



# 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 되지 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

## ▶ 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

Maven™

Gradle



Kotlin

## ▶ 프로젝트 생성

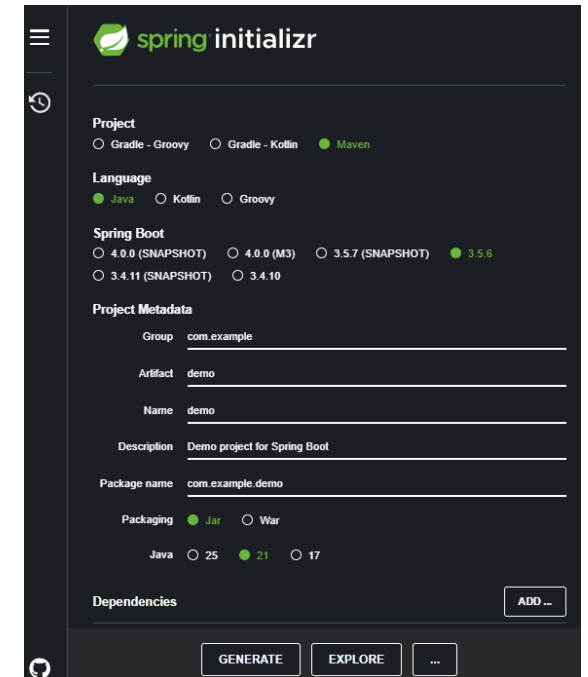
- ▶ Spring Initializr (공식): <https://start.spring.io>

\* 선택 사항:

- 1) Project: Maven or Gradle
- 2) Language: Java
- 3) Spring Boot version: 최신 안정 버전 (예: 3.2.x)
- 4) 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:8080에서 서버 구동 확인

