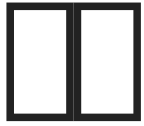


Spring Boot 入门



入门 快速入门 进阶 进阶

<https://wikidocs.net/160023> 快速入门 进阶 .

- [01. Spring Boot 入门](#)
- [02. Controller 入门](#)
- [03. Spring Boot 进阶](#)

01. 1.1.1 1.1 1.1

01. JDK 1.1



1.1 : <https://www.oracle.com/java/technologies/downloads/#jdk24-windows>

ORACLE

Products Industries Resources Customers Partners Developers Company

🔍 View Accounts Cont

Java downloads Tools and resources Java archive

Earlier JDK versions are available below.

JDK 24 JDK 21 GraalVM for JDK 24 GraalVM for JDK 21

Java SE Development Kit 24.0.2 downloads

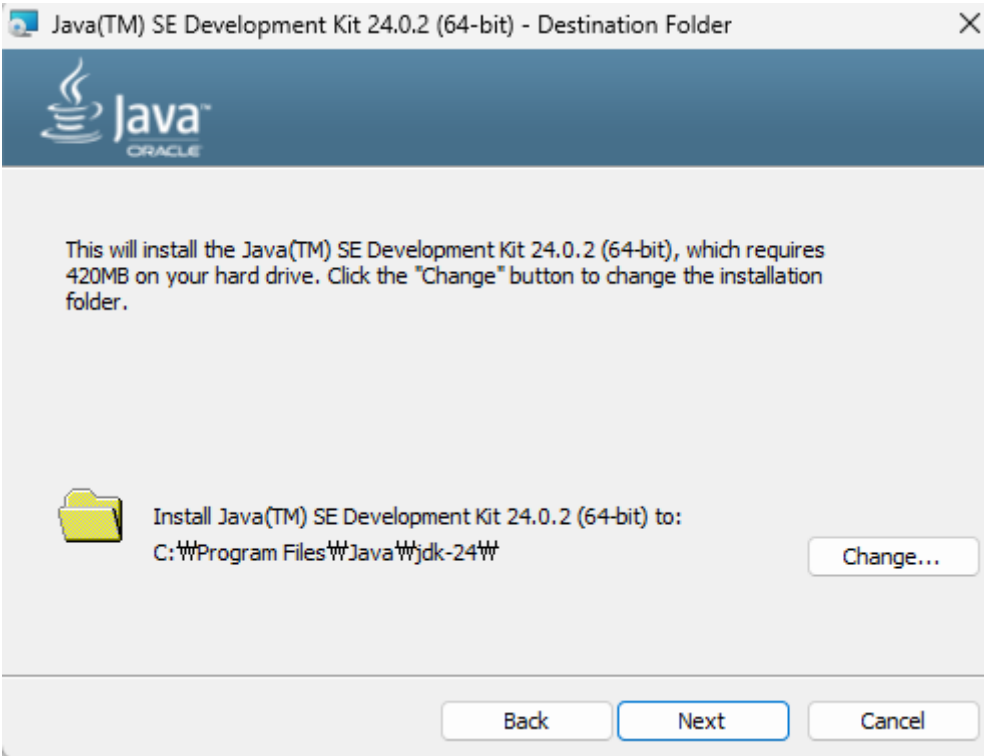
JDK 24 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 24 will receive updates under these terms, until September 2025, when it will be superseded by JDK 25.

Linux macOS Windows

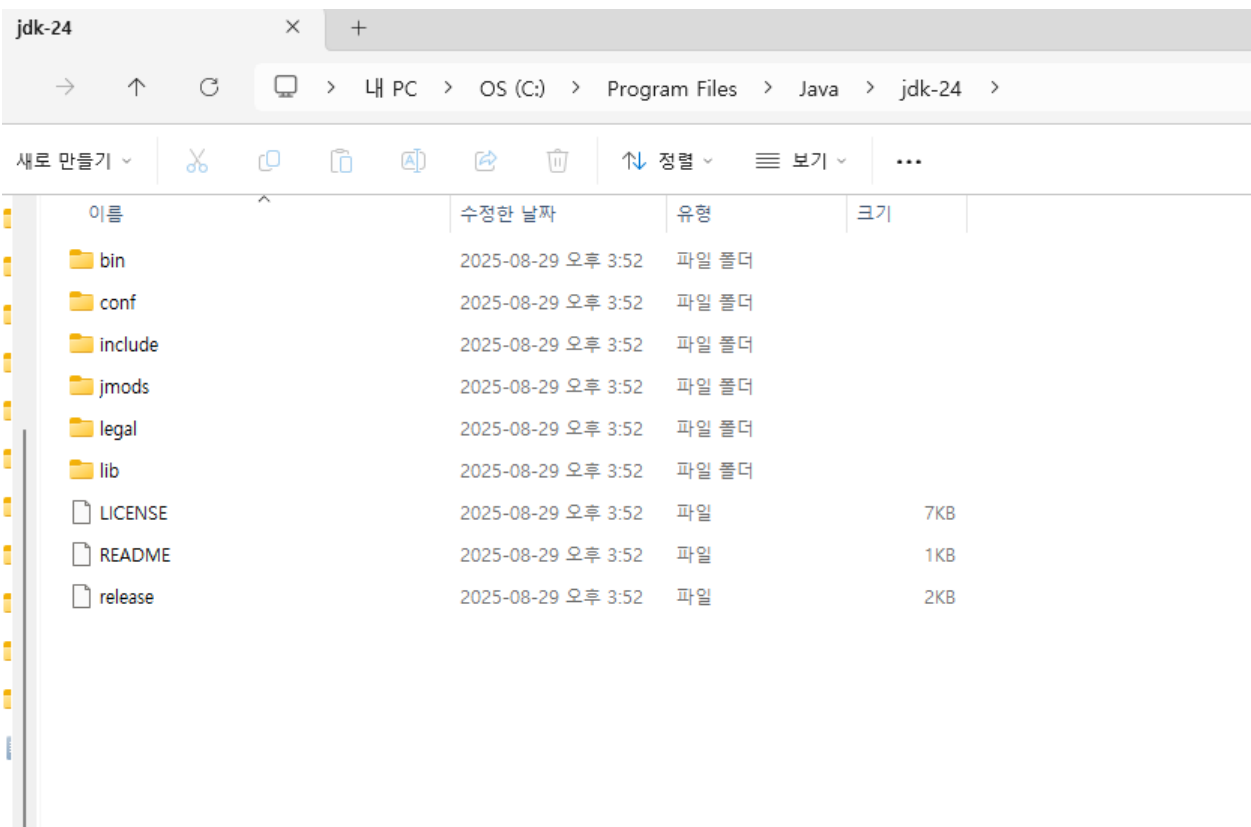
Product/file description	File size	Download
x64 Compressed Archive	229.62 MB	https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.zip (sha256)
x64 Installer	205.86 MB	https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.exe (sha256)
x64 MSI Installer	204.61 MB	https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.msi (sha256)

- 1.1 Windows 1.1 x64 installer 1.1 1.1.1 .

-

--	--	--	--

 : **C:\Program Files\Java\jdk-24**



02. STS (Spring Tool Suite)

- STS (Spring Tool Suite)

STS : <https://spring.io/tools>

Spring Tools for Eclipse

Free. Open source.

