

03. 数据库 连接池

数据库连接池 (Database Connection Pool) 是一种用于管理数据库连接的技术。它通过预先创建并维护一定数量的数据库连接，当应用程序需要访问数据库时，可以从池中获取连接，使用完毕后再将连接放回池中，从而实现连接的复用，提高数据库操作的效率。

数据库连接池的主要作用包括：

- 减少数据库连接的创建和销毁开销。
- 提高数据库操作的并发性能。
- 防止数据库连接被耗尽。

01 . Spring Boot Devtools 连接池

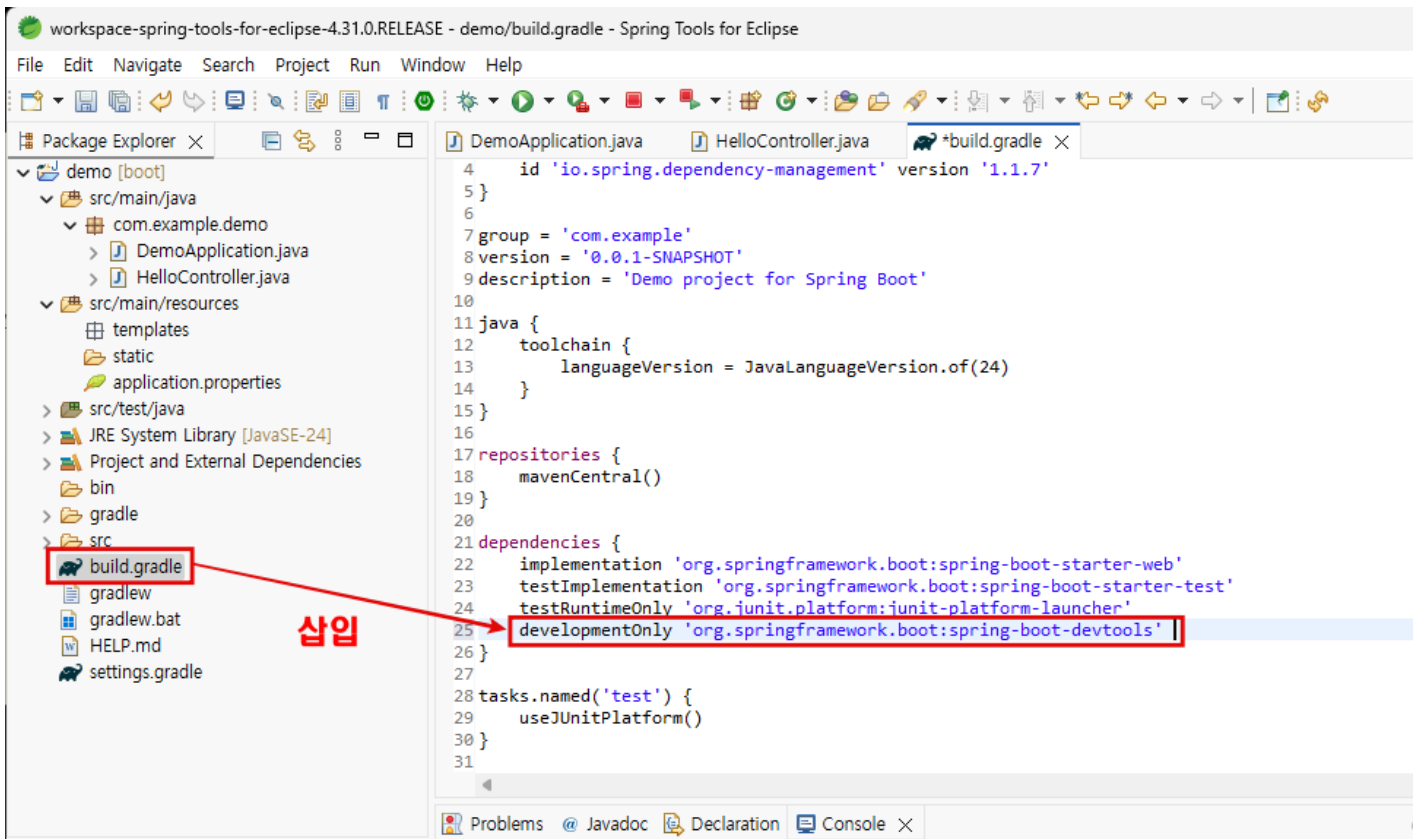
在 Spring Boot 应用中，使用 Devtools 连接池可以方便地管理数据库连接。Devtools 连接池支持多种数据库，如 MySQL、Oracle、PostgreSQL 等。