## ▶ 데이터베이스 설정

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

### Properties

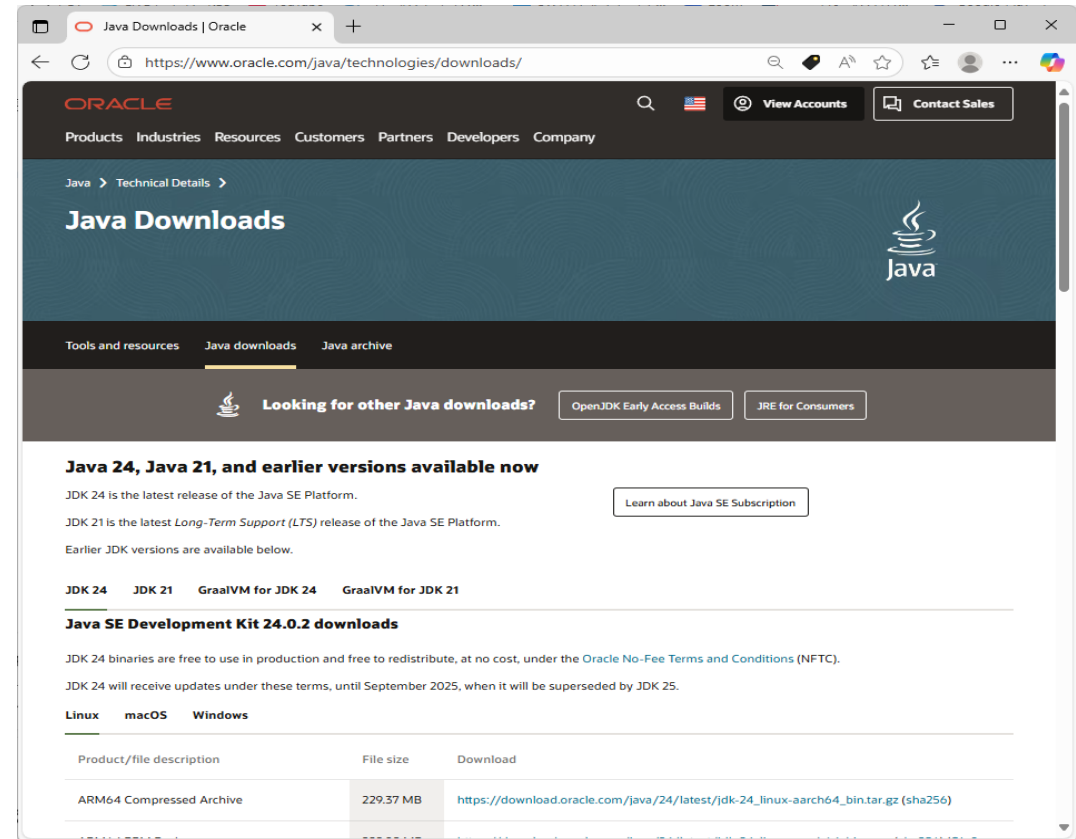
```
spring.datasource.url=jdbc:mysql://localhost:3306/testdb
spring.datasource.username=root
spring.datasource.password=1234
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

# 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

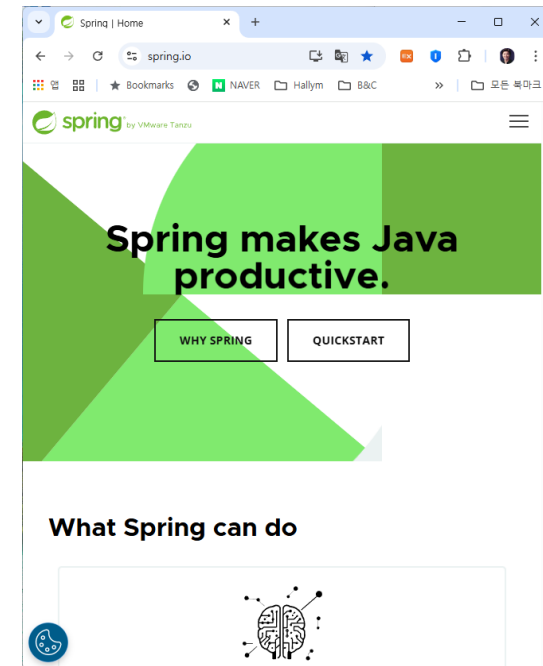
- ▶ <https://www.oracle.com/java/technologies/downloads/>
- ▶ 오라클 회원가입 (계정 만들기)
- ▶ JDK 17 이상 설치
- ▶ 본인의 OS에 맞는 JDK 설치





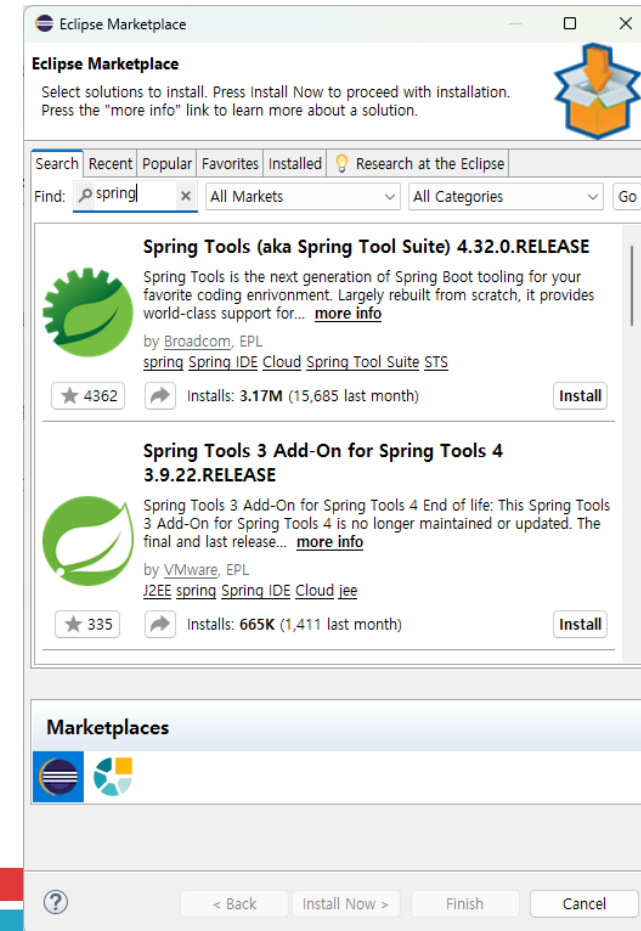
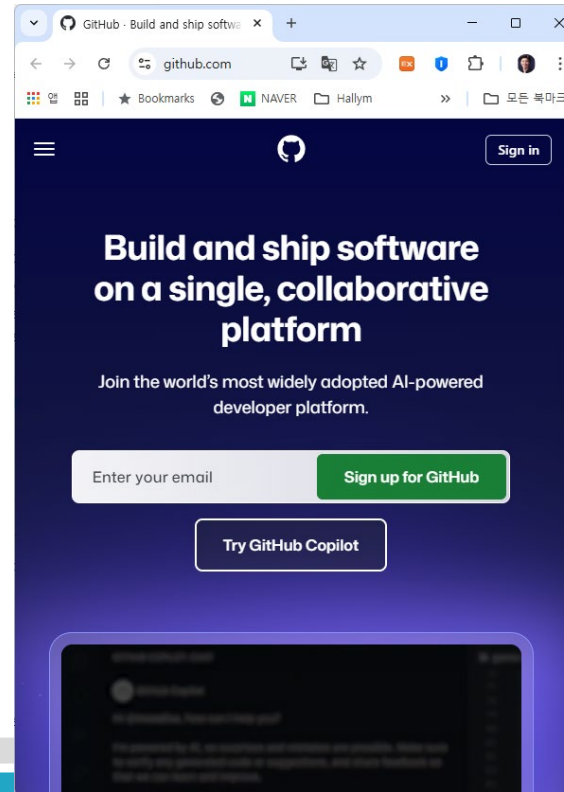
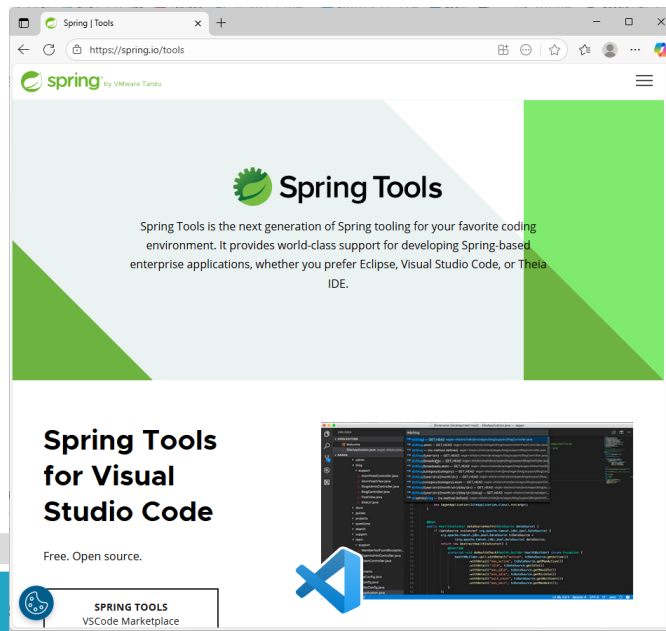