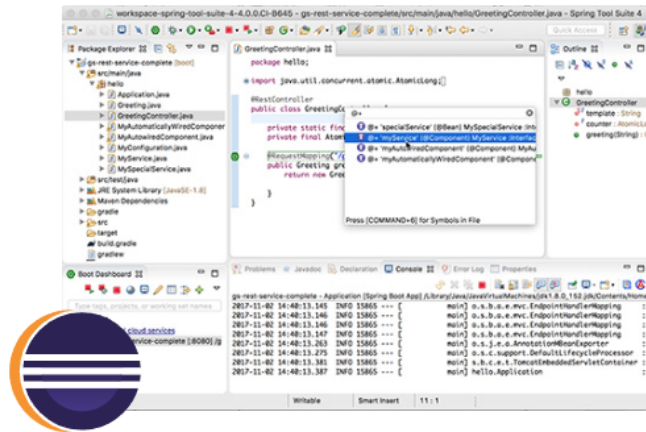
4.31.0 - LINUX X86_64

4.31.0 - LINUX ARM_64

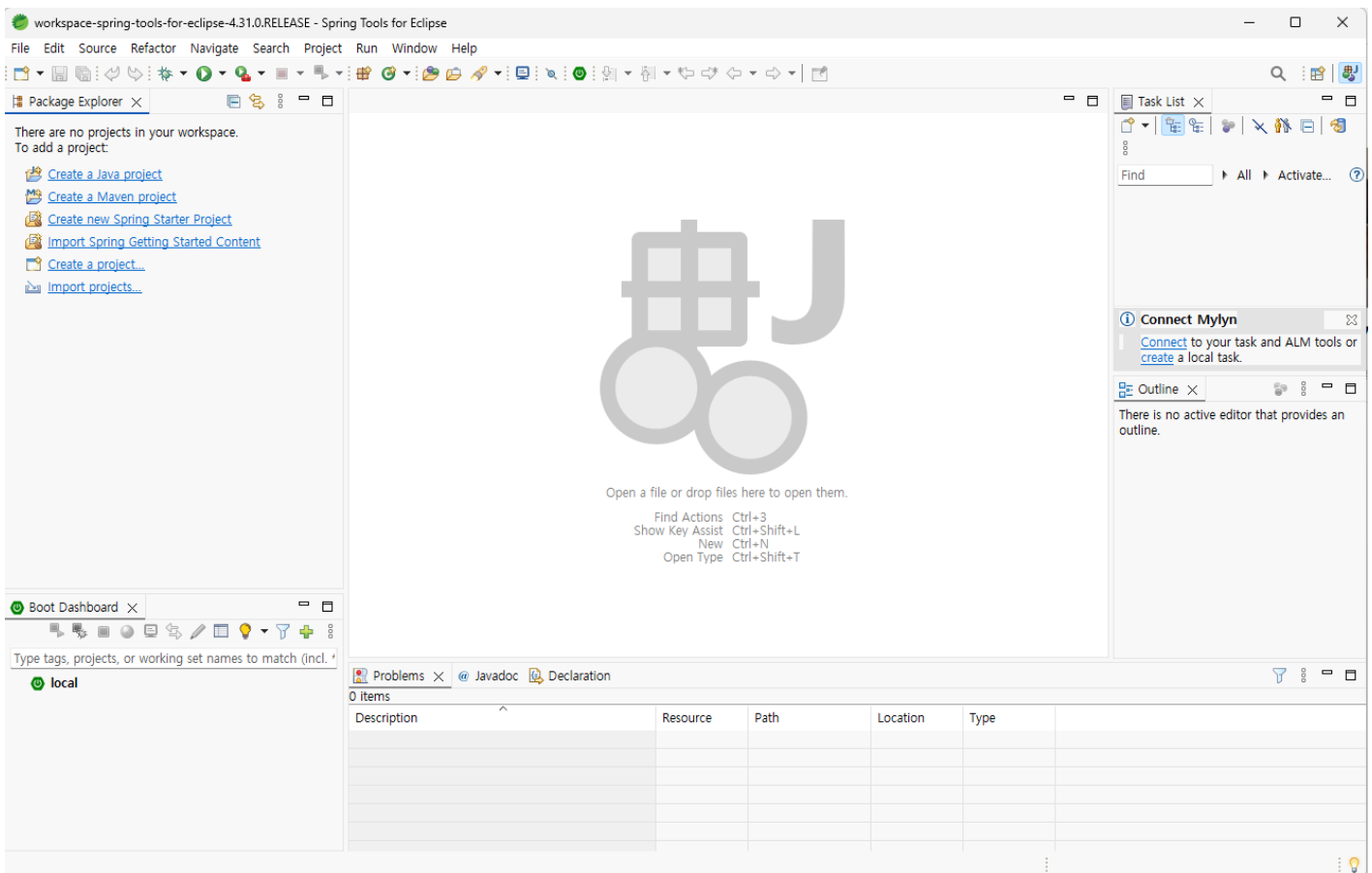
4.31.0 - MACOS X86_64

4.31.0 - MACOS ARM_64

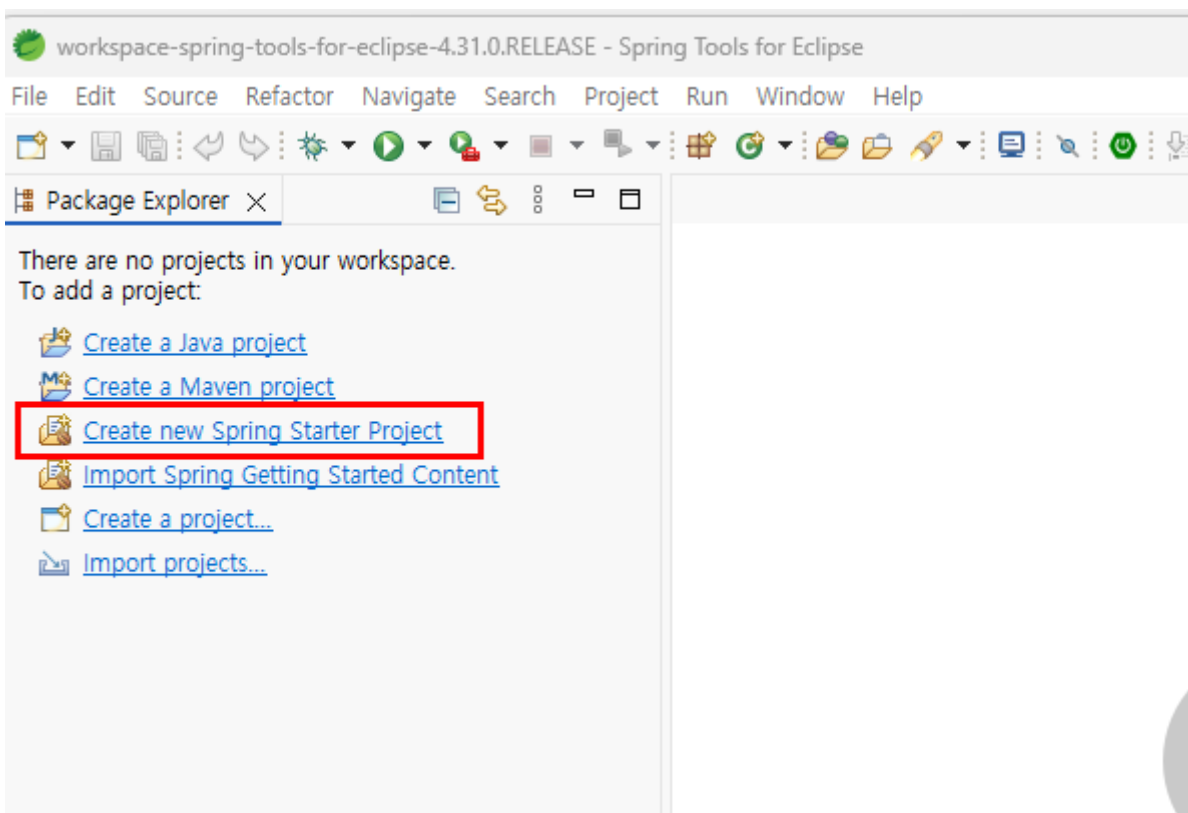
4.31.0 - WINDOWS X86_64



STS (Spring Tool Suite)



Create new Spring Starter Project





New Spring Starter Project



Service URL

https://start.spring.io

Name

demo 프로젝트 명

☒ Use default location

Location

C:\Users\Wrekw\Documents\workspace-spring-tools-for-eclipse~ Browse

Type:

Gradle - Groovy

Packaging:

Jar

Java Version:

24

Language:

Java

Group

com.example

Artifact

demo

Version

0.0.1-SNAPSHOT

Description

Demo project for Spring Boot

Package

com.example.demo

Working sets

☐ Add project to working sets

New...

Working sets:

Select...



< Back

Next >

Finish

Cancel

□□□□ □□□□□□ □□□ □□□□ .

New Spring Starter Project Dependencies



1

Spring Boot Version: 3.5.5

2

Available: web

3

▼ Messaging

☐ WebSocket

▼ Template Engines

☐ Thymeleaf

☐ Apache Freemarker

☐ Mustache

▼ Testing

☐ Testcontainers

▼ Web

☒ Spring Web

☐ Spring Reactive Web

☐ Reactive HTTP Client

☐ Spring Web Services

☐ Jersey

☐ Vaadin

Selected:

X Spring Web

Make Default

Clear Selection

?