* 在 Spring Boot 应用中，使用 Devtools 连接池的步骤如下：

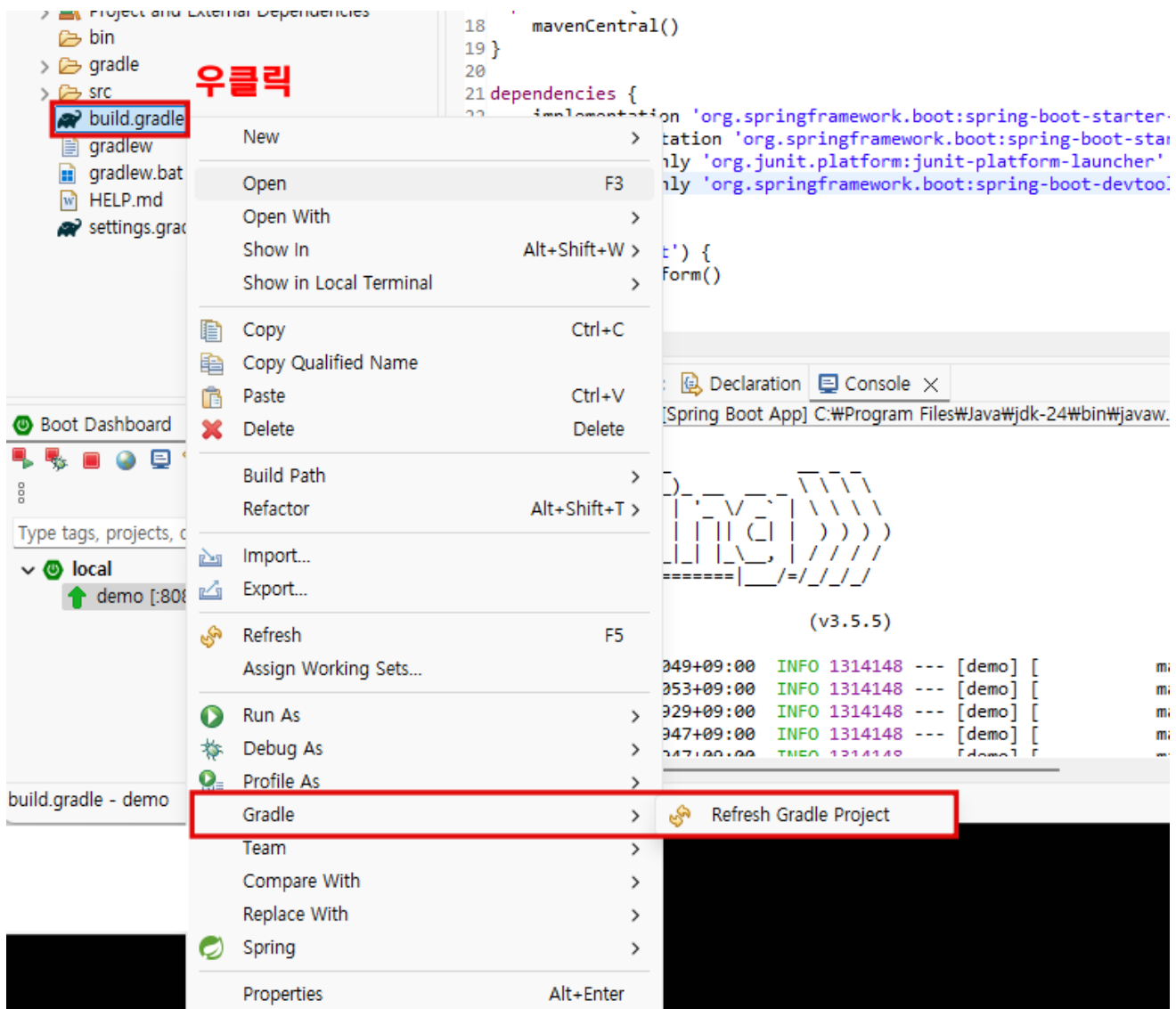
1. 在 Spring Boot 应用中，添加 Devtools 连接池的依赖。