# STS(Spring Tool Suit )

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

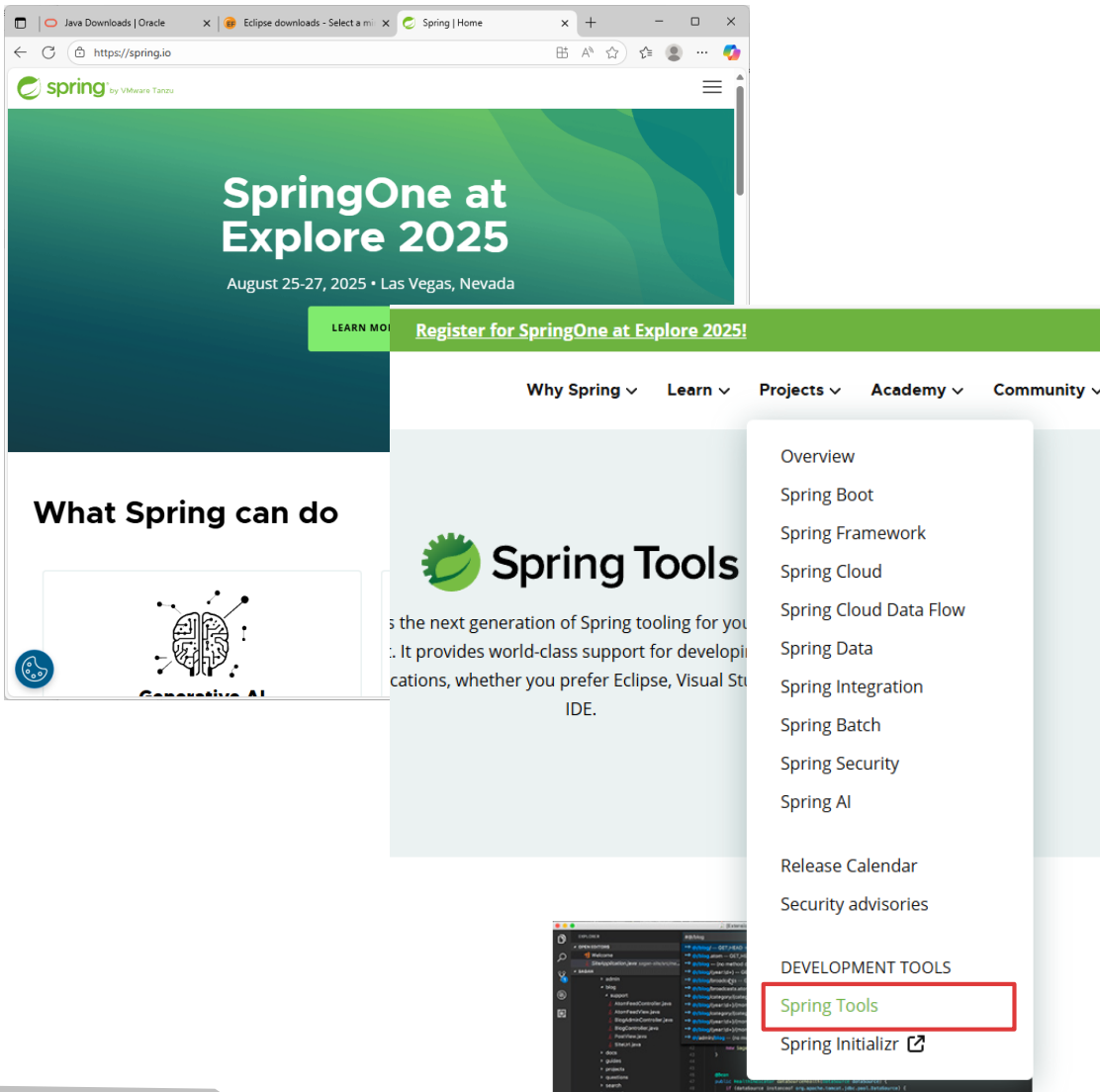


# STS 설치

- ▶ 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.

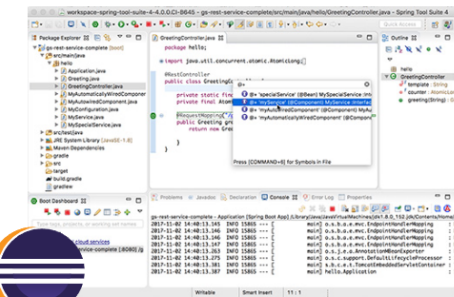
4.31.0 - LINUX X86\_64

4.31.0 - LINUX ARM\_64

4.31.0 - MACOS X86\_64

4.31.0 - MACOS ARM\_64

4.31.0 - WINDOWS X86\_64



## 다운로드

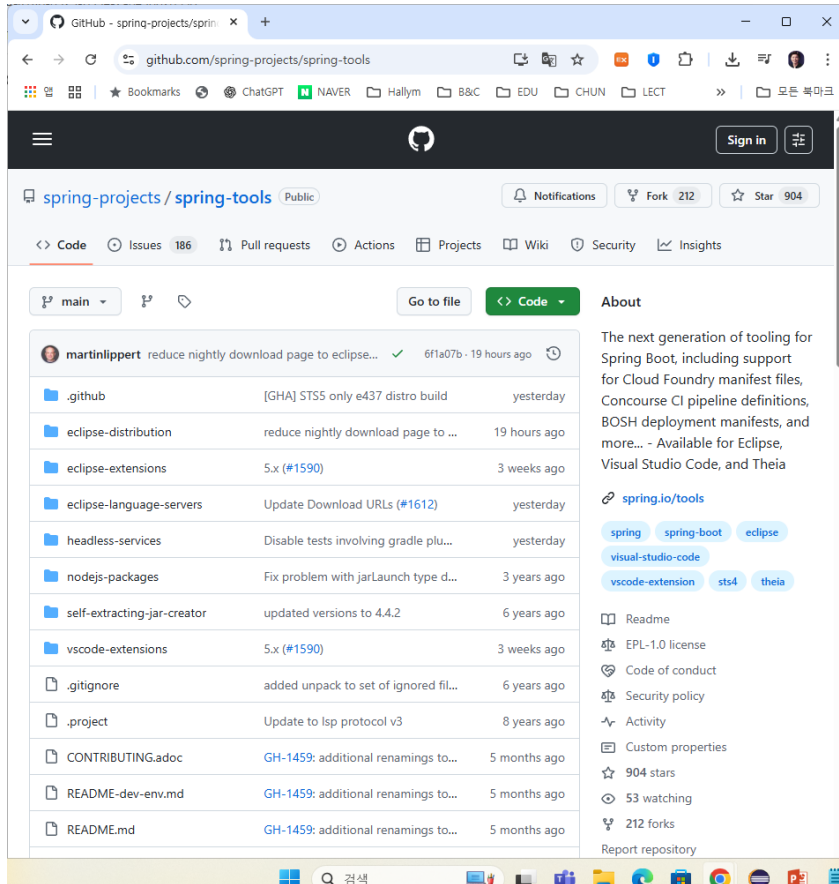
spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x86\_64.zip

파일 열기

자세히 보기

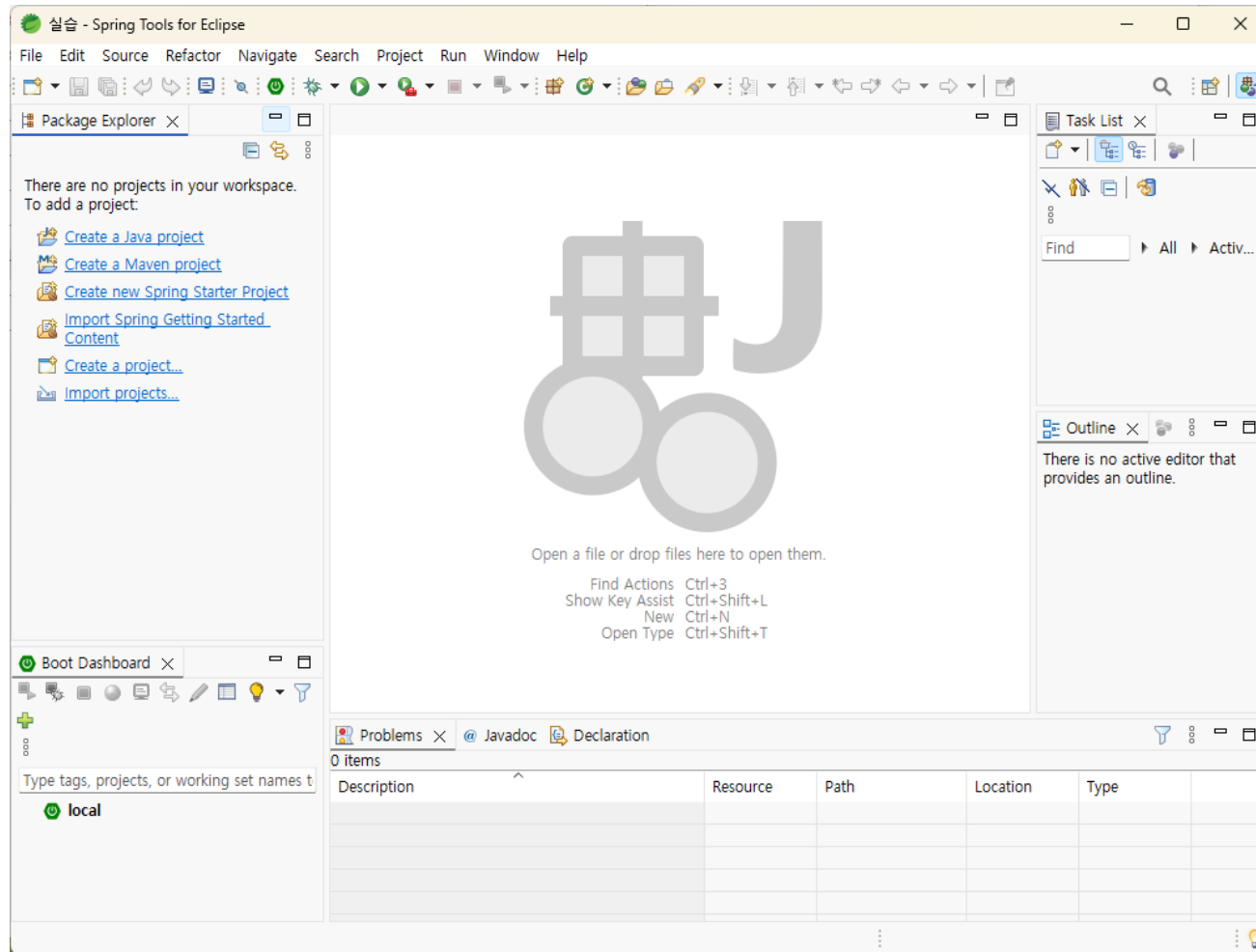
# STS 깃허브 공식 사이트

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



