

# 롬복 설치 가이드

롬복은 생성자, getter, setter등을 Annotation 을 통해 자동 생성하는 plugin

IntelliJ 참조

<https://inpa.tistory.com/entry/IntelliJ-%F0%9F%92%BD->

Lombok-%EC%84%A4%EC%B9%98-%EB%B0%A9%EB%B2%95-%EC%98%A4%EB%A5%98-%ED%95%B4%EA%B2%B0

## 1. 설치

### 1.1 다운받기

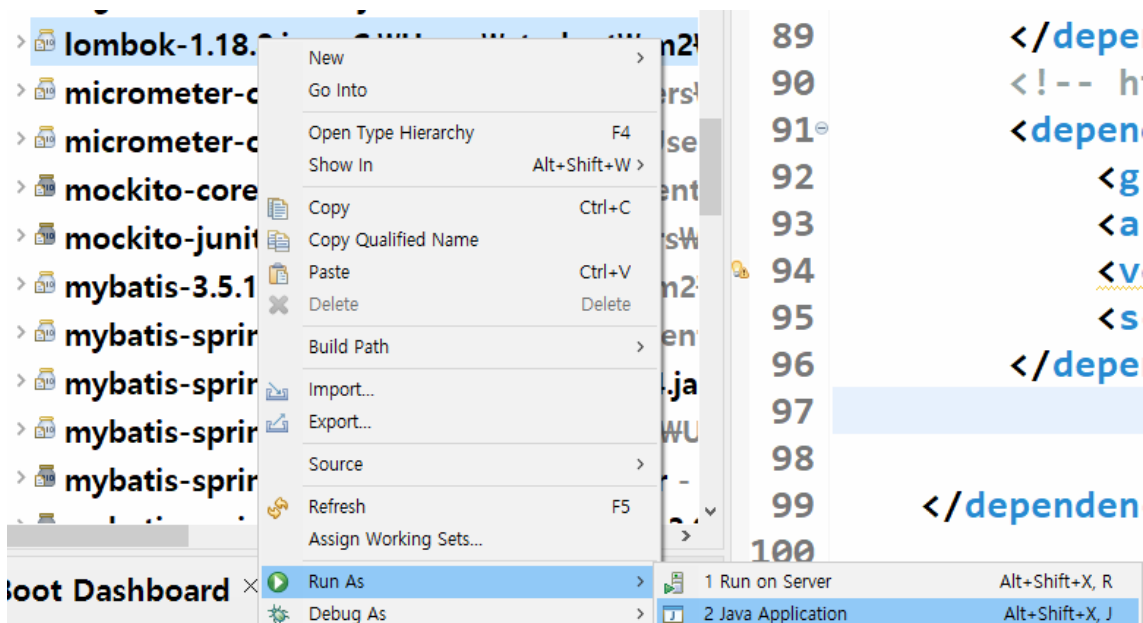
<https://mvnrepository.com/artifact/org.projectlombok/lombok>

maven repository에서 Lombok 버전을 선택해 dependency를 pom.xml에 추가한다.

```
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.36</version>
  <scope>provided</scope>
</dependency>
```