01. 在 Spring Boot 应用中，添加 Devtools 连接池的依赖。

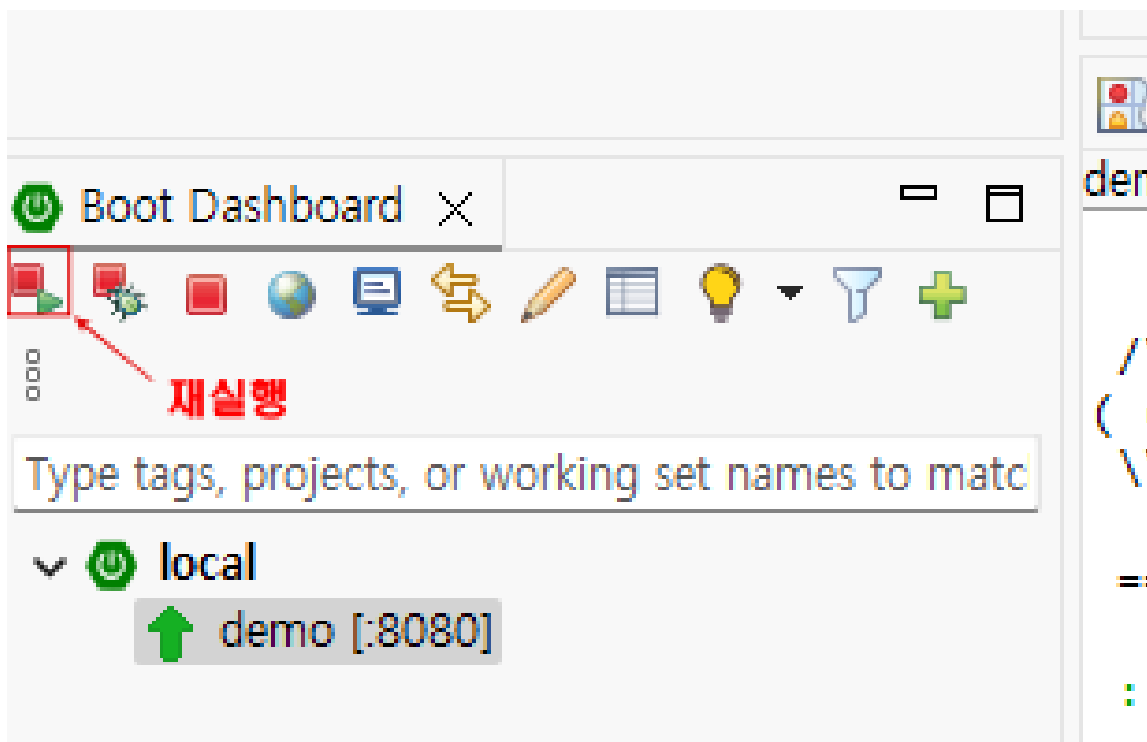

```
developmentOnly 'org.springframework.boot:spring-boot-devtools'
```

