 설정 필수



# Spring 개발 환경 구축 절차



## Spring 프로젝트 설정

### 빌드 도구 설정 (Maven 또는 Gradle)

- Spring 프로젝트는 라이브러리 의존성 관리가 필수이므로 빌드 도구 사용은 필수

도구	특징
Maven	XML 기반 설정 Spring 공식 문서에서 가장 자주 사용됨
Gradle	Groovy/Kotlin 기반 스크립트 빠르고 유연함

### 프로젝트 생성

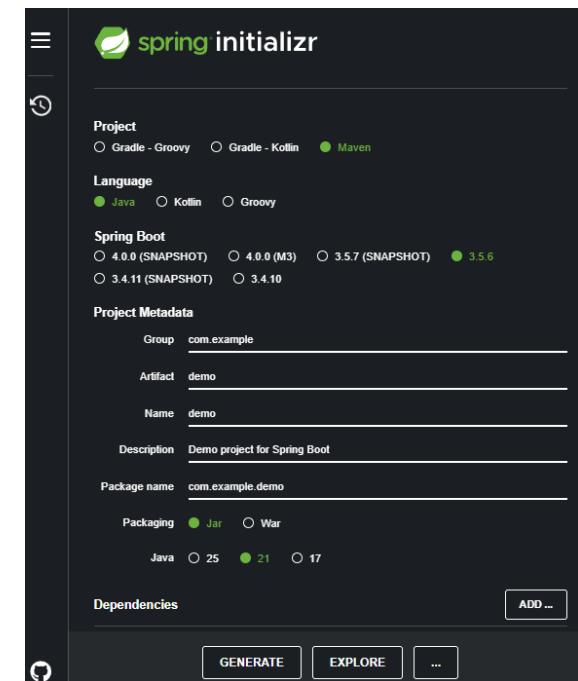
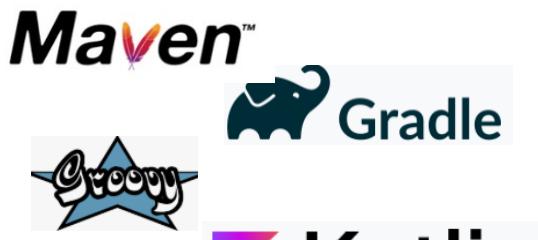
- Spring Initializr (공식) : <https://start.spring.io>

\* 선택 사항:

- Project: Maven or Gradle
- Language: Java
- Spring Boot version: 최신 안정 버전 (예: 3.2.x)
- Dependencies:

Spring Web  
Spring Boot DevTools  
Spring Data JPA  
Thymeleaf  
H2 Database 등

- IDE 내에서 생성 : IntelliJ 또는 STS에서 "Spring Initializr" 템플릿으로 바로 생성 가능



# Spring 개발 환경 구축 절차

## ▶ 디렉토리 구조 확인

```
src/  
└ main/  
    └ java/      ← Java 소스 코드  
    └ resources/ ← 설정 파일, 템플릿, 정적 리소스  
        └ application.properties (또는 .yml)  
└ test/        ← 테스트 코드
```

- ▶ src/main/java: Java 소스 코드
- ▶ src/main/resources: 설정 파일 (application.properties 등)
- ▶ src/test/java: 테스트 코드
- ▶ pom.xml: 의존성 관리 파일
- ▶ DemoApplication.java: 실행 진입점 (main 메서드)

## ▶ 설정 파일 구성

- ▶ **application.properties** 또는 **application.yml**을 통해 기본 설정

### Properties

```
server.port=8080  
spring.datasource.url=jdbc:h2:mem:testdb  
spring.jpa.hibernate.ddl-auto=update
```

# Spring 개발 환경 구축 절차

- ▶ 서버 실행 및 확인
  - ▶ DemoApplication.java 또는 MainApplication.java 내 main() 메서드 실행
  - ▶ 내장 Tomcat이 실행되어 <http://localhost:8080>에서 서버 구동 확인
- ▶ 데이터베이스 설정