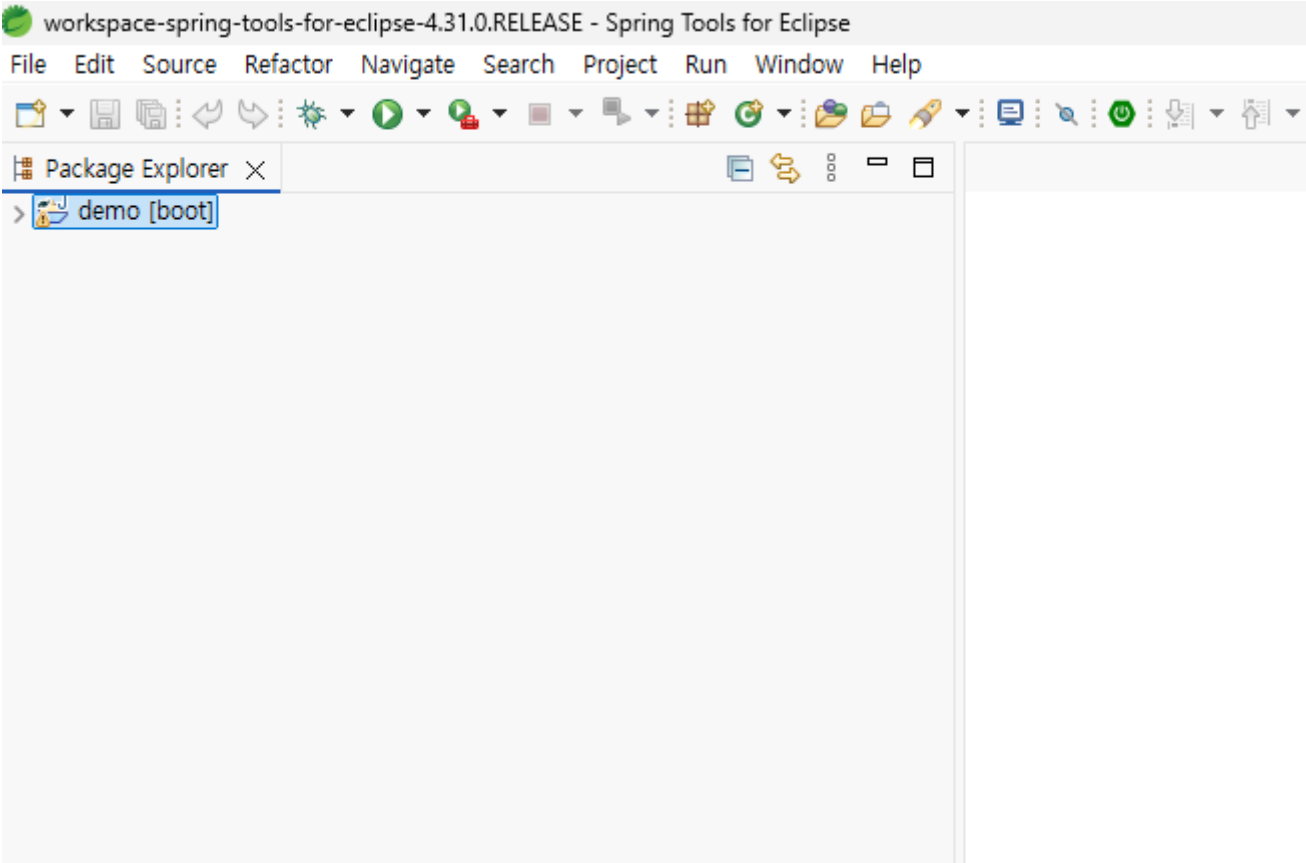
< Back

Next >

Finish

Cancel

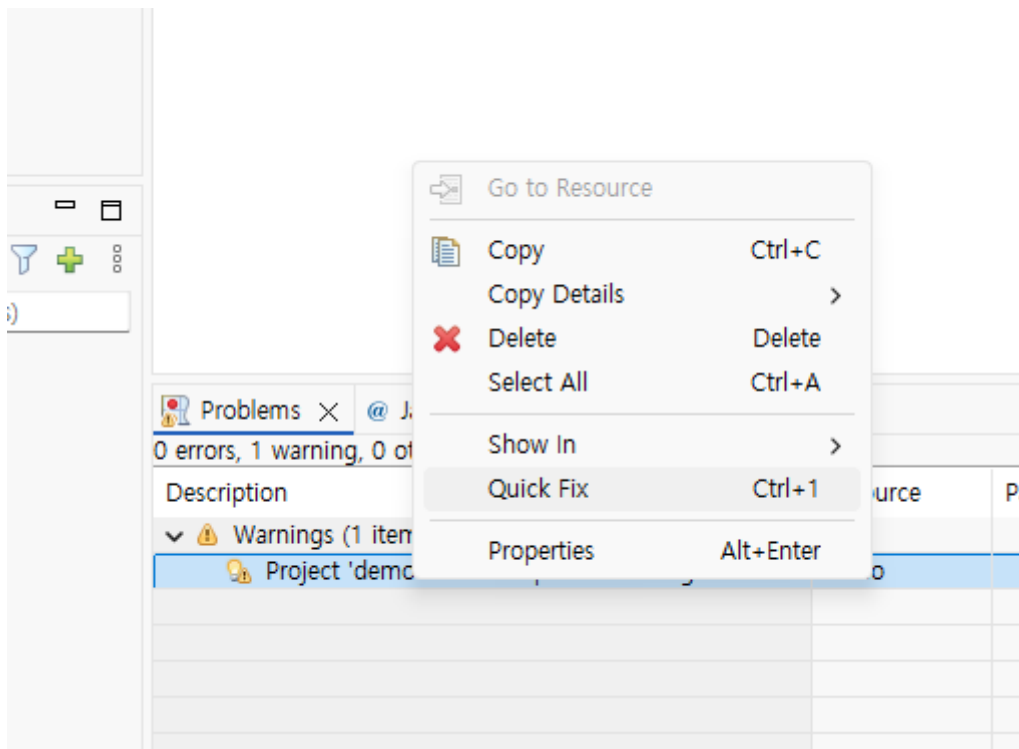
import .



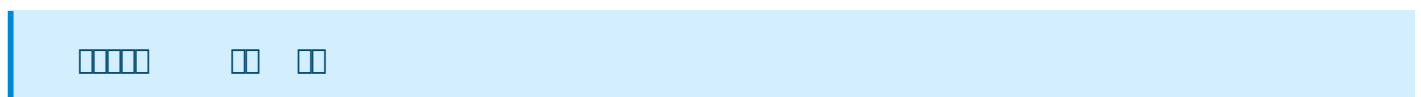
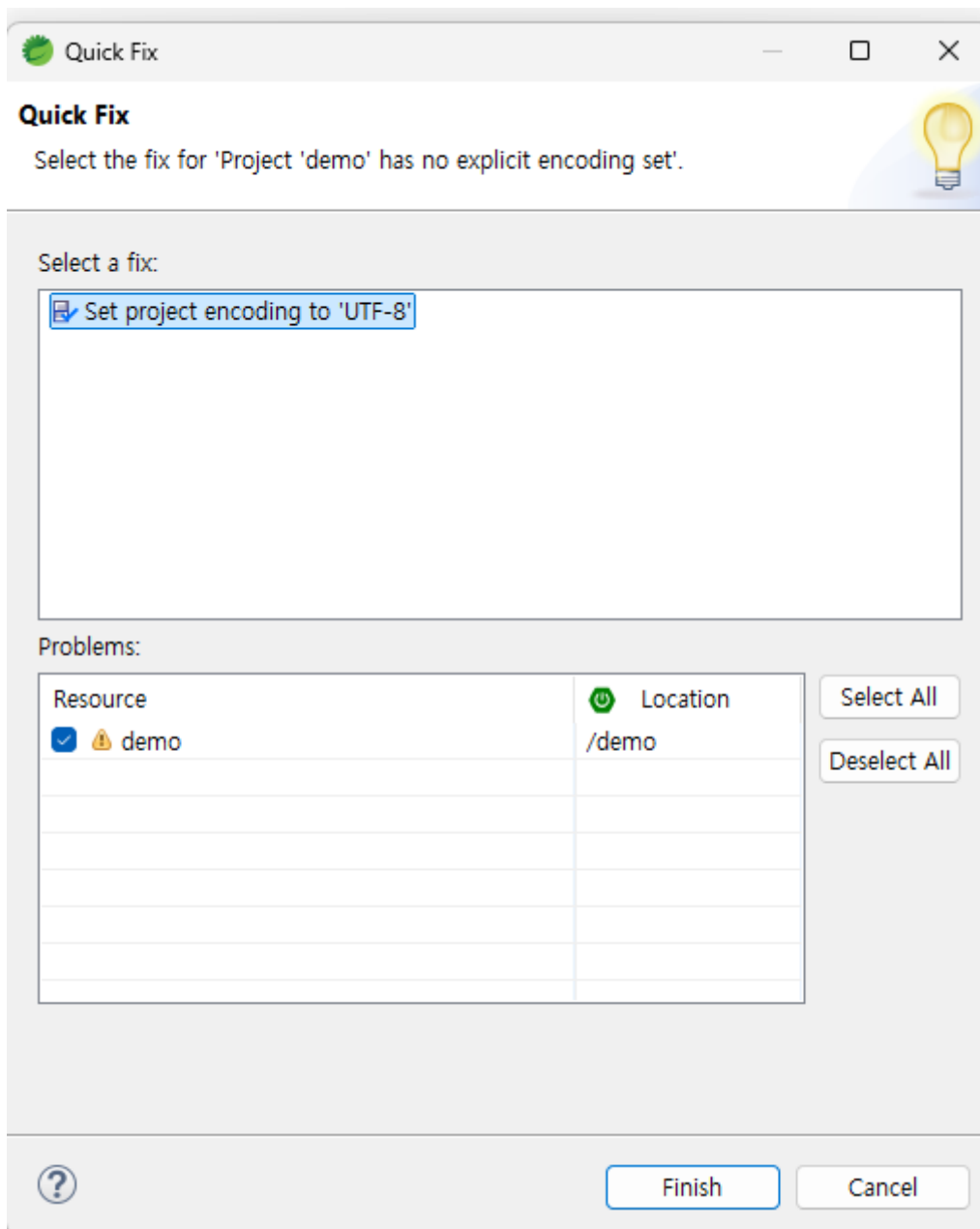
□□ □□□□ □□□□ .

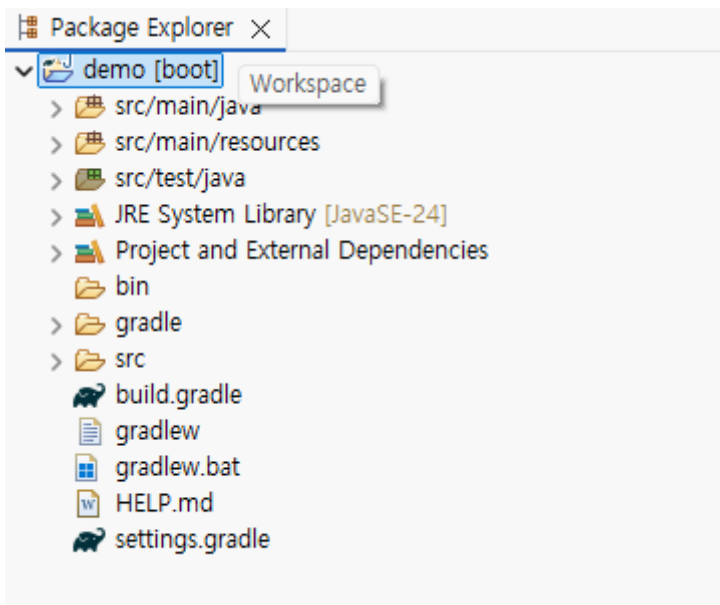
Problems × @ Javadoc Declaration					
0 errors, 1 warning, 0 others					
Description	Resource	Path	Location	Type	
Warnings (1 item)					
Project 'demo' has no explicit encoding set	demo		/demo	No explicit pr...	