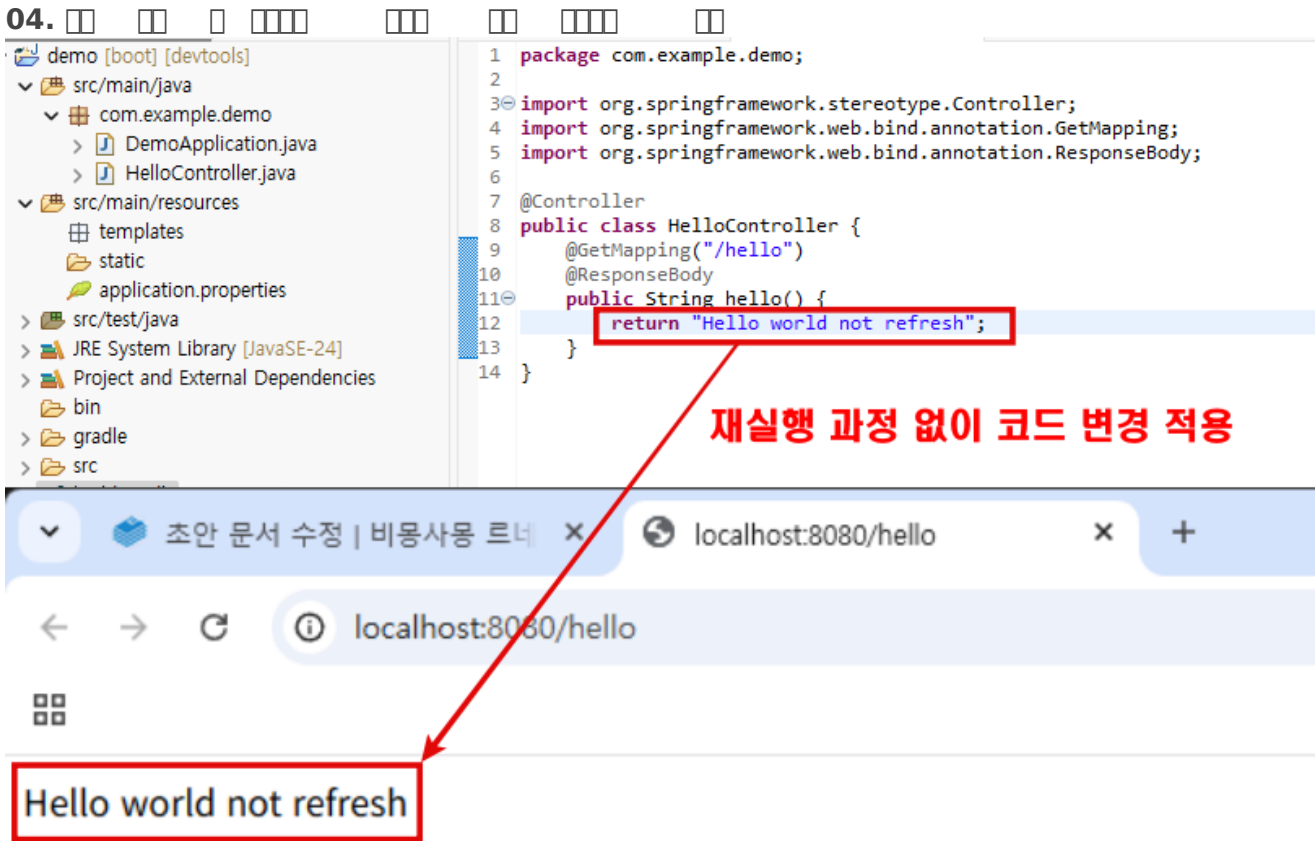


02. Refresh Gradle Project ☐ ☐ ☐ ☐



03. devtool

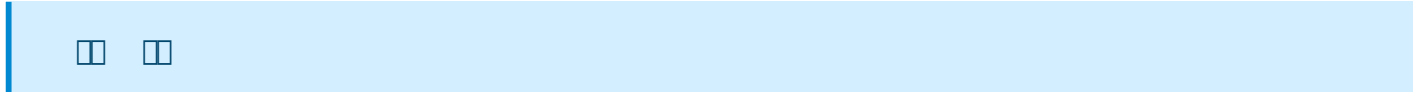




02. Lombok()

DTO

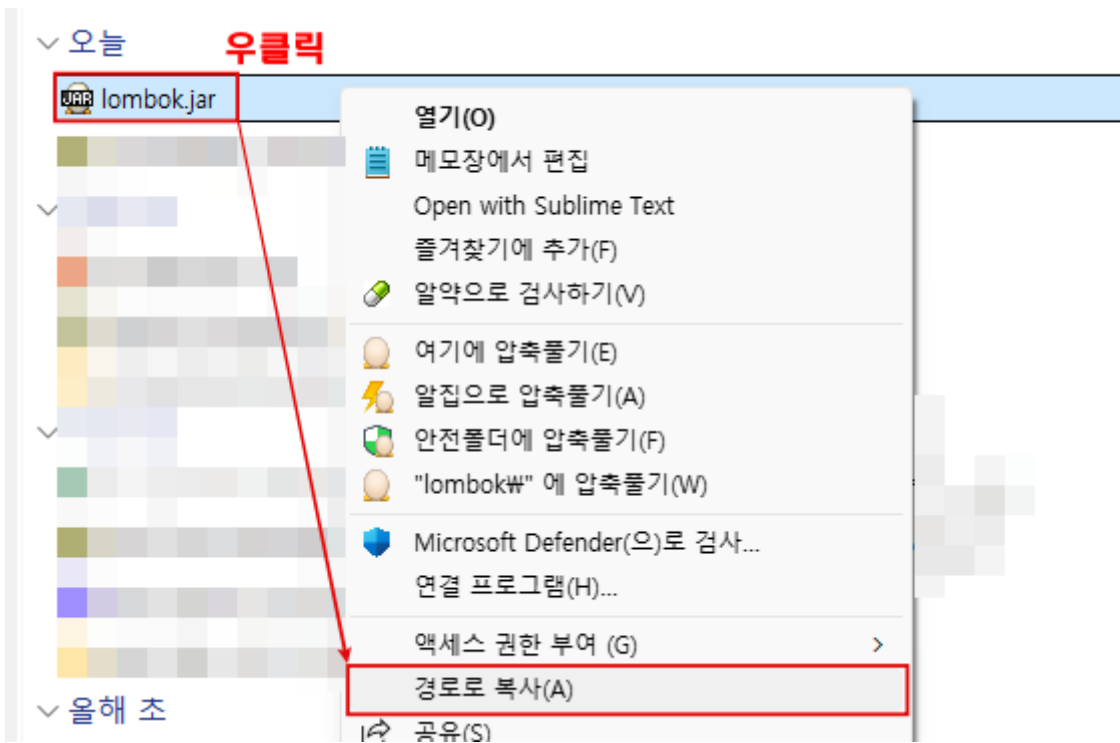