DB 종류	예
내장	H2 (개발/테스트용)
외부	MySQL, PostgreSQL, Oracle 등 설정 가능

## Properties

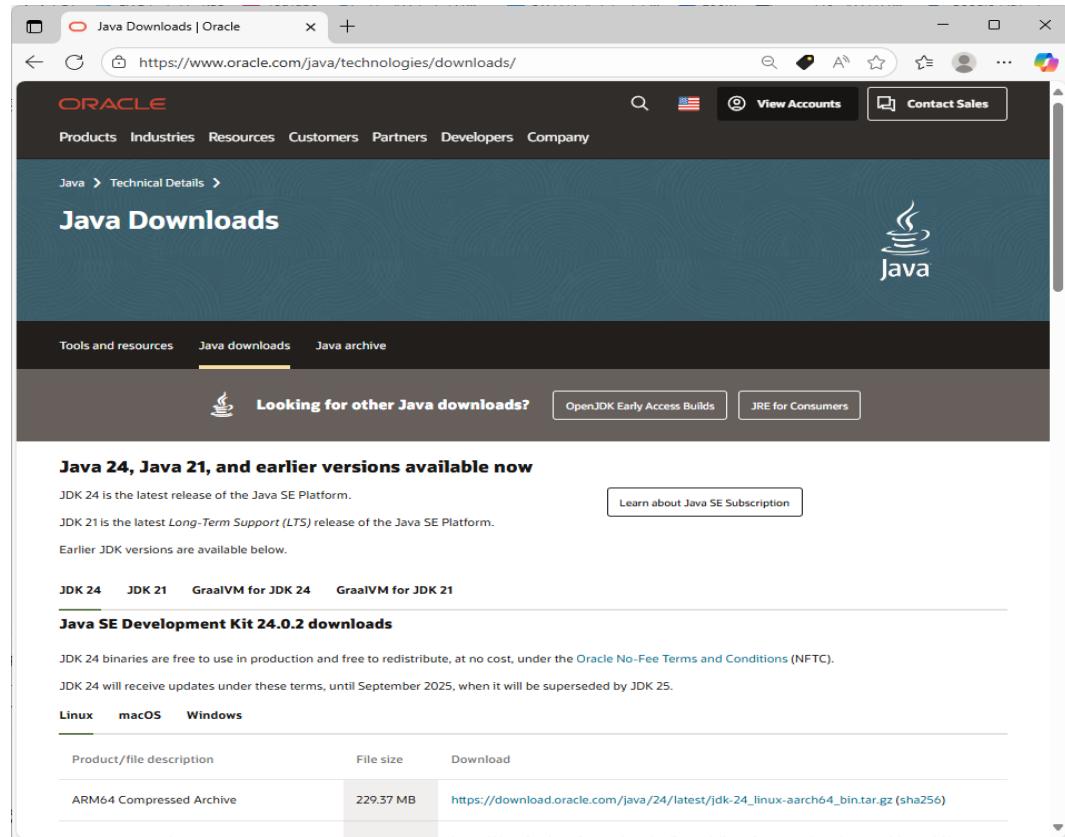
```
spring.datasource.url=jdbc:mysql://localhost:3306/testdb  
spring.datasource.username=root  
spring.datasource.password=1234  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true
```

# 스프링 개발환경 구축

- ▶ Java 설치 (<https://www.oracle.com/index.html>)
  - ▶ JDK 17 (JDK17) 이상 설치
- ▶ STS 설치 (<https://spring.io/tools>)
  - ▶ Eclipse + Spring Tools
- ▶ Tomcat 설치 (STS설치 시 생략 가능)
  - ▶ Tomcat 9.x 이상

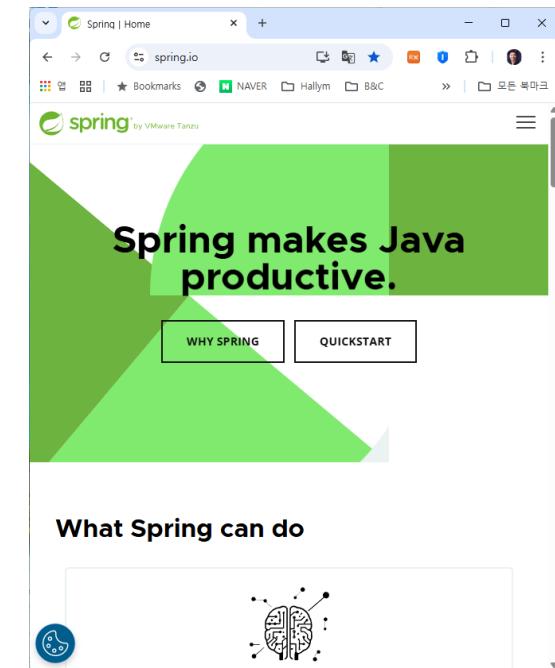
# Java 설치

- ▶ <https://www.oracle.com/java/technologies/downloads/>
- ▶ 오라클 회원가입 (계정 만들기)
- ▶ JDK 17 이상 설치
- ▶ 본인의 OS에 맞는 JDK 설치