# 압축 해제 후 STS 실행

- configuration
- dropins
- features
- p2
- plugins
- readme
- .eclipseproduct
- artifacts
- license
- open-source-licenses
- SpringToolSuite4**
- SpringToolSuite4
- SpringToolSuite4c



# 전자정부 표준 프레임워크

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



## 2) 다운 받은 후 압축해제

### 다운로드

실용환경 | 개발환경 | 운영환경 | 공통컴포넌트 | 모바일 실용환경 | 모바일 공통컴포넌트

Release Notes | **4.x 다운로드** | 3.x 다운로드 | 2.x 다운로드 | 1.0 다운로드

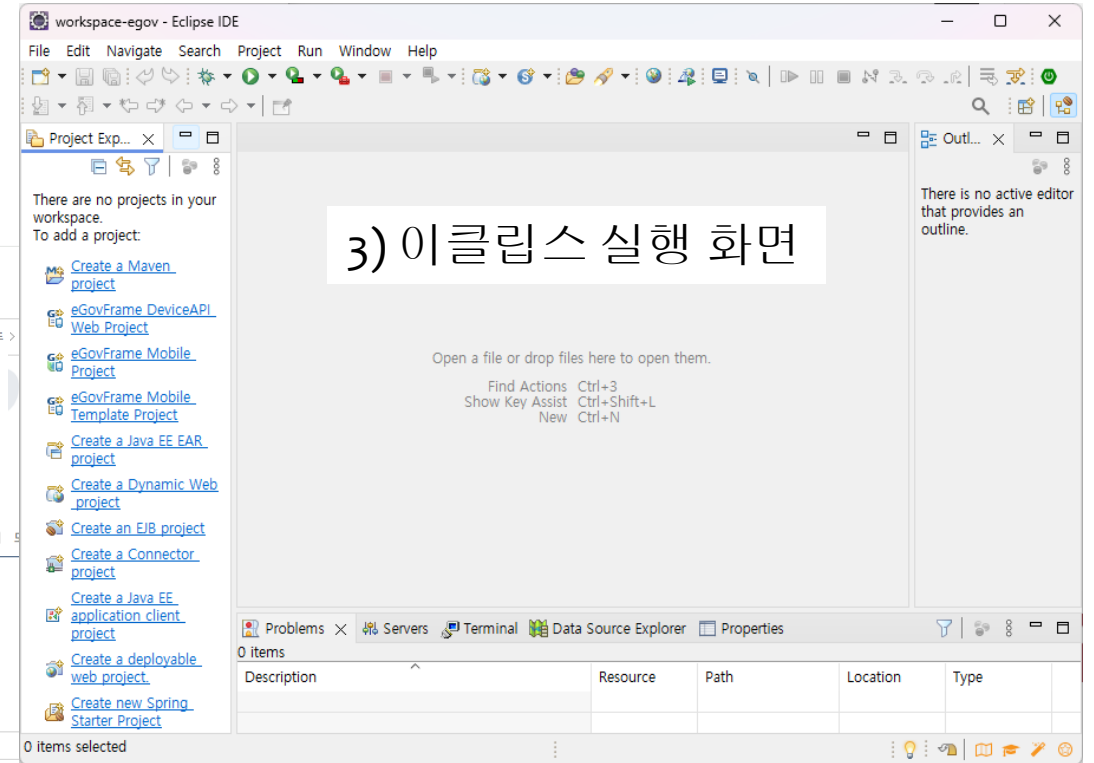
### 개발환경 - 4.x 다운로드

4.x 다운로드

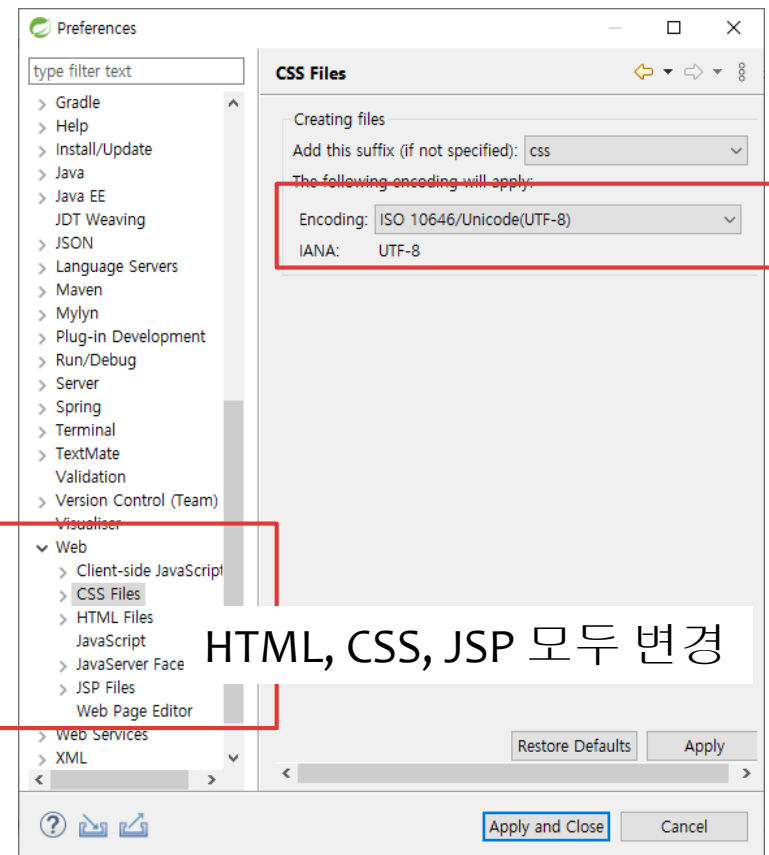
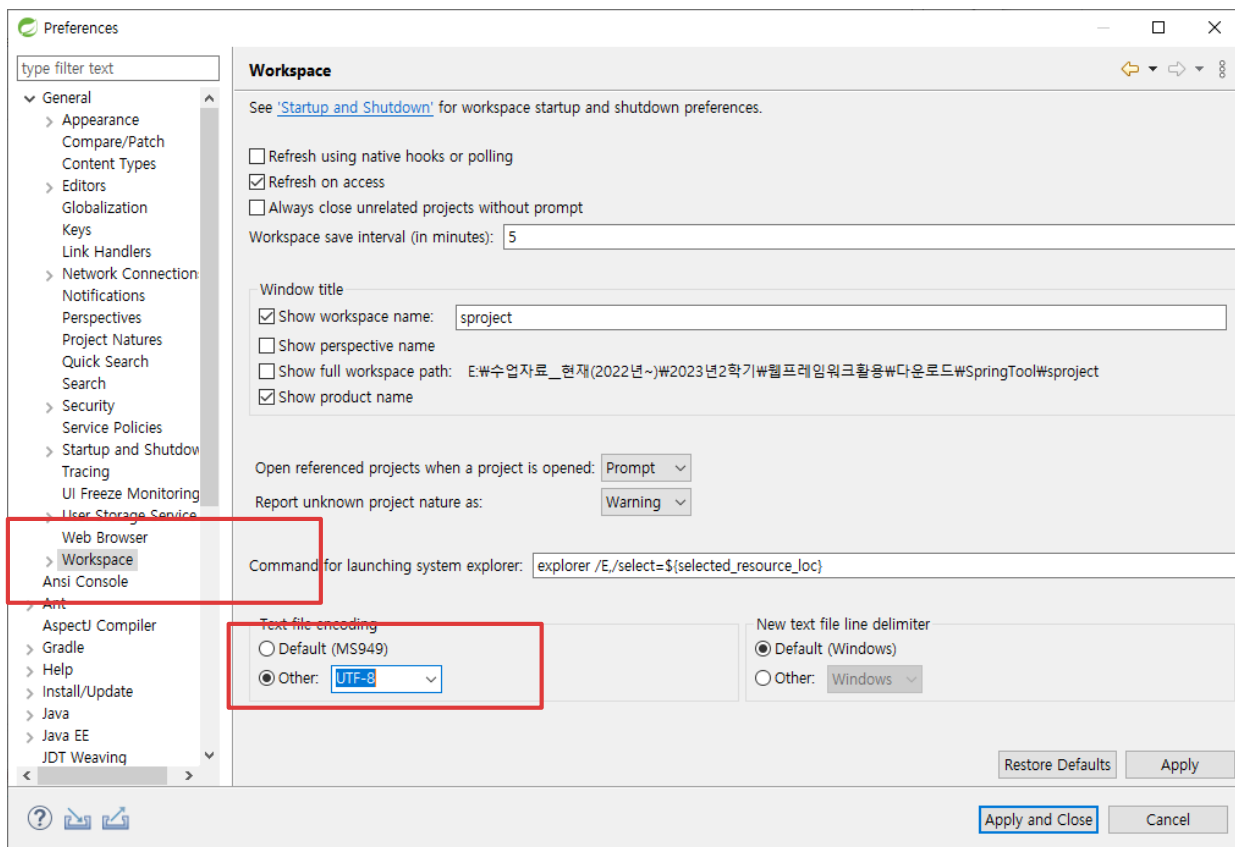
버전	개발자용 개발환경 for Windows 64bit (Implementation Tool) Version 4.3.1	릴리즈 일자	2025-05-12		