Getter, Setter



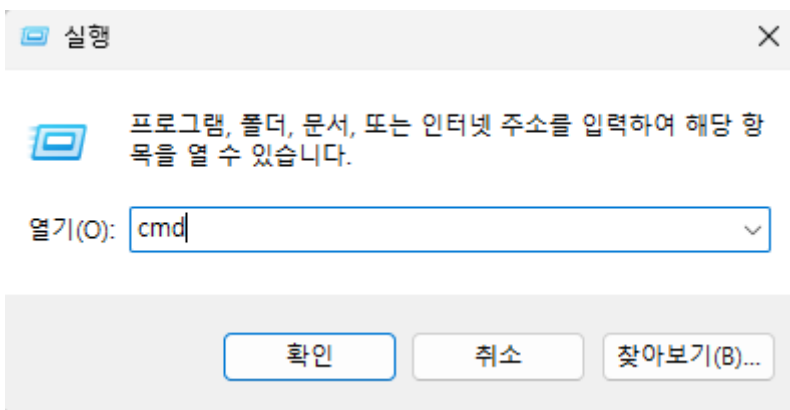
01. .

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

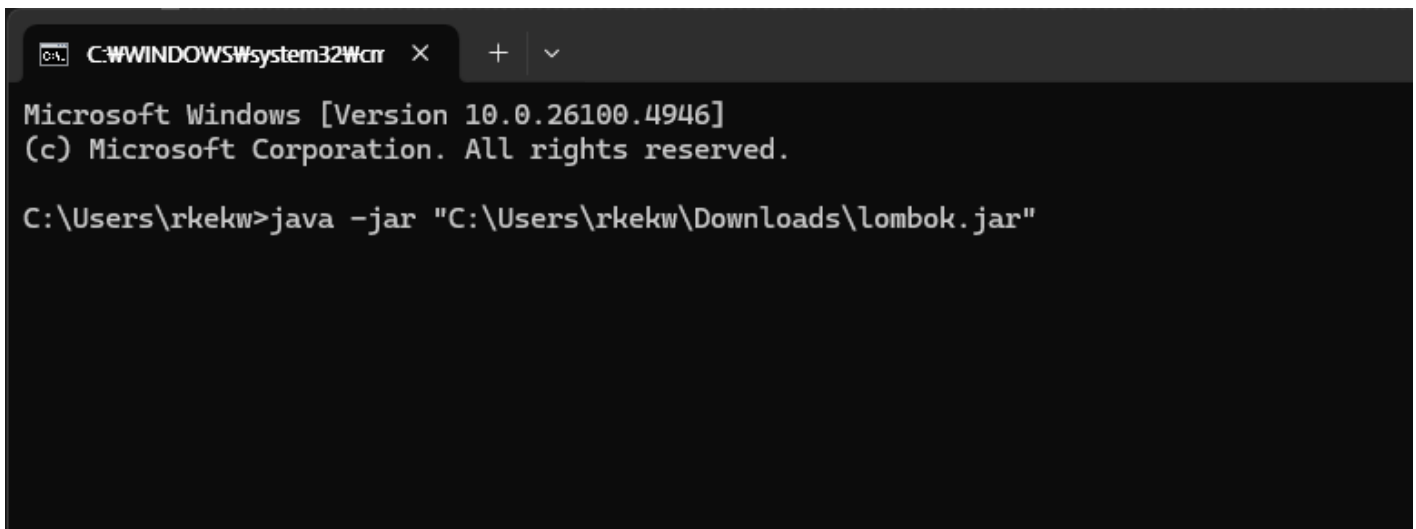
02. .jar



03. win + r [] [] [] cmd []