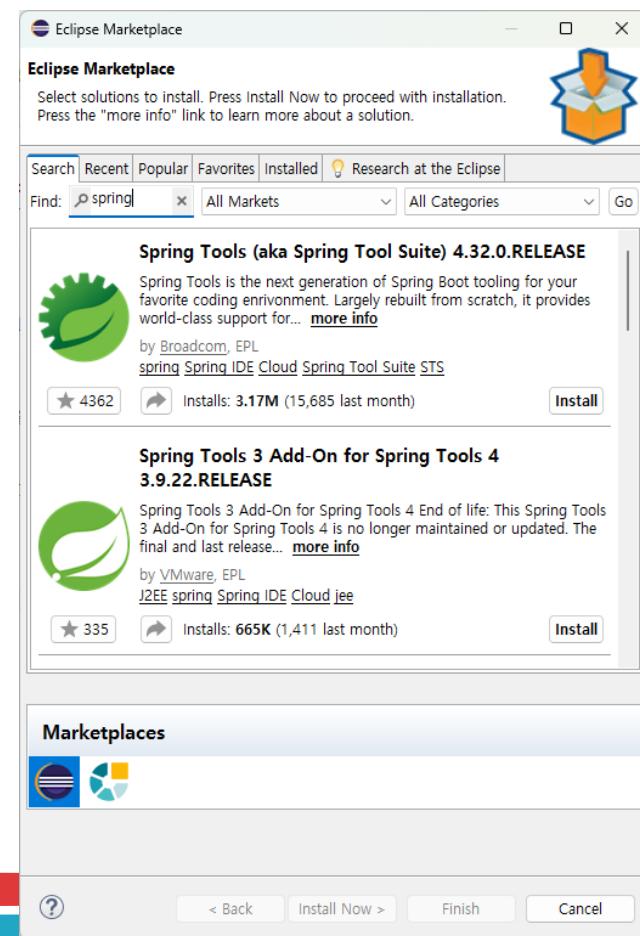
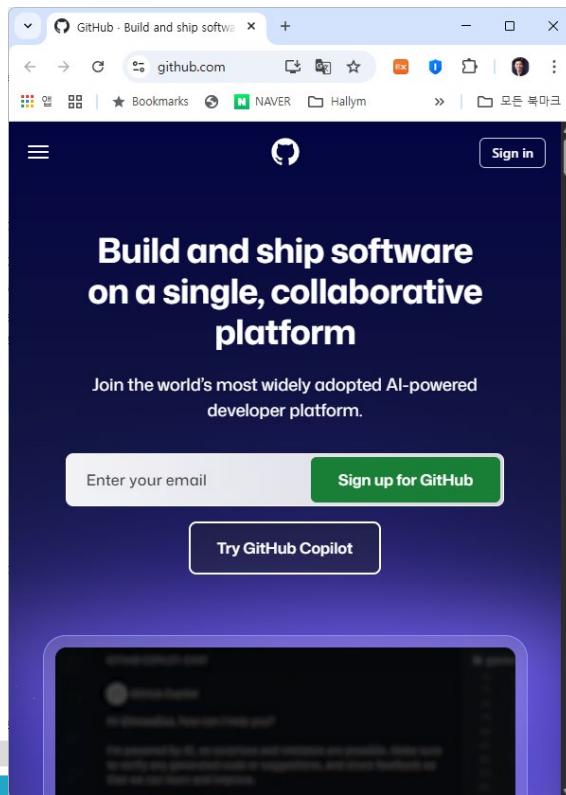
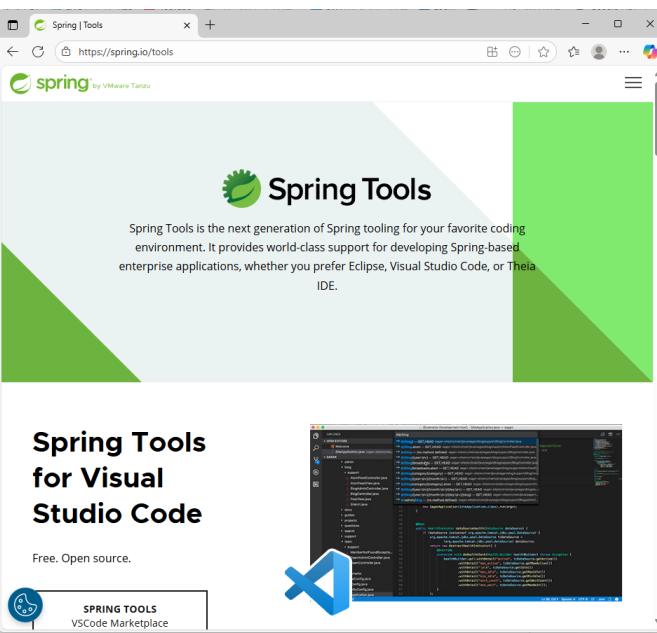
# STS(Spring Tool Suit )

- ▶ 스프링을 이용한 프로그램 개발을 쉽게 진행할 수 있도록 고안된 프로젝트
- ▶ Eclipse + Spring Plugin
- ▶ IDE 툴이란 통합 개발환경으로 코딩, 디버그, 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경
- ▶ 스프링을 개발하기 위해서는 개발에 필요한 Spring plugin 설치
- ▶ Eclipse 이외의 tool에서도 가능함
- ▶ 본인의 OS에 맞는 IDE 버전을 설치