□□ □□ □□ -> □□ □□ □□□□ -> Quick Fix □□



Finish





02. Controller

01.

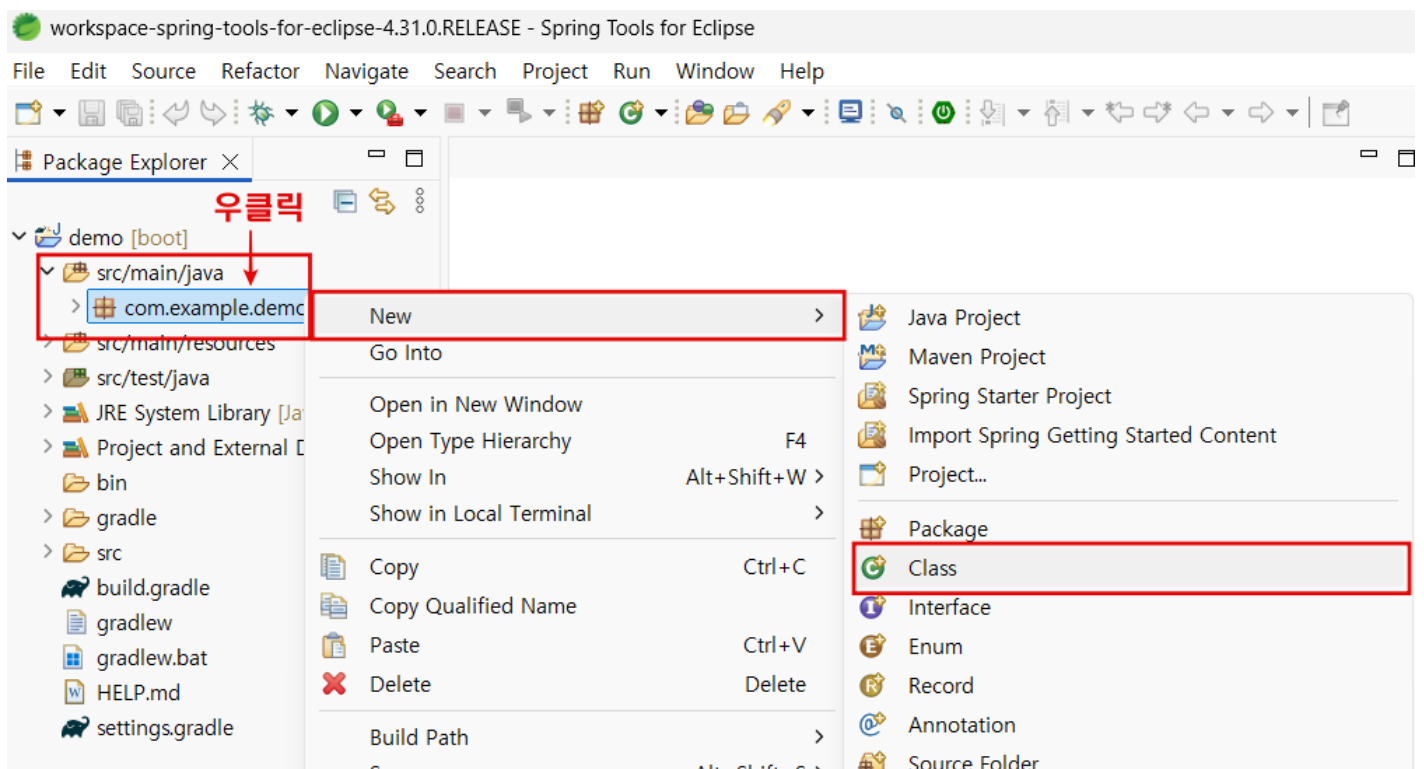
Controller 클래스는 HTTP 요청을 받아서 처리하고, 결과를 반환하는 역할을 합니다.

예를 들어, `http://localhost:8080/hello` URL로 요청이 들어오면, Controller 클래스에서 처리를 합니다.

Controller 클래스는 Spring MVC의 `DispatcherServlet`에 의해 호출됩니다.

* Controller 클래스는 `@Controller` 어노테이션을 사용하여 선언합니다.

Controller 클래스는 `Class` 타입의 객체를 반환합니다.



Controller 클래스는 `Class` 타입의 객체를 반환합니다.

New Java Class

Java Class

Create a new Java class.

Source folder:

demo/src/main/java

Browse...

Package:

com.example.demo

Browse...

☐ Enclosing type:

Browse...

컨트롤러 명 입력 후 생성

Name:

HelloController

Modifiers:

☒ public

☐ package

☐ private

☐ protected

☐ abstract

☐ final

☐ static

☒ none

☐ sealed

☐ non-sealed

☐ final

Superclass:

java.lang.Object

Browse...

Interfaces:

Add...

Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Finish

Cancel





* .

$$-\geq \begin{array}{|c|c|c|c|c|c|c|c|} \hline & & & & & & & \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline & \\ \hline \end{array} .$$

->

--	--	--	--

 ,

--	--	--

 ,

--	--	--	--	--	--

 "

--

--	--	--

--	--

--	--	--

--	--

 . "

--	--	--	--

--

-> , ,

@Controller

-> **HelloController** 클래스를 생성하고 **@Controller** 어노테이션을 추가한다.

@GetMapping("/hello")

-> **URL** 주소가 **hello**일 때 **GET** 요청을 받는다.

-> **Get** 메서드를 **URL** 주소에 **@GetMapping** 어노테이션을 추가한다.

-> **Post** 메서드를 **URL** 주소에 **@PostMapping** 어노테이션을 추가한다.

@ResponseBody

-> **JSON** 형식으로 데이터를 반환한다.

-> **String** 타입의 데이터를 반환할 때는 **"hello World"**을 **html** 형식으로 반환한다.

-> **String** 타입의 데이터를 반환할 때는 **"hello World"**을 **String** 타입으로 반환한다. (데이터베이스에서 데이터를 반환할 때는 **String** 타입으로 반환한다.)

-> **String** 타입의 데이터를 반환할 때는 **"/hello"**에 **@ResponseBody** 어노테이션을 추가한다. **String** 타입의 데이터를 반환할 때는 **"/hello"**에 **html** 형식으로 데이터를 반환한다. **String** 타입의 데이터를 반환할 때는 **"/hello"**에 **String** 타입으로 데이터를 반환한다.

-> **String** 타입의 데이터를 반환할 때는 **String** 타입으로 데이터를 반환한다. **String** 타입의 데이터를 반환할 때는 **String** 타입으로 데이터를 반환한다. **String** 타입의 데이터를 반환할 때는 **String** 타입으로 데이터를 반환한다.

02. Spring Boot

Spring Boot는 Spring Framework를 기반으로 한 오픈 소스 프레임워크이다. Spring Framework를 사용하기 위한 많은 설정을 자동으로 처리해준다.

Spring Boot는 Spring Framework를 사용하기 위한 많은 설정을 자동으로 처리해준다.

Boot Dashboard는 Spring Boot 애플리케이션을 실행하고 모니터링할 수 있는 웹 인터페이스이다.

□ □ □ □ □ □ □ □ □ □ □ □ □ □ <http://localhost:8080/hello> □ □ □ □ □

← → ↺ ⓘ localhost:8080/hello

Hello World

03.

--	--	--	--	--

--	--

--	--	--	--












$$\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} = \begin{array}{|c|} \hline \\ \hline \end{array} + \begin{array}{|c|} \hline \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline & & \\ \hline \end{array} + \begin{array}{|c|c|} \hline & \\ \hline \end{array} + \begin{array}{|c|} \hline \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} + \begin{array}{|c|} \hline \\ \hline \end{array} + \begin{array}{|c|} \hline \\ \hline \end{array} + \begin{array}{|c|c|} \hline & \\ \hline \end{array} .$$