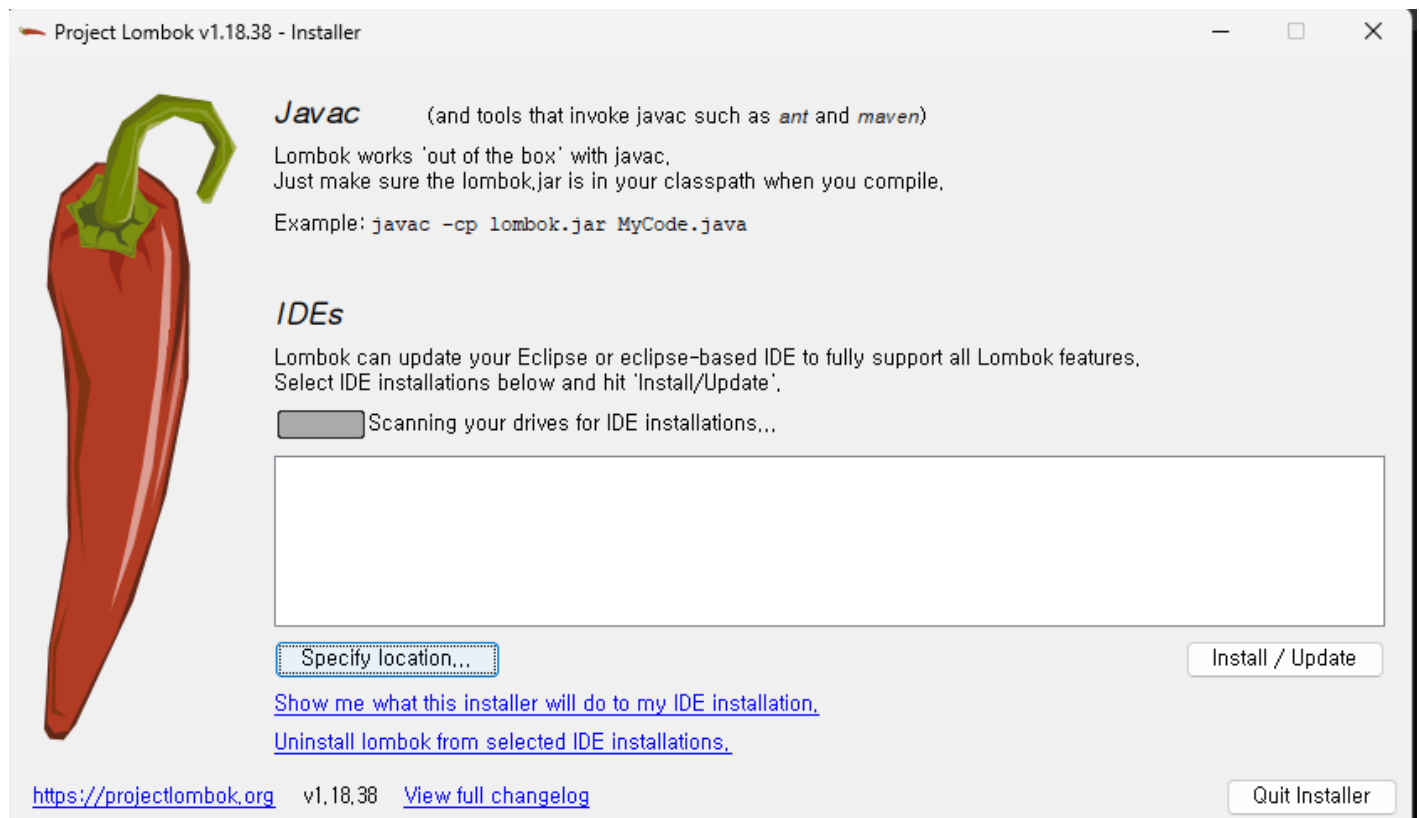


04. java -jar [] [] [] [] (cmd [] [] [] [] [] [] [] [])