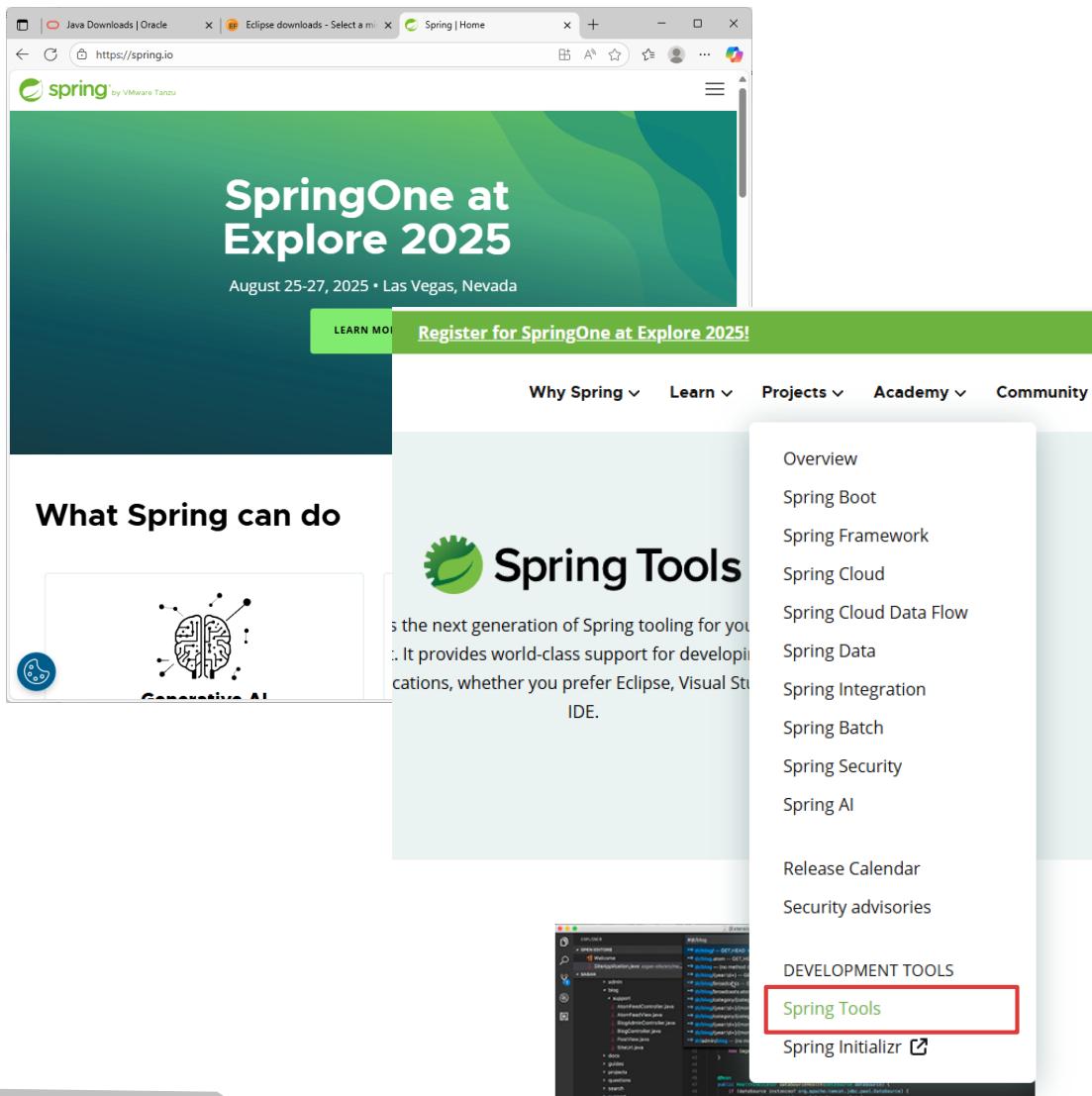


# STS 설치

- ▶ STS 공식 사이트에서 다운로드해서 사용하는 방법
- ▶ 이클립스에 STS 플러그인을 설치해서 사용하는 방법
- ▶ 스프링 깃허브에서 단독으로 내려 받아 사용하는 방법



# STS 공식 사이트(<https://spring.io>)



1. 사이트에 접속
2. 본인의 OS에 맞는 스프링 다운로드
3. 압축해제
4. SpringToolSuite4.exe 응용프로그램 실행

## Spring Tools for Eclipse

Free. Open source.

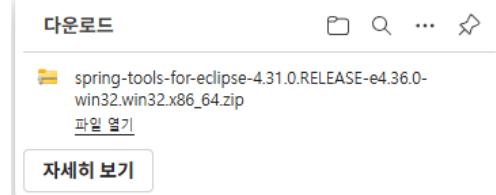
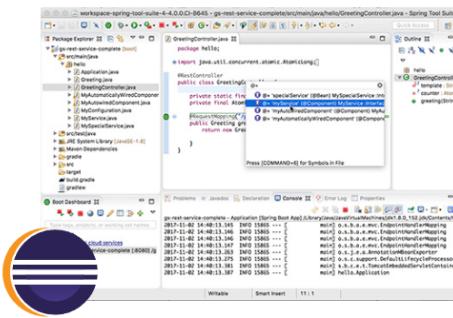
4.31.0 - LINUX X86\_64