작성자	관리자	작성일	2025-05-12	조회수	5064
첨부파일	eGovFrameDev-4.3.1-64bit.zip [1,522,817,612 Byte]				

전자정부 표준프레임워크 라이선스 (바로그기)  
전자정부 표준프레임워크 활용의 안정성 보장을 위해 위험성을 지속적으로 모니터링하고 있으나, 오픈소스의 특성상 문제가 발생할 수 있습니다.  
전자정부 표준프레임워크는 Apache 2.0 라이선스를 따르고 있는 오픈소스 프로젝트입니다.  
Apache 2.0 라이선스에 따라 표준프레임워크를 활용하여 발생된 업무중단, 컴퓨터 고장 또는 오동작으로 인한 손해 등에 대해서 책임이 없습니다.  
단, 표준프레임워크 내에서 사용된 외부 오픈소스의 경우 원 오픈소스의 라이선스 정책을 유지합니다.

[다운로드](#)

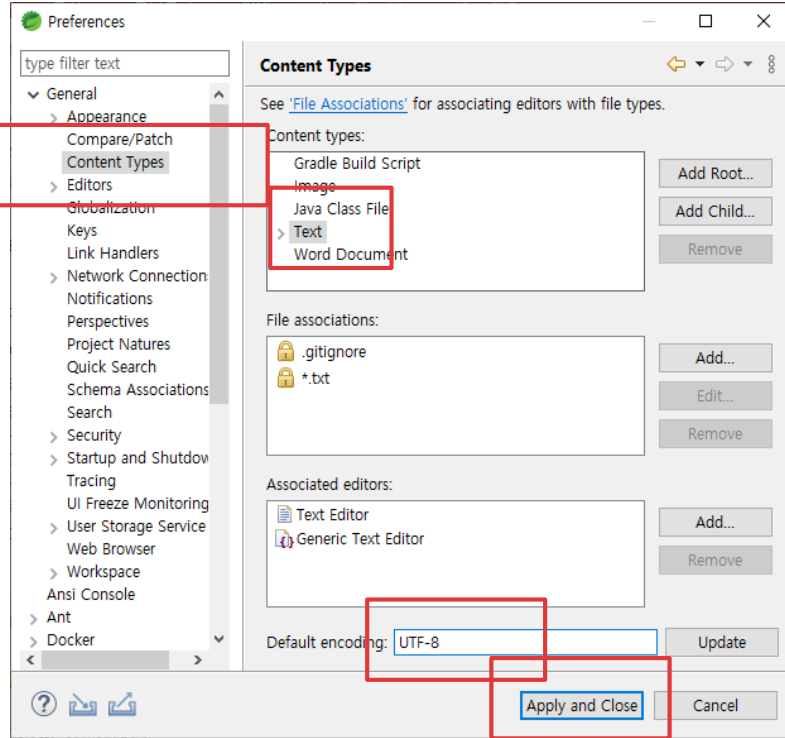


# 인코딩과 글꼴 변경



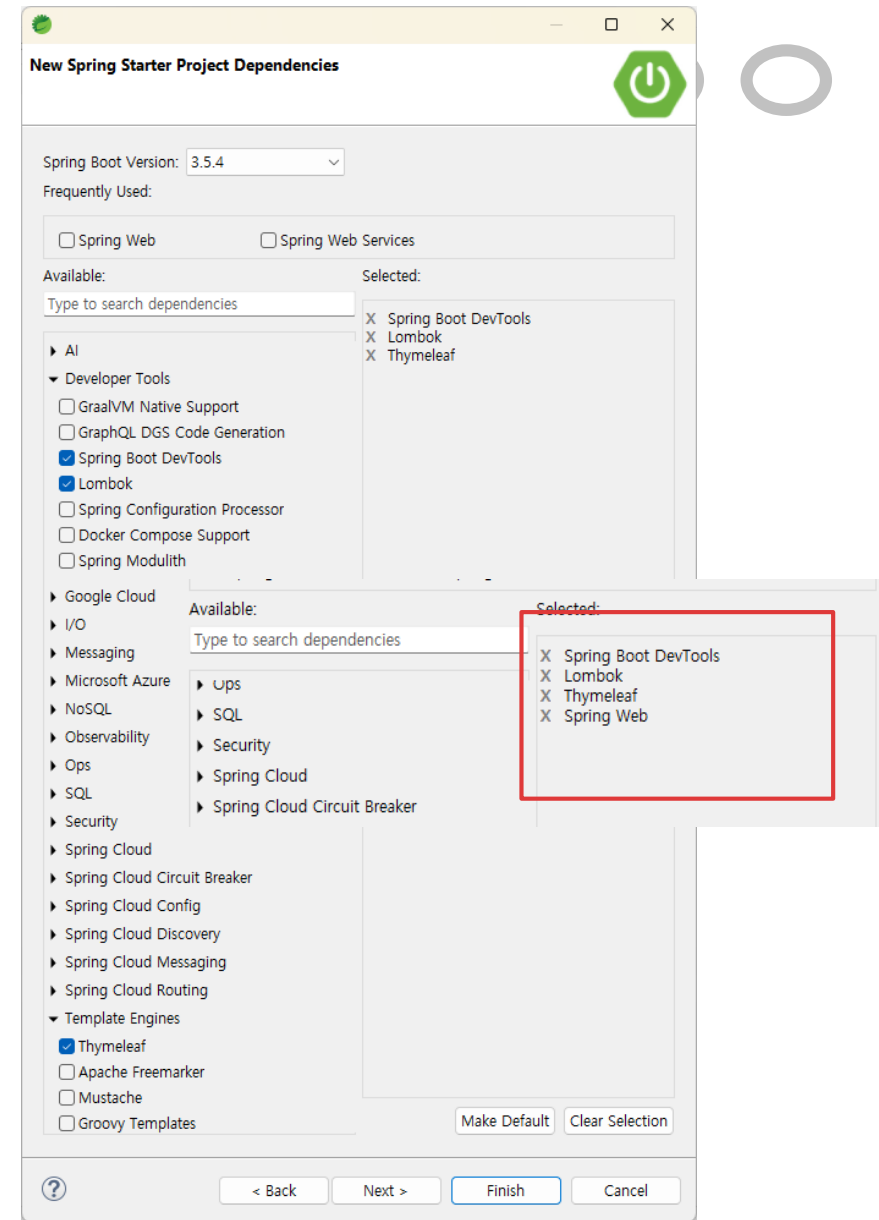
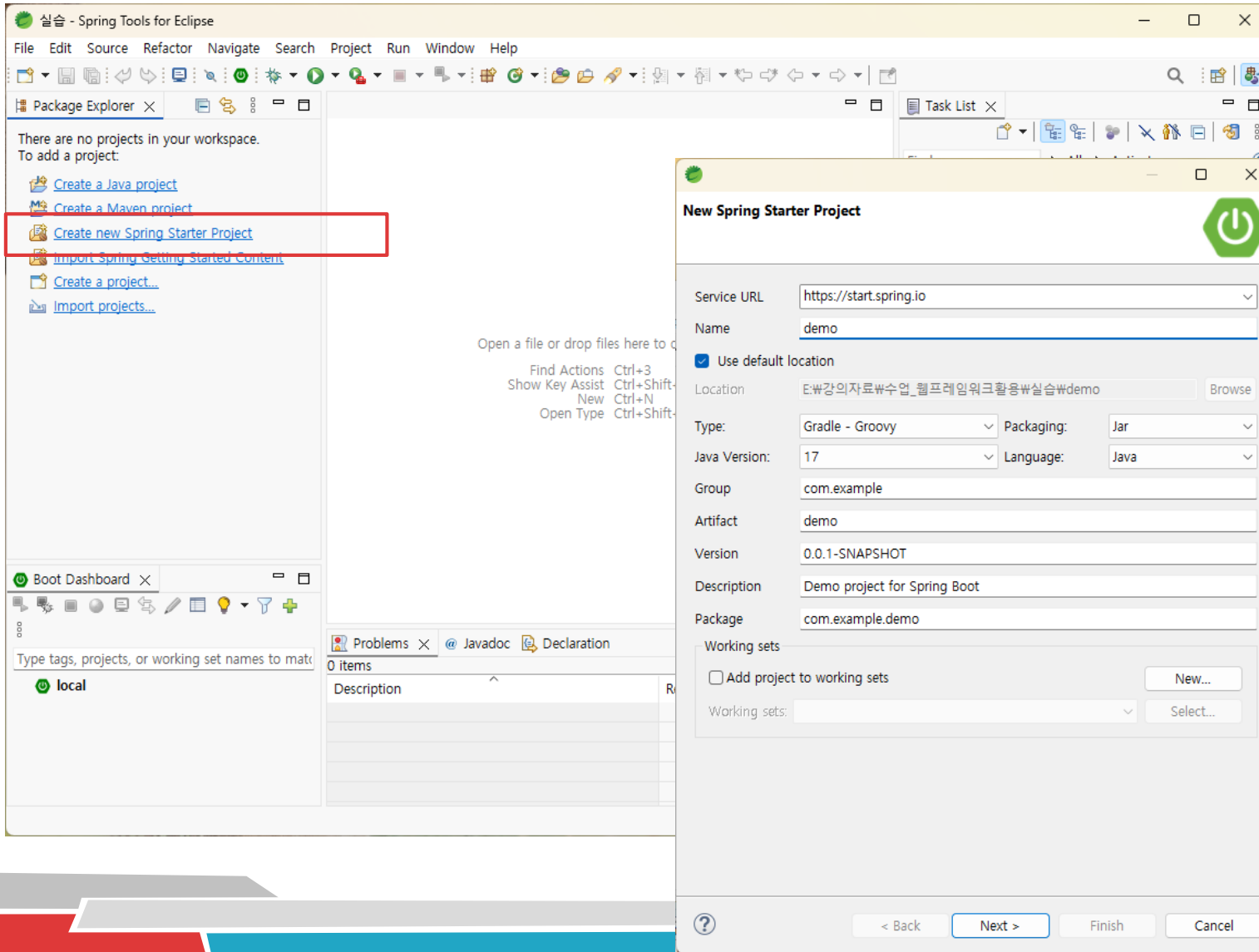
# ContentTypes Text 글꼴 변경

## 1. STS 실행 후 한글 코드 통일





# Spring 시작





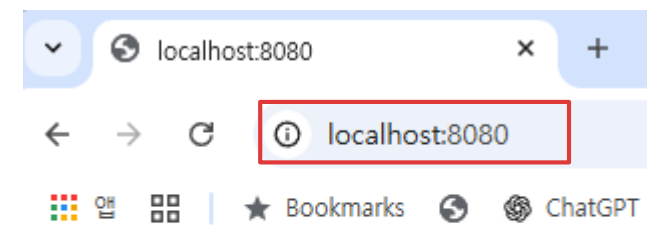
# Spring 실습1)

The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Displays the project structure for 'demo [boot]' and 'demo2 [boot] [devtools]'. The 'demo2' project is expanded, showing 'src/main/resources/templates/index.html' and 'static' directory.
- Task List:** A panel on the right showing a 'Find' search bar and 'All' and 'Activate...' buttons.
- Code Editor:** Displays the content of 'index.html' with the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     안녕하세요 Spring 반가워요~
5   </body>
6
7 </html>
8
```
- Boot Dashboard:** A panel at the bottom left showing the 'local' environment.
- Console:** A panel at the bottom right showing the output of the application. It includes the following log messages:

```
Root WebApplicationContext: initialization completed in 473 ms
ServletMapping : Adding welcome page template: index
ServletServer : LiveReload server is running on port 35729
ServletServer : Tomcat started on port 8080 (http) with context path '/'
Servlet : Started Demo2Application in 0.96 seconds (process running for 1.361)
Servlet : Initializing Spring DispatcherServlet 'dispatcherServlet'
Servlet : Initializing Servlet 'dispatcherServlet'
Servlet : Completed initialization in 0 ms
```