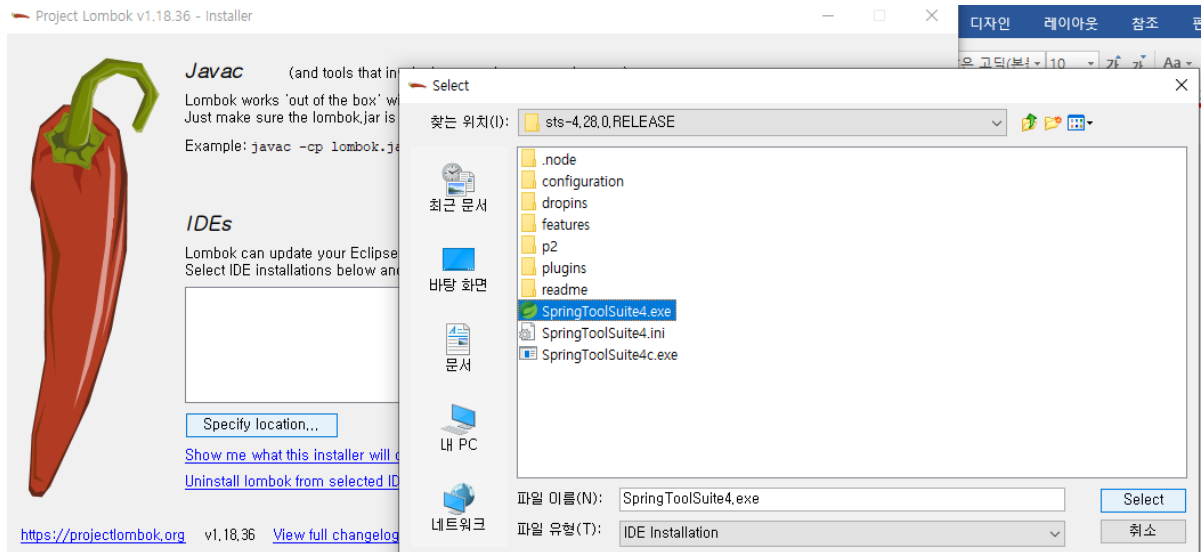
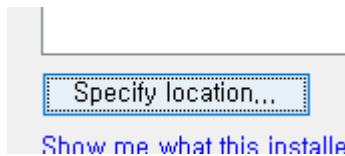
### 1.2 설치하기

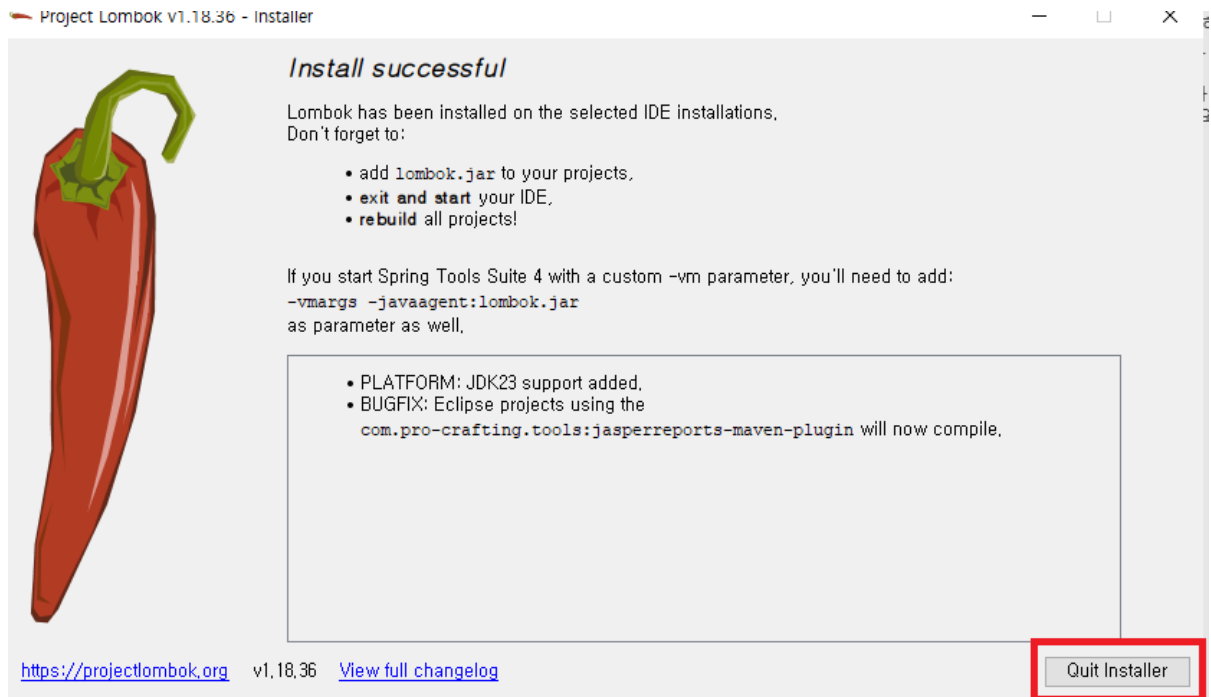
maven에서 Lombok.XXX.jar를 우클릭해서



Eclipse를 찾을 수 없다는 팝업이 뜨면 Ok 버튼을 클릭한다.

설치할 eclipse(sts)를 Specify location 버튼을 클릭해서 찾는다.