4.31.0 - LINUX ARM\_64

4.31.0 - MACOS X86\_64

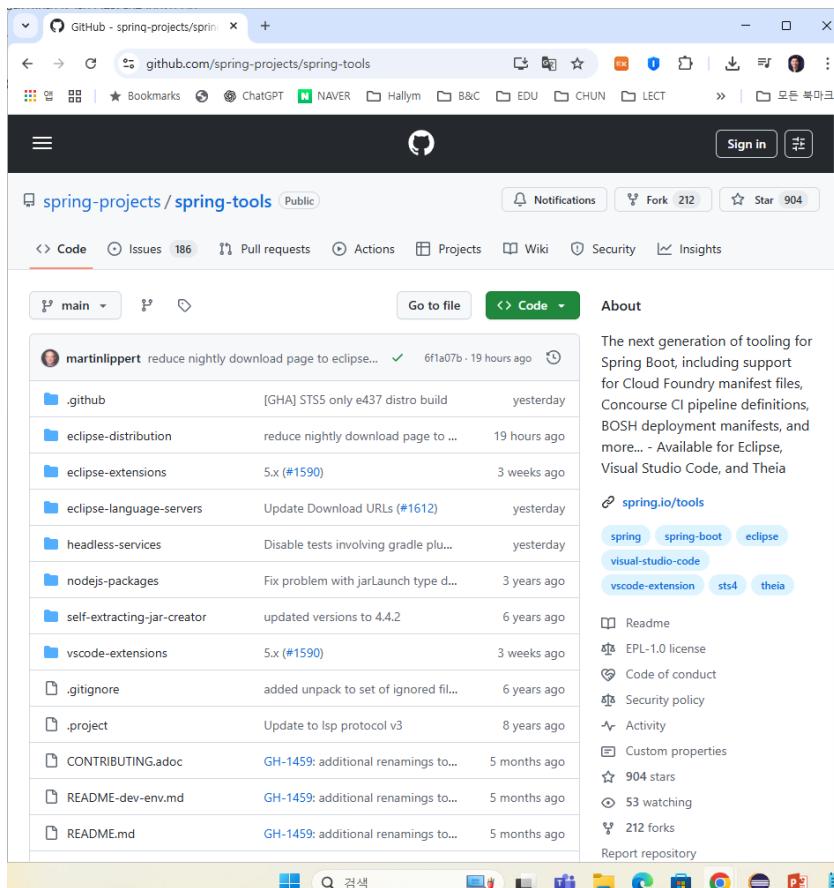
4.31.0 - MACOS ARM\_64

4.31.0 - WINDOWS X86\_64

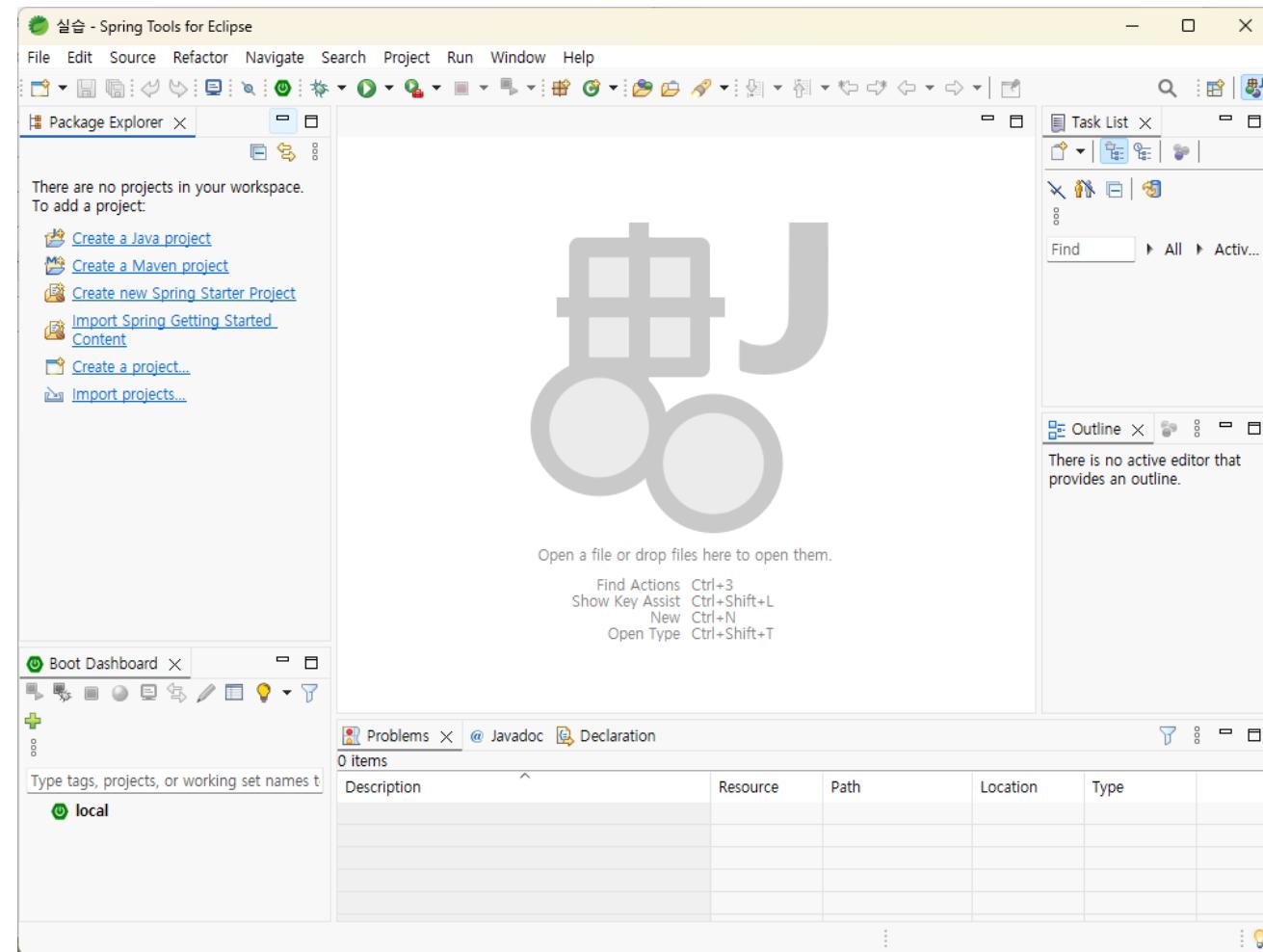
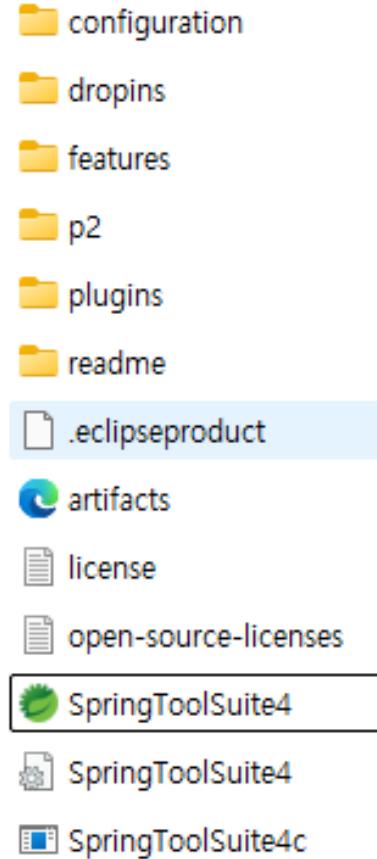


# STS 깃허브 공식 사이트

▶ <https://github.com/spring-projects/spring-tools>



# 압축 해제 후 STS 실행



# 전자정부 표준 프레임워크

▶ <https://www.egovframe.go.kr/home/main.do>

1) 전자정부 사이트 접속  
보안 패지 안내

2) 다운 받은 후 압축해제  
다운로드

3.1 개발자용 개발환경 for Windows 64bit (Implementation Tool) Version 4.3.1  
작성자 관리자 작성일 2025-05-12 조회수 5064  
첨부파일 eGovFrameDev-4.3.1-64bit.zip [1,522,817,612 Byte]

workspace-egov - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Exp... X Out... X

There are no projects in your workspace.  
To add a project:

- Create a Maven project
- eGovFrame Device API Web Project
- eGovFrame Mobile Project
- eGovFrame Mobile Template Project
- Create a Java EE EAR project
- Create a Dynamic Web project
- Create an EJB project
- Create a Connector project
- Create a Java EE application client project
- Create a deployable web project
- Create new Spring Starter Project

3) 이클립스 실행 화면

Open a file or drop files here to open them.