01 . Spring Boot Devtools



Devtools































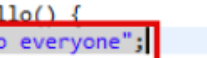




*

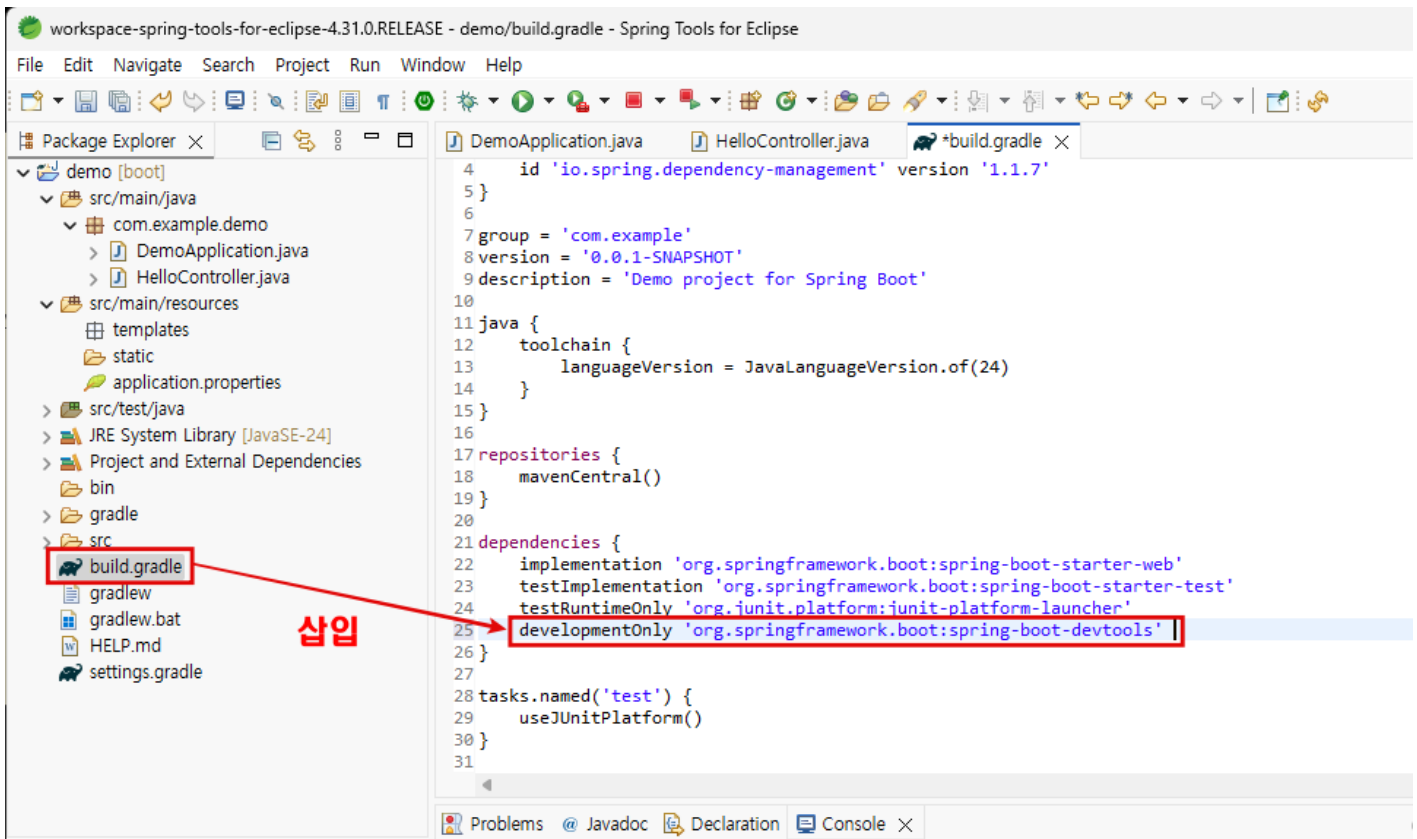
11

01. ☐ ☐ ☐ ☒ ☐ ☐

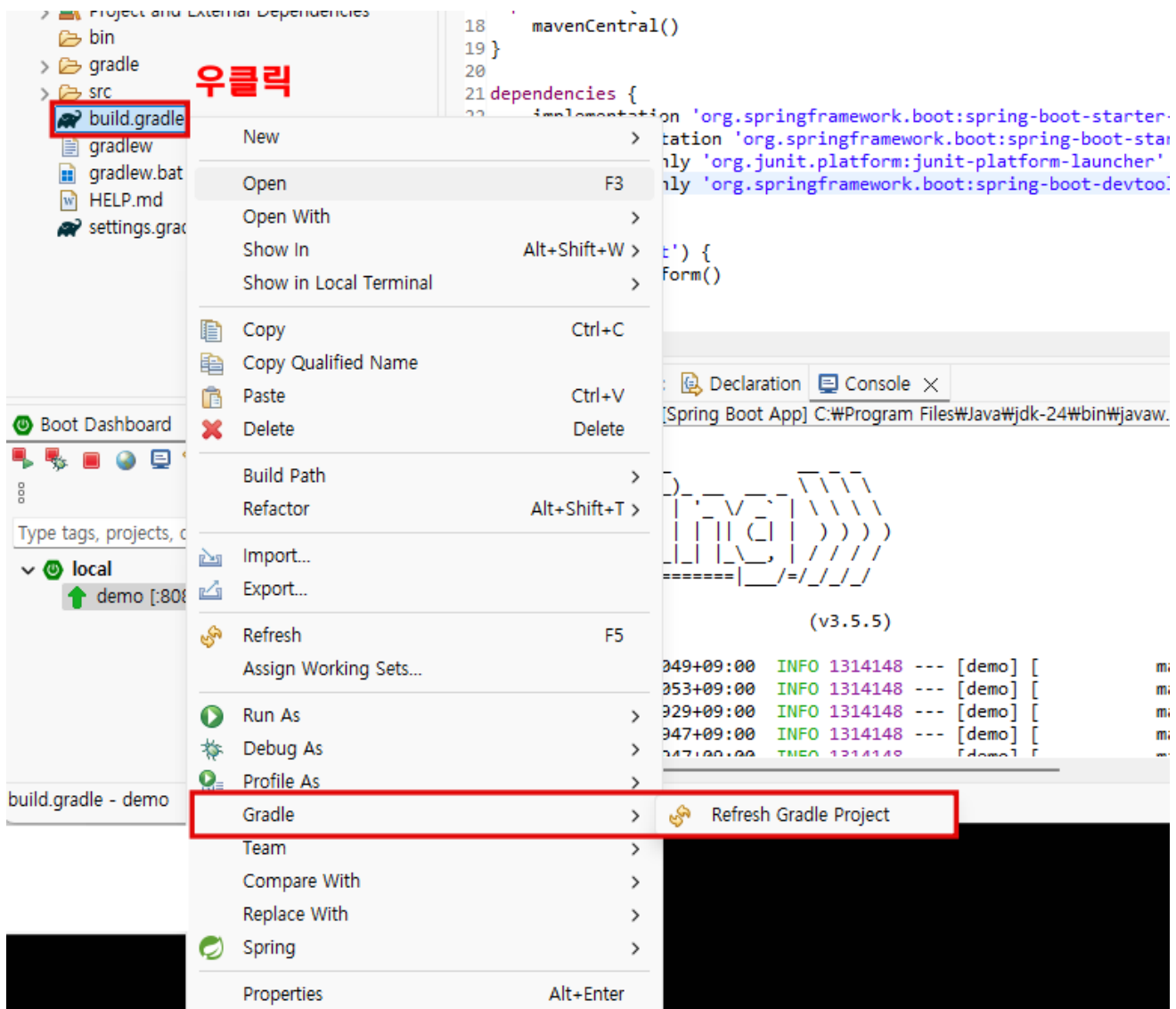


01. ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ **build.gradle** ☐ ☐ ☐ ☐ **Devtool** ☐ ☐

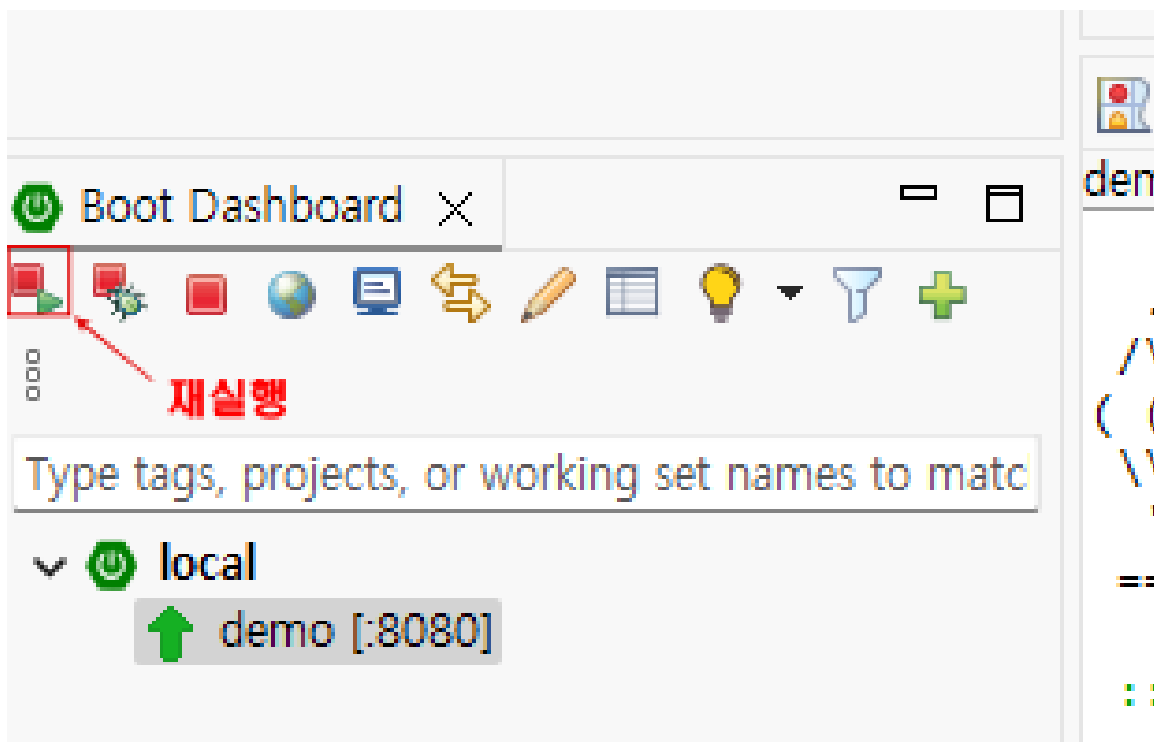
```
developmentOnly 'org.springframework.boot:spring-boot-devtools'
```