### 1.3 SpringToolSuite4.ini 파일에 Lombok.jar 설정

SpringToolSuite4.ini의 맨 아래에 추가된 Lombok 설정을 잘라내기 한다.

\*SpringToolSuite4.ini - Windows 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
--illegal-access=permit  
--add-modules=ALL-SYSTEM  
-Dfile.encoding=UTF-8
```

```
-javaagent:C:\wuplus\sts-4.28.0.RELEASE\lombok.jar
```

설정 중간에 -vmargs를 찾아 바로 아래 붙여넣기 한다.

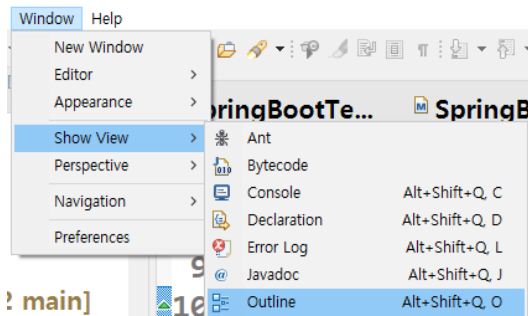
```
openFile  
-vm  
plugins/org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.  
6.v20250130-0529/jre/bin  
-vmargs  
-javaagent:C:\wuplus\sts-4.28.0.RELEASE\lombok.jar  
-XX:CompileCommand=exclude  
org.eclipse.jdt.internal.core.dom.rewrite.ASTRewriteAnalyzer::getExt  
endedRange
```

Eclips(sts)를 다시 실행한다.

## 2. Annotation 종류

Annotation	설명	비고
@Getter	code가 컴파일 될 때 getter 메서드들을 생성한다.	
@Setter	code가 컴파일 될 때 setter 메서드들을 생성한다.	
@ToString	toString() 메서드를 생성한다.	
@EqualsAndHashCode	사용 객체에 대해서 equals(), hashCode() 메서드를 생성한다.	
@NoArgsConstructor	파라미터(매개변수)가 없는 생성자를 생성한다.	DTO
@RequiredArgsConstructor	final, @NonNull이 있는 필드만 포함하여 생성자를 생성한다.	
@AllArgsConstructor	모든 필드를 파라미터(매개변수)로 갖는 생성자를 생성한다.	주입 받을 의존 객체가 있는 경우
@Data	@Getter(모든속성), @Setter(final이 붙지 않은), @ToString, @EqualsAndHashCode, @RequiredArgsConstructor 위의 어노테이션들을 합쳐둔 어노테이션이다.	
@Log	log라는 변수를 이용하여 로그 기능을 사용할 수 있다.	
@Log4j, @Slf4j	Log4J(Slf4J) 설정을 이용하여 로그 기능 사용할 수있다. 마찬가지로 log 변수를 통해 사용 한다.	
@NonNull	필드의 값이 null이 될 수 없음을 명시해준다.	

Annotation에 대한 결과는 outline(window > show view> 을 통해 볼 수 있다.



결과 확인

```

1 package com.uplus.eureka.book.model.dto;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10 @Data
11 public class Book implements Serializable {
12     private static final long serialVersionUID=1L;
13     @Schema(description = "책 일련 번호" , example = "979-11-602-00000-0")
14     private String isbn ;
15     @Schema(description = "제목" , example="스프링 부트 3" )
16     private String title ;
17     @Schema(description = "저자" , example="동글양" )
18     private String author ;
19     @Schema(description = "가격" , example="42000" )
20     private int price ;
21     @Schema(description = "책 상세 설명" , example="자바 백" )
22     private String describ ;
23     @Schema(description = "이미지" , example="979-11-602-00000-0")
24     private String img ;
25 }
26

```

com.uplus.eureka.book.model.dto

- Book
  - getIsbn() : String
  - getTitle() : String
  - getAuthor() : String
  - getPrice() : int
  - getDescrib() : String
  - getImg() : String
  - setIsbn(String) : void
  - setTitle(String) : void
  - setAuthor(String) : void
  - setPrice(int) : void
  - setDescrib(String) : void
  - setImg(String) : void
  - equals(Object) : boolean
  - canEqual(Object) : boolean
  - hashCode() : int
  - toString() : String
  - Book()
    - serialVersionUID : long
    - isbn : String
    - title : String
    - author : String
    - price : int