Find Actions Ctrl+3  
Show Key Assist Ctrl+Shift+L  
New Ctrl+N

Problems X Servers Terminal Data Source Explorer Properties

# 인코딩과 글꼴 변경

The image shows two screenshots of the Eclipse IDE preferences interface.

**Left Screenshot (Workspace Preferences):**

- Text file encoding:** The "Other" dropdown is set to **UTF-8**, highlighted with a red box.
- New text file line delimiter:** The "Default (Windows)" radio button is selected.

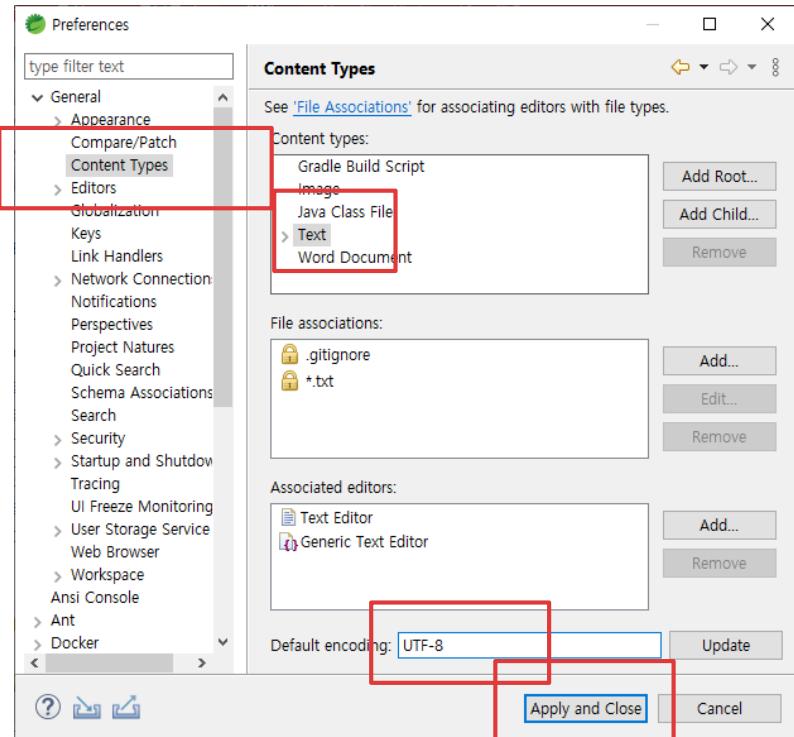
**Right Screenshot (CSS Files Preferences):**

- Creating files:** The "Encoding" dropdown is set to **ISO 10646/Unicode(UTF-8)**, highlighted with a red box.
- IANA:** The value is **UTF-8**.