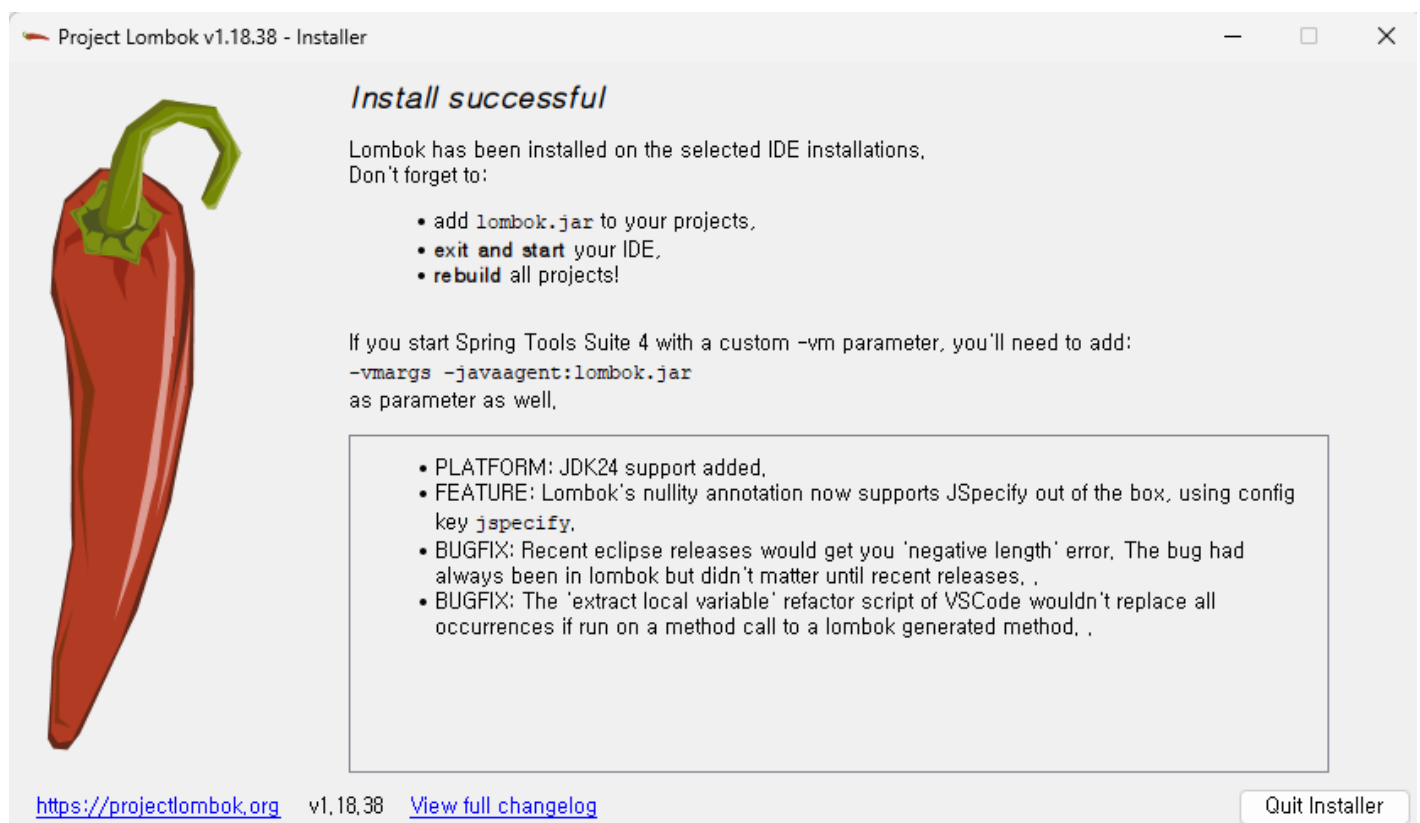


05. [] [] [] [] [] [] STS [] [] [] [] .

* 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.

```
..
```

.ini 파일을 --add-opens로 수정하여 Java 24를 사용할 수 있도록 합니다.

이제 Lombok을 사용하여 Java 24에서 IDE를 실행할 수 있습니다.

이제 (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 ini 파일을 --vmargs로 -javaagent로 수정합니다.

--vmargs로 다음 줄을 추가합니다.

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

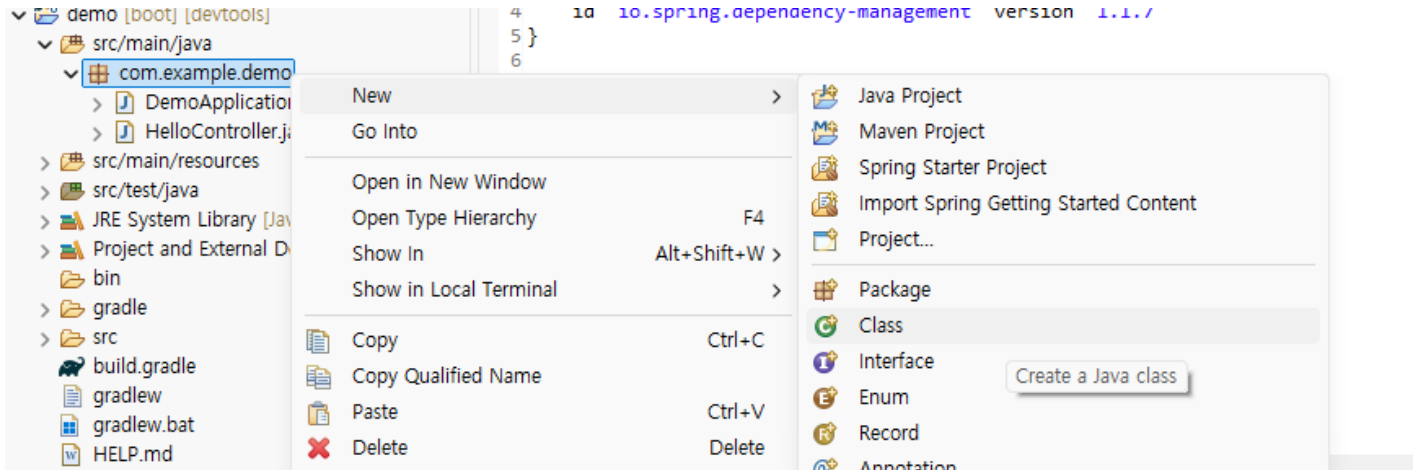
--add-opens=java.base/java.util=ALL-UNNAMED

-javaagent로 lombok.jar를 추가합니다.