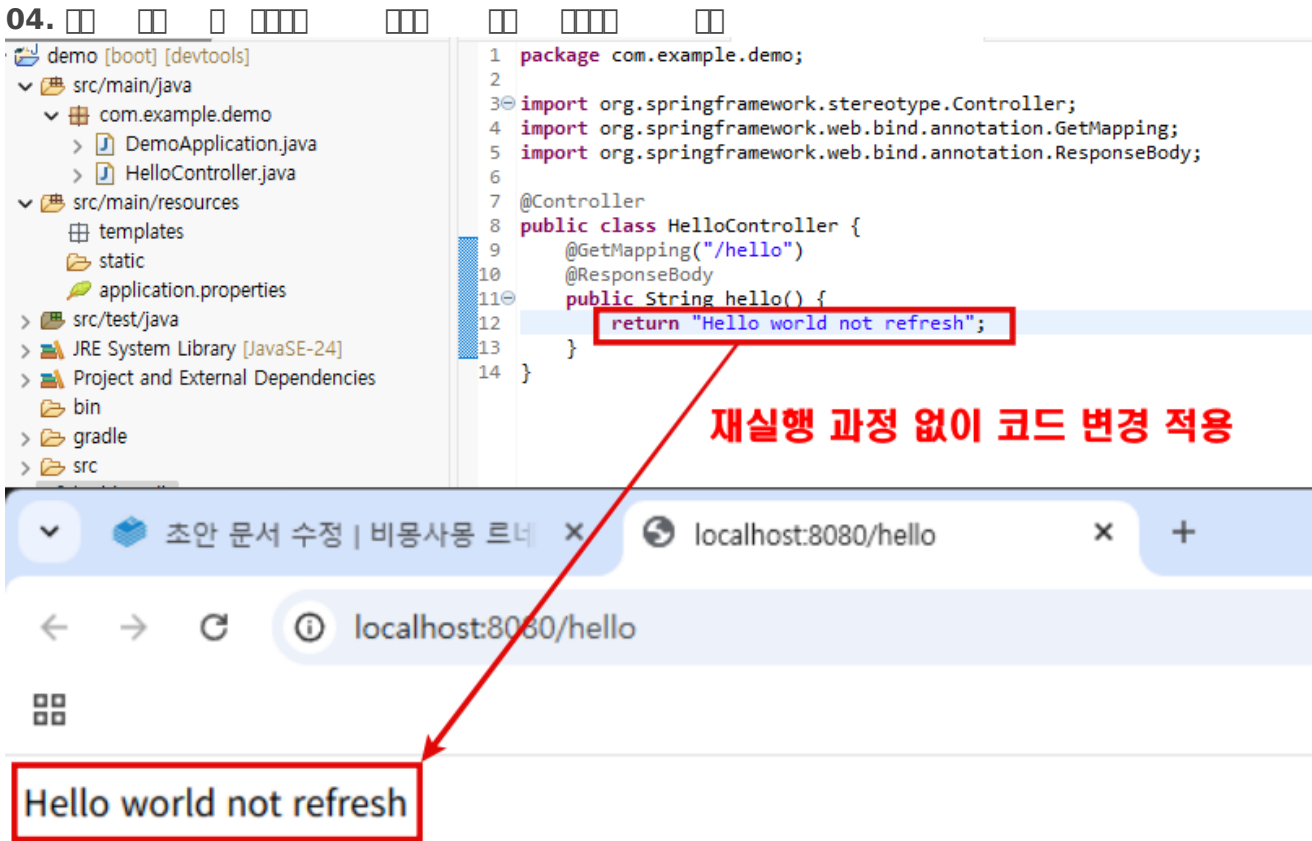


02. Refresh Gradle Project ☐ ☐ ☐ ☐



03. devtool

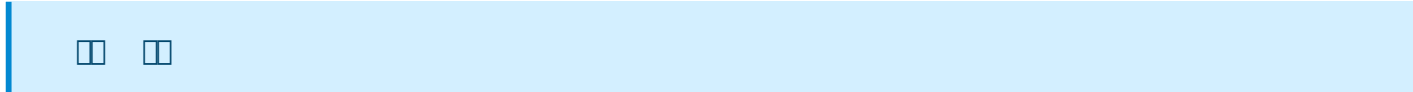




02. Lombok()

DTO

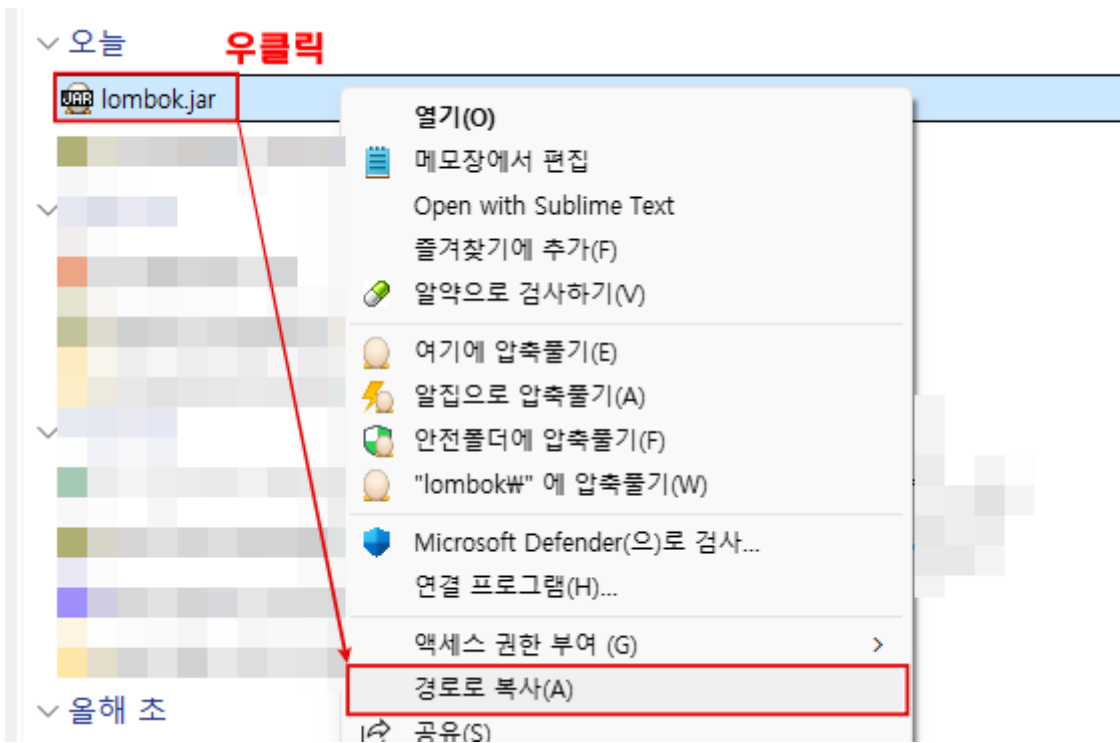
Getter, Setter



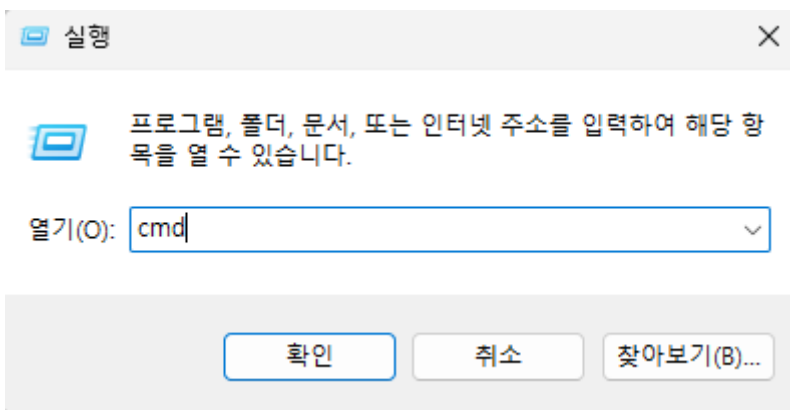
01. .