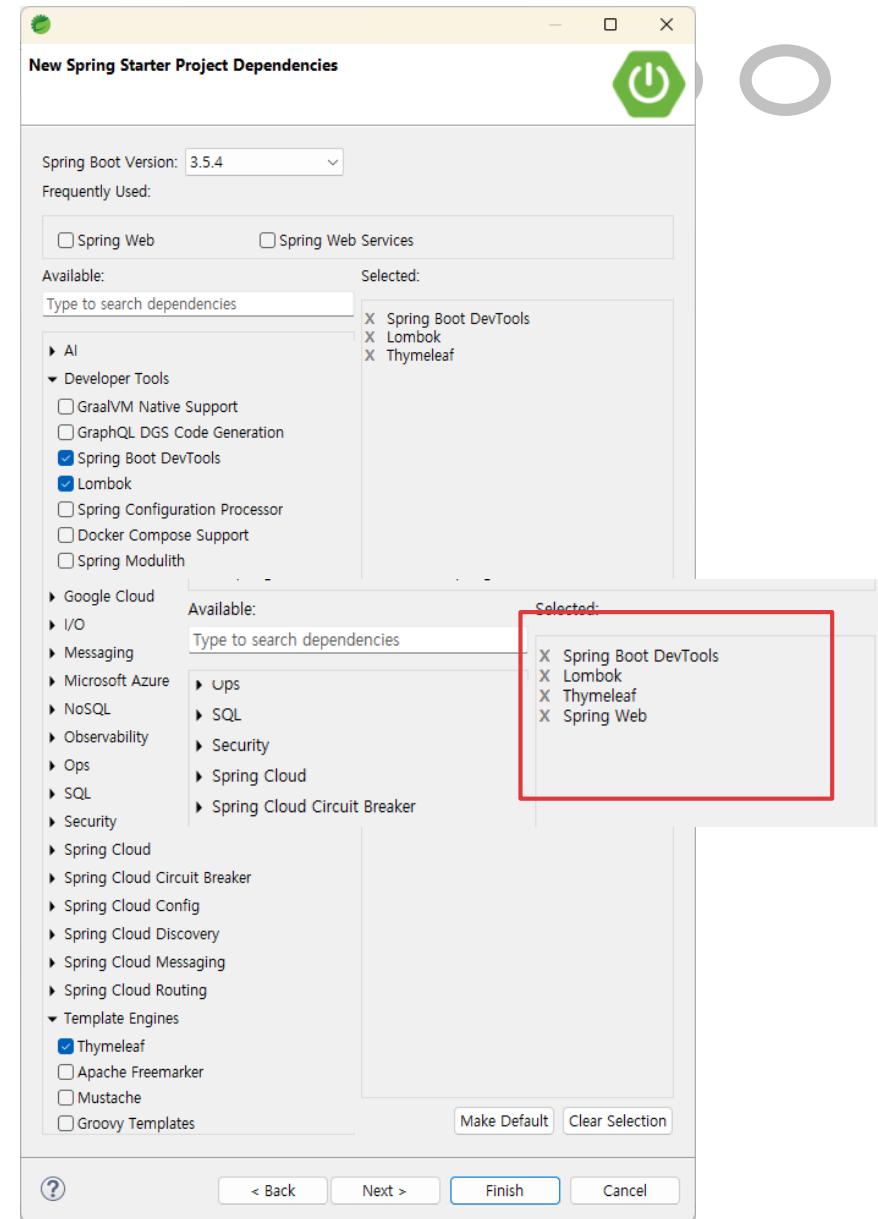
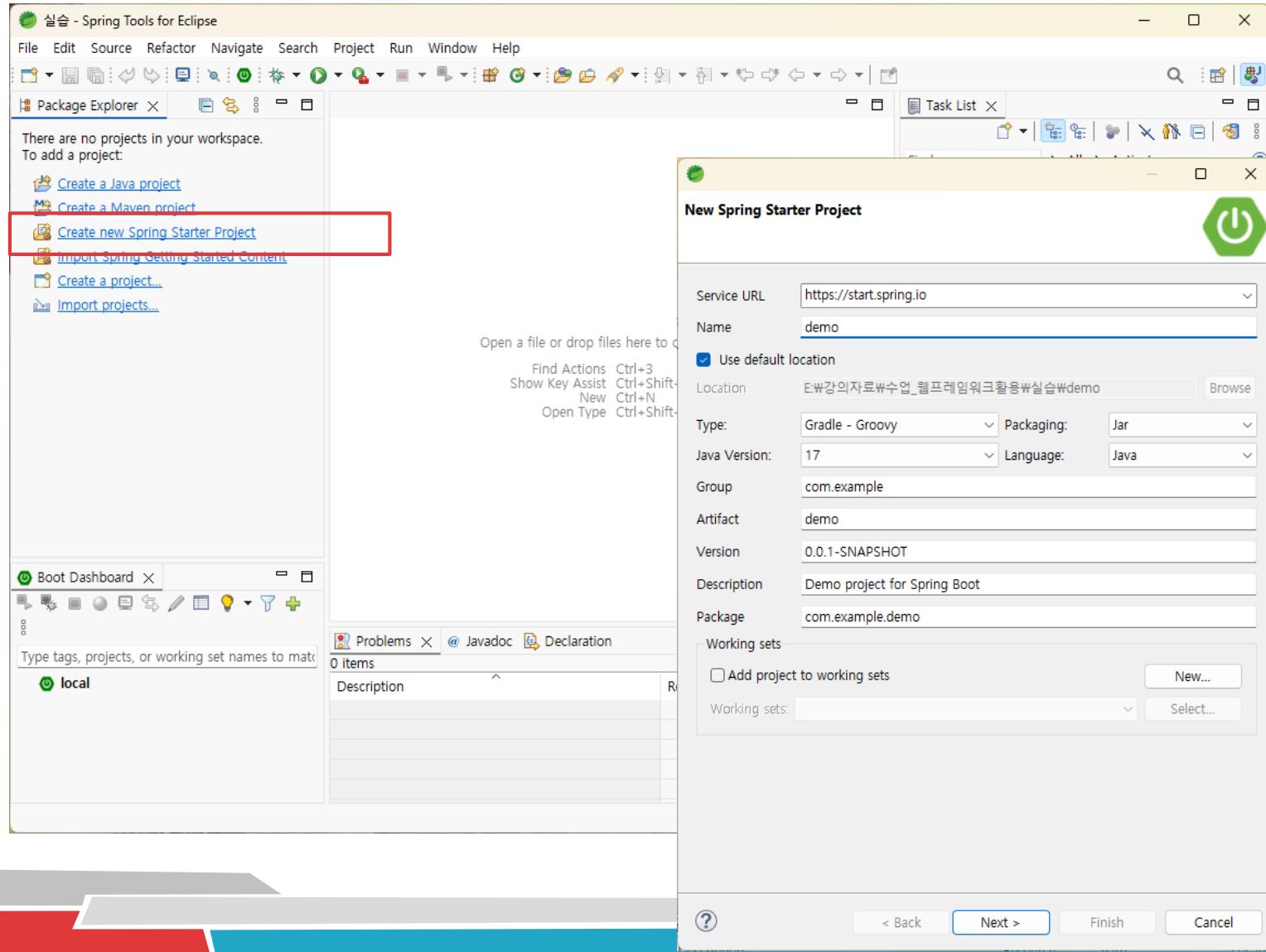
**Bottom Text:** **HTML, CSS, JSP 모두 변경** (Change all for HTML, CSS, JSP)