예를 들어 : -javaagent:C:\Users\rkekw\Desktop\lombok.jar \2025-08-29 \01. \02. STS - sts-4.31.0.RELEASE\lombok.jar

예를 들어 : -javaagent:lombok.jar

01. com.example.demo



02.

```

package com.example.demo;

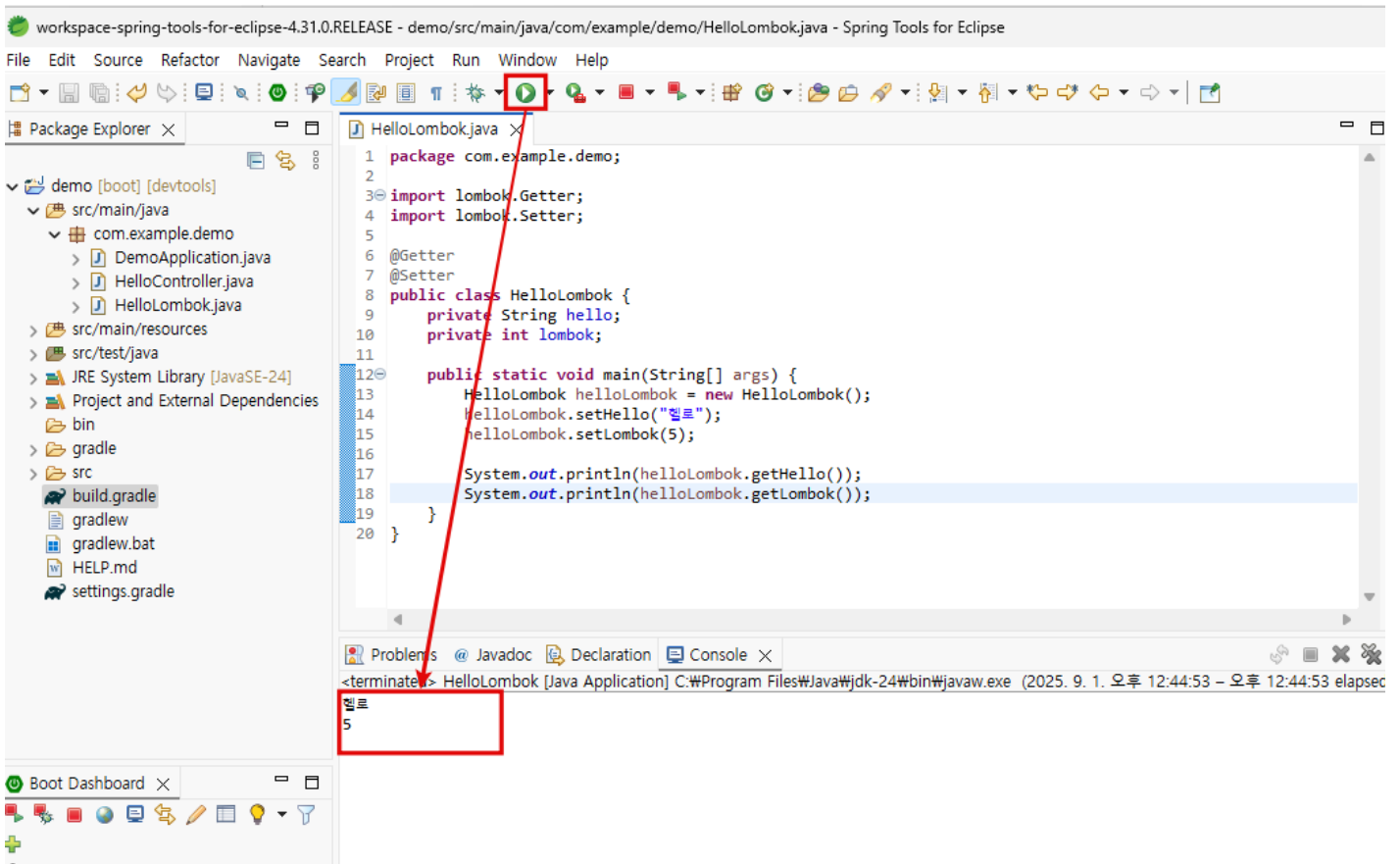
import lombok.Getter;
import lombok.Setter;

@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("Hi");
    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("Hi", 5);
        System.out.println(helloLombok.getHello());
        System.out.println(helloLombok.getLombok());
    }
}

```

@Getter  **RequiredArgsConstructor**        .

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

@RequiredArgsConstructor -        

Revision #5

Created 2025-09-01 10:28:49 UTC by minwoo

Updated 2025-09-02 12:07:18 UTC by minwoo