: <https://projectlombok.org/download>

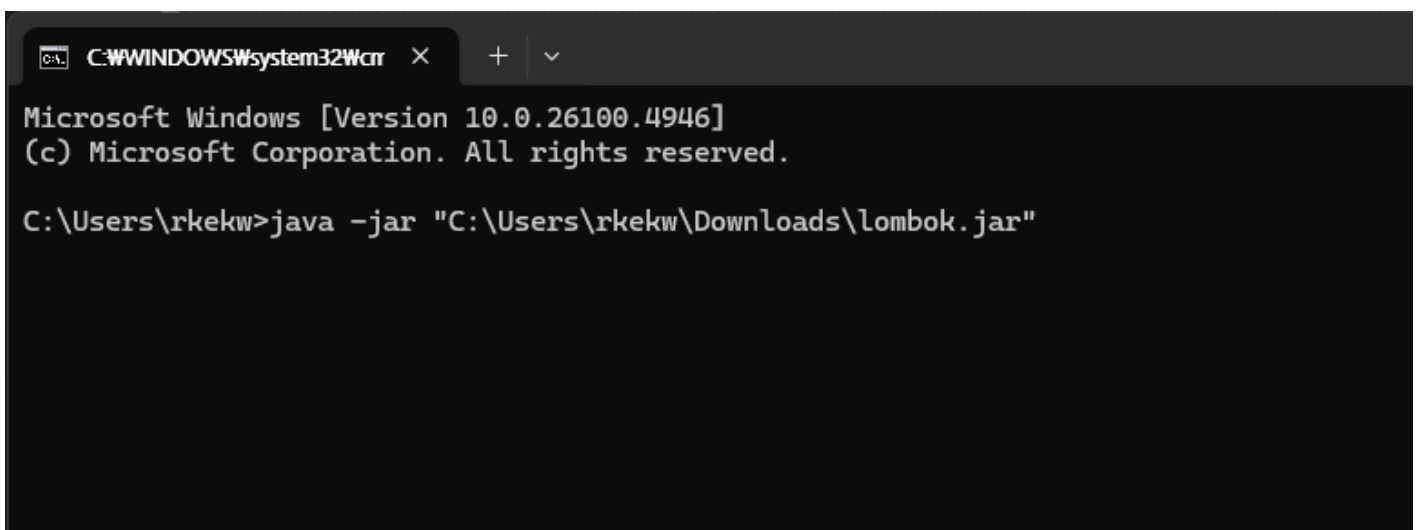
02. .jar



03. win + r cmd

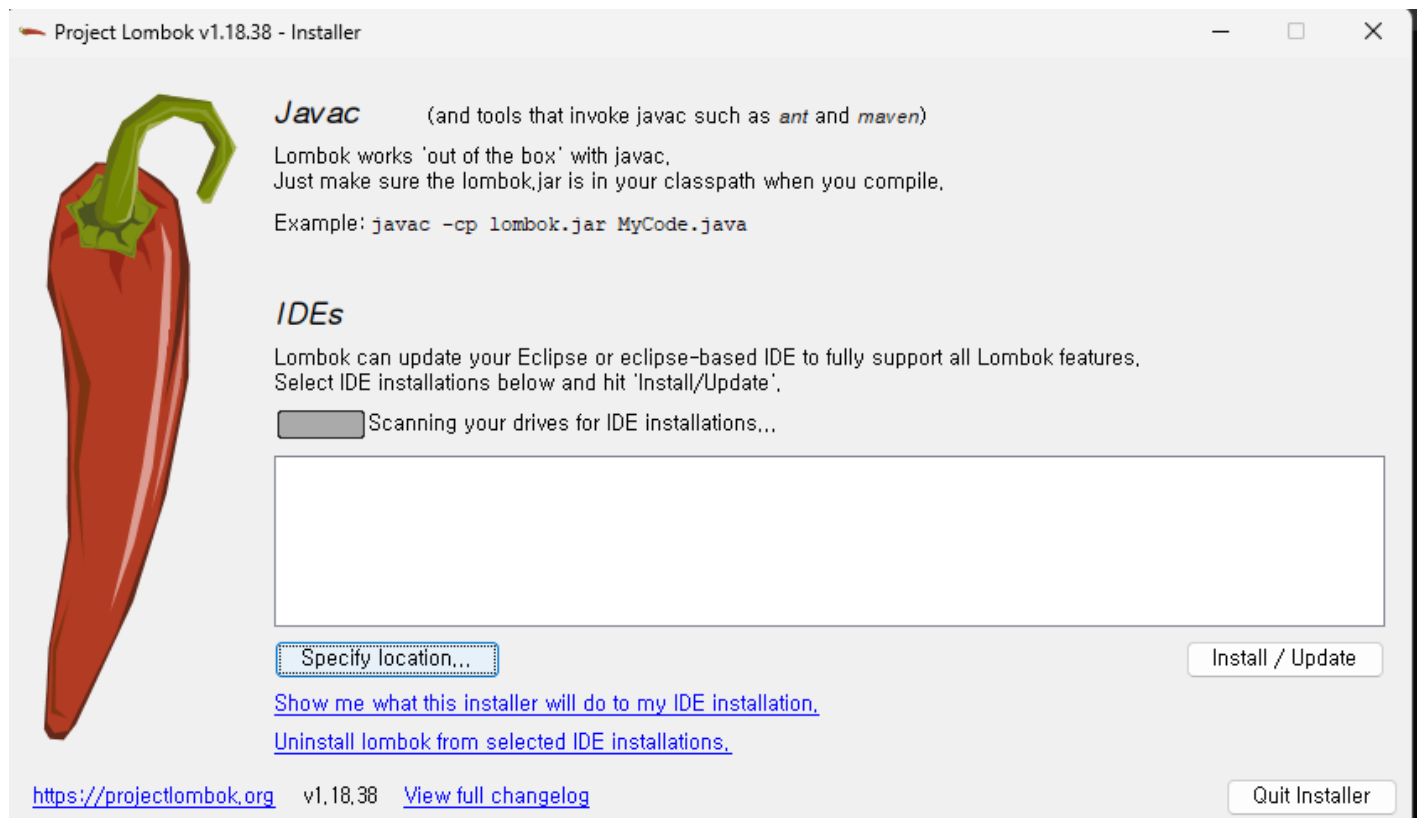


04. java -jar (cmd)

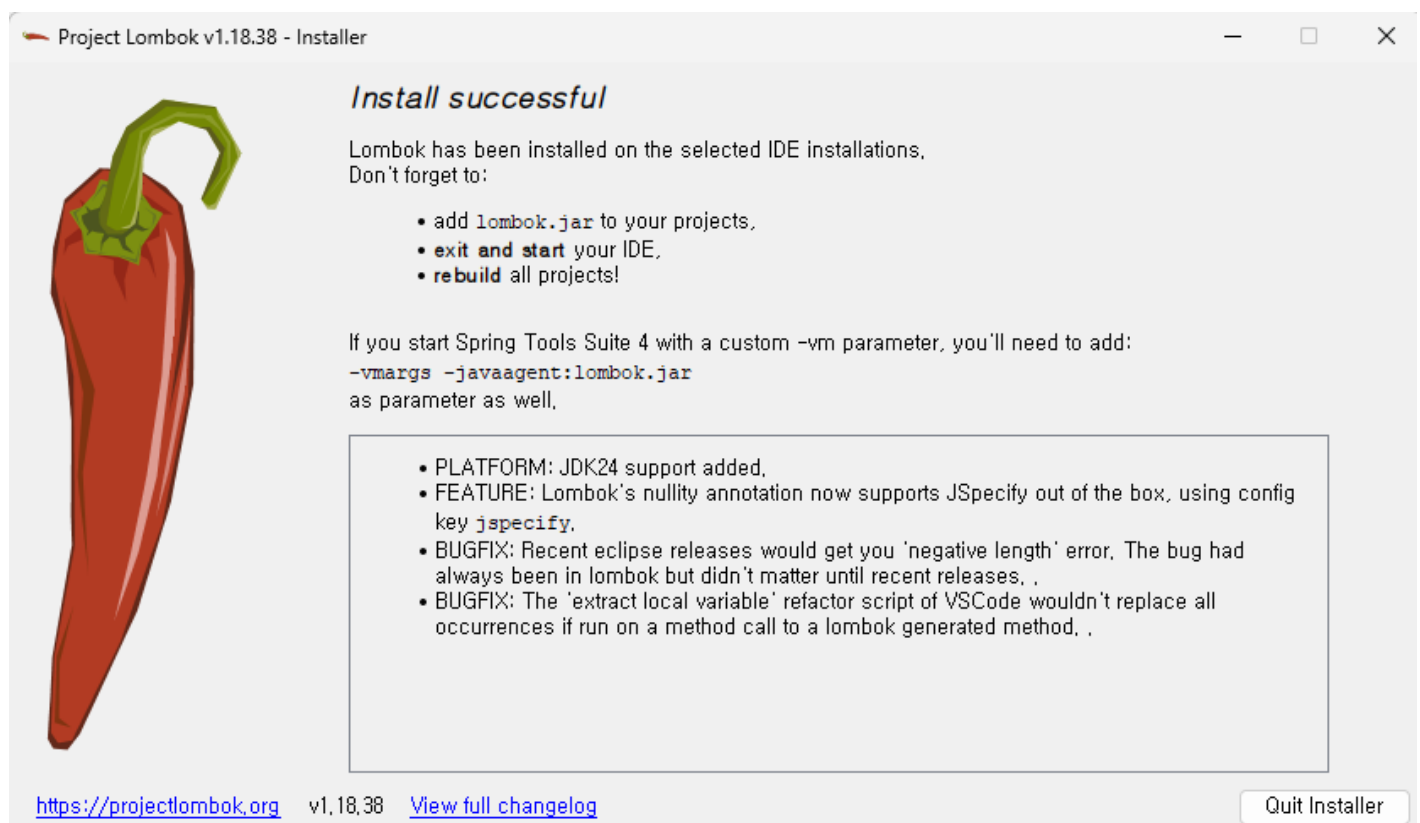


05. STS .

* Specify location .



06.



07. STS build.gradle

이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다.

SpringToolSuite4.ini 파일을 열고 다음 줄을 추가합니다. **lombok**은 Lombok 라이브러리의 경로입니다. **lombok.jar**은 Lombok 라이브러리의 JAR 파일입니다.

이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다.

.ini 파일을 열고 --add-opens 줄을 추가합니다. **Java**은 Java 런타임의 경로입니다.

이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다. Lombok은 Java 24 이상에서 사용할 수 있습니다. IDE에서 Lombok을 사용할 수 있도록 설정합니다.