# ContentTypes Text 글꼴 변경

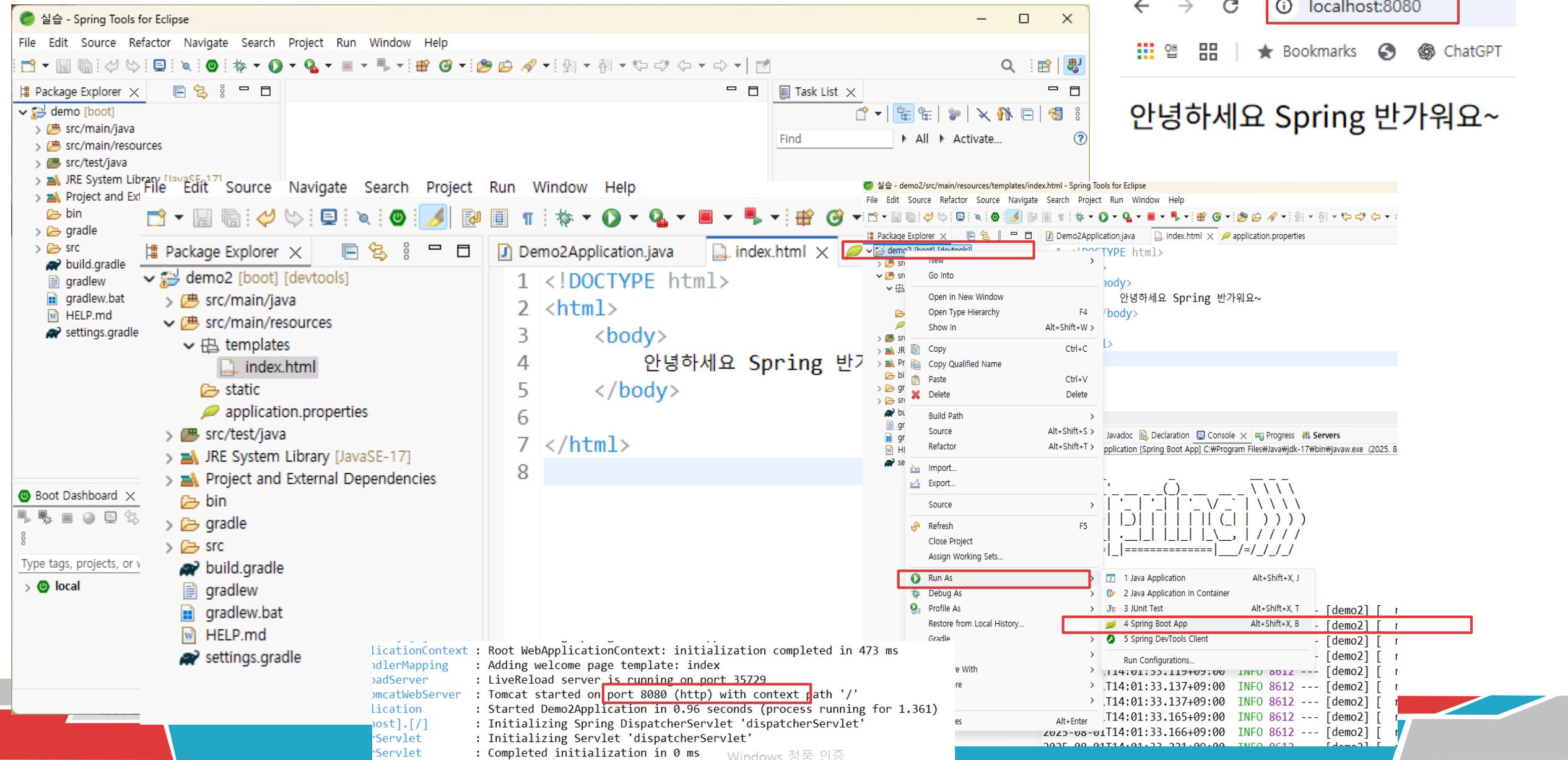
## 1. STS 실행 후 한글 코드 통일



# Spring 시작



# Spring 실습 1)



# Spring 실습2)

The screenshot shows the Spring Tools for Eclipse interface. On the left, the Package Explorer displays the project structure for 'demo2'. It includes files like Demo2Application.java, HelloController.java, and ServletInitializer.java under 'src/main/java/com.example.demo'. A 'hello.html' file is shown in the 'templates' folder under 'src/main/resources'. On the right, the code editors show 'HelloController.java' and 'hello.html'. The 'HelloController.java' code defines a controller with a single method that returns a Thymeleaf template. The 'hello.html' template uses Thymeleaf syntax to display a greeting message. To the right of the editors is a browser window showing the result of the application running at 'localhost:8080/do', which displays the text '안녕하세요 Spring~~'.

```
File Edit Source Refactor Navigate Search Project Run Window Help  
Package Explorer X  
demo2 [boot] [devtools]  
src/main/java  
com.example.demo  
Demo2Application.java  
HelloController.java  
ServletInitializer.java  
src/main/resources  
templates  
hello.html  
static  
index.html  
application.properties  
src/test/java  
JRE System Library [JavaSE-17]  
Project and External Dependencies  
bin  
gradle  
src  
build.gradle  
gradlew  
gradlew.bat  
HELP.md  
settings.gradle
```

```
hello.html X  
<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<body>  
<p th:text=" '안녕하세요 ' + ${data}"> 안녕하세요. 손님</p>  
</body>  
</html>
```

```
HelloController.java X  
package com.example.demo;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
@Controller  
public class HelloController {  
    @GetMapping("do") // URL주소상의 위치  
    public String hello(Model model) {  
        model.addAttribute("data", "Spring~~");  
        return "hello"; // templates/hello.html 호출  
    }  
}
```

```
localhost:8080/do  
안녕하세요 Spring~~
```