안녕하세요 Spring 반가워요~

The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Displays the project structure for 'demo2 [boot] [devtools]'. The 'demo2' project is expanded, showing 'src/main/resources/templates/index.html' and 'static' directory.
- Code Editor:** Displays the content of 'index.html' with the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     안녕하세요 Spring 반가워요~
5   </body>
6
7 </html>
8
```
- Run As Menu:** A context menu is open over the 'demo2' project, showing the following options:
  - Run As (highlighted with a red box)
  - Debug As
  - Profile As
  - Restore from Local History...
  - Gradle
- Run Configurations:** A panel on the right showing the 'Run As' configuration for 'demo2'. It includes the following options:
  - 1 Java Application (Alt+Shift+X, J)
  - 2 Java Application in Container
  - 3 JUnit Test (Alt+Shift+X, T)
  - 4 Spring Boot App (Alt+Shift+X, B) (highlighted with a red box)
  - 5 Spring DevTools Client
- Console:** A panel at the bottom right showing the output of the application. It includes the following log messages:

```
Root WebApplicationContext: initialization completed in 473 ms
ServletMapping : Adding welcome page template: index
ServletServer : LiveReload server is running on port 35729
ServletServer : Tomcat started on port 8080 (http) with context path '/'
Servlet : Started Demo2Application in 0.96 seconds (process running for 1.361)
Servlet : Initializing Spring DispatcherServlet 'dispatcherServlet'
Servlet : Initializing Servlet 'dispatcherServlet'
Servlet : Completed initialization in 0 ms
```

# Spring 실습2)

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: 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. The main editor shows the code for HelloController.java and hello.html. The HelloController.java code is as follows:

```
1 package com.example.demo;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6
7 @Controller
8 public class HelloController {
9
10     @GetMapping("do") // URL주소상의 위치
11     public String hello(Model model) {
12         model.addAttribute("data", "Spring~~");
13         return "hello"; // templates/hello.html 호출
14     }
15 }
```

The hello.html code is as follows:

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3     <body>
4         <p th:text=" '안녕하세요 ' + ${data}"> 안녕하세요. 손님</p>
5     </body>
6
7 </html>
```