이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다. (grok)

 .eclipseproduct	2025-08-29 오후 4:00	ECLIPSEPRODUCT...	1KB
 artifacts.xml	2025-08-29 오후 4:00	Microsoft Edge H...	159KB
 license.txt	2025-08-29 오후 4:00	TXT 파일	12KB
 lombok.jar	2025-09-01 오전 11:44	ALZip JAR File	2,008KB
 open-source-licenses.txt	2025-08-29 오후 4:00	TXT 파일	633KB
 SpringToolSuite4.exe	2025-08-29 오후 4:00	응용 프로그램	538KB
 SpringToolSuite4.ini	2025-09-01 오후 12:02	구성 설정	2KB
 SpringToolSuite4c.exe	2025-08-29 오후 4:01	응용 프로그램	248KB

01. STS를 사용하여 Lombok을 사용하는 방법을 설명합니다. ini 파일을 열고 -vmargs 줄을 추가합니다. -javaagent 줄을 추가합니다.

-vmargs 줄을 추가합니다. -vmargs 줄을 추가합니다.

--add-opens=java.base/java.lang=ALL-UNNAMED

--add-opens=java.base/java.util=ALL-UNNAMED

--add-opens=java.base/java.util=ALL-UNNAMED

--add-opens=java.base/java.util=ALL-UNNAMED

-javaagent 줄을 추가합니다. lombok.jar 줄을 추가합니다.

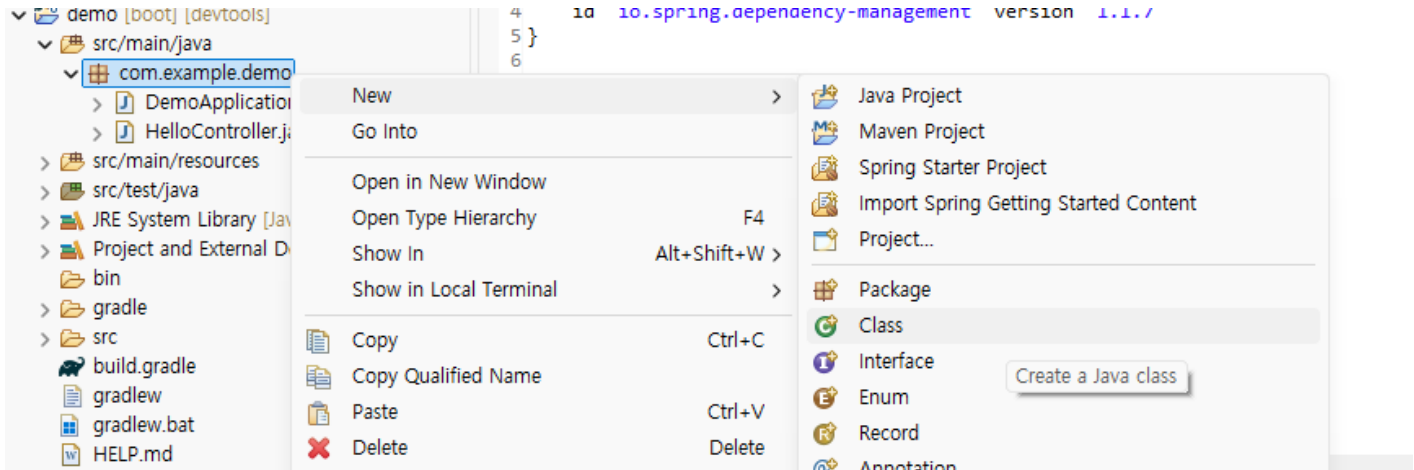
이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다. \01. STS - sts-4.31.0.RELEASE\lombok.jar

이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다. \02. STS - sts-4.31.0.RELEASE\lombok.jar

이 문서에서는 STS를 사용하여 Lombok을 사용하는 방법을 설명합니다. -javaagent:lombok.jar

Lombok

01. com.example.demo



02.

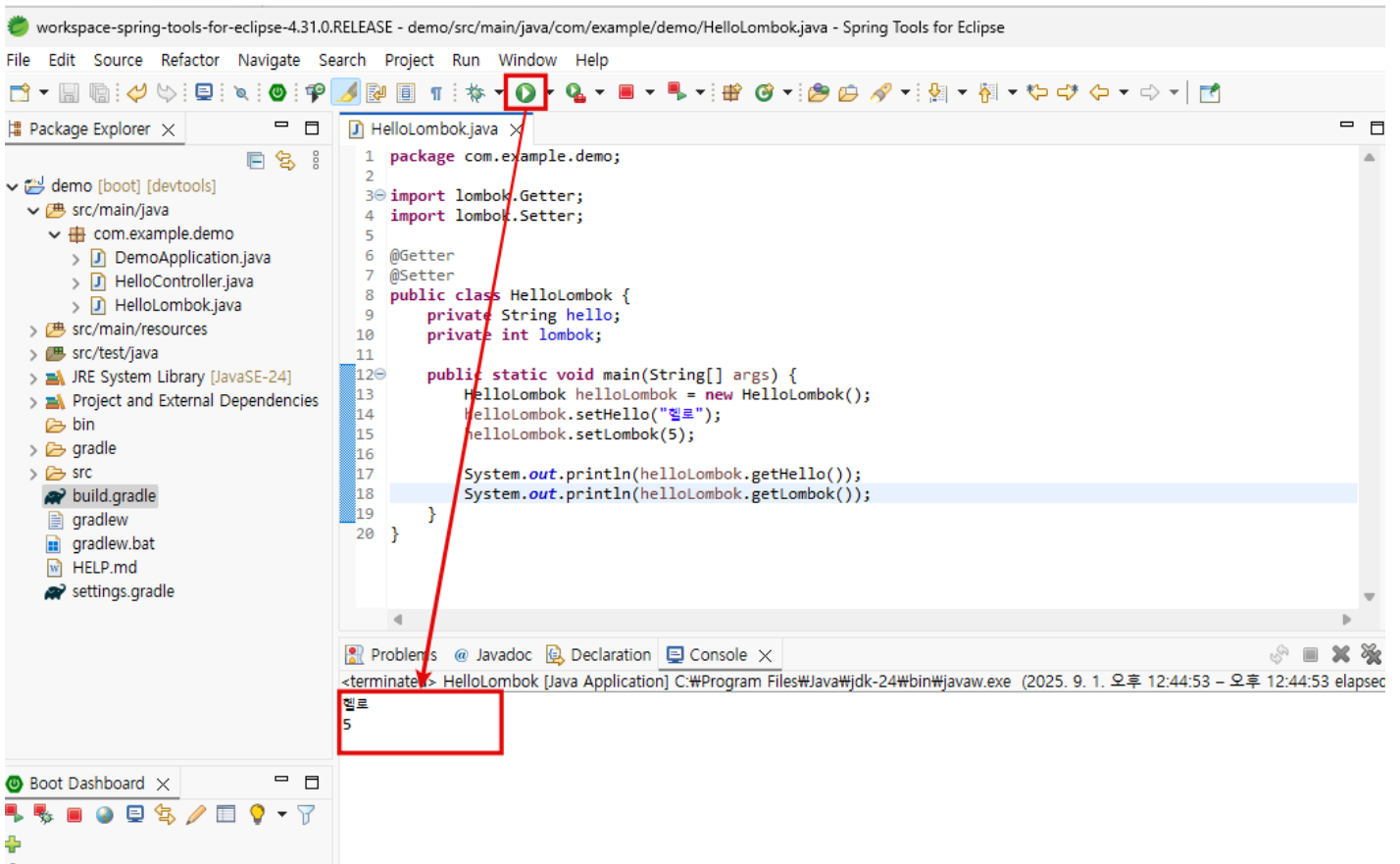
```
package com.example.demo;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
public class HelloLombok {
    private String hello;
    private int lombok;

    public static void main(String[] args) {
        HelloLombok helloLombok = new HelloLombok();
        helloLombok.setHello("");
        helloLombok.setLombok(5);

        System.out.println(helloLombok.getHello());
        System.out.println(helloLombok.getLombok());
    }
}
```



□ □ □ □ □ □ □ □ □ □

***Lombok** □ □ □ □ □ □ □ □ □ □

Lombok □ □ □ □ □ □ □ □

```
package com.example.demo;

public class HelloLombok {
    private String hello;
    private int lombok;

    public void setHello(String hello) {
        this.hello = hello;
    }

    public void setLombok(int lombok) {
        this.lombok = lombok;
    }
}
```

```

public String getHello() {
    return this.hello;
}

public int getLombok() {
    return this.lombok;
}

public static void main(String[] args) {
    HelloLombok helloLombok = new HelloLombok();
    helloLombok.setHello("Halo");
    helloLombok.setLombok(5);

    System.out.println(helloLombok.getHello());
    System.out.println(helloLombok.getLombok());
}
}

```

Lombok   

```

package com.mysite.sbb;
import lombok.Getter;
import lombok.RequiredArgsConstructor;

@RequiredArgsConstructor
@Getter
public class HelloLombok {
    private final String hello;
    private final int lombok;

    public static void main(String[] args) {
        HelloLombok helloLombok = new HelloLombok("Halo", 5);
        System.out.println(helloLombok.getHello());
        System.out.println(helloLombok.getLombok());
    }
}

```

@Getter  **RequiredArgsConstructor**        .

@Getter - [] [] [] getter [] [] [] [] [] [] []

@RequiredArgsConstructor -