

# 1장. 스프링부트(SpringBoot) 개발 환경구축

*스프링부트(SpringBoot)*



Spring Boot

# 스프링 부트(Spring Boot)

- **스프링부트(Spring Boot)란?**

스프링 프레임워크의 서브 프로젝트로 만들어 졌으며, 이름에서 알 수 있듯 스프링과 부트의 합성어다.

부트(Boot)는 “컴퓨터를 부팅한다”는 말처럼 시스템을 사용 가능한 상태로 만드는 것을 의미하며, 웹 애플리케이션을 빠르게 제작하여 출시할 수 있는 경량의 오픈소스 프레임 워크이다.

2014년 쯤 출시되었으며 점차 스프링에 비해 사용이 늘어나고 있는 추세이다.

- **스프링 부트의 특징**

- ① **라이브러리 관리 자동화**

기존 자바프로젝트에서는 메이븐(Maven)이나 그레이들(Gradle)을 이용해서 라이브러리 의 의존성을 관리해왔으나, 스프링 부트에서는 스타터(Starter)라는 것을 이용해 특정 기능에 필요한 라이브러리 의존성을 더욱 간단히 처리할 수 있음



# 스프링 부트(Spring Boot)

## ② 설정의 자동화

스프링부트에서는 프로젝트에 추가된 라이브러리를 기반으로 실행에 필요한 환경을 자동으로 설정해 준다. 라이브러리를 인지해 관련된 스프링 설정을 자동으로 처리해준다.

## ③ 테스트 환경과 내장 서버(톰캣)

스프링부트로 생성한 프로젝트에는 Junit을 비롯한 테스트 관련 라이브러리들이 기본적으로 포함되어 있다. 또한 톰캣(Tomcat) 서버를 내장하고 있어서 main() 메소드를 가진 클래스를 실행하는 방식으로 구동하기 때문에 빠르게 실행 결과를 볼 수 있다.

## ④ 독립적으로 실행 가능한 JAR

애플리케이션을 실제 운영 서버에 배포하기 위해서 패키징(Packaging)을 해야하는데, WAR 파일이 아닌 JAR 파일로 패키징할 수 있다.






# 개발 환경 구축

## ◆ JDK 1.8 설치

### Java SE Development Kit 8u202

This software is licensed under the [Oracle Binary Code License Agreement for Java SE Platform Products](#)

Solaris x64	85.38 MB	 <a href="#">jdk-8u202-solaris-x64.tar.gz</a>
Windows x86	201.64 MB	 <a href="#">jdk-8u202-windows-i586.exe</a>
Windows x64	211.58 MB	 <a href="#">jdk-8u202-windows-x64.exe</a>



# 이클립스(Eclipse) IDE 설치

## ◆ 이클립스(Eclipse) 설치

2021년 3월 버전 -> Workspace(작업공간) 설정 : C:\SpringWorks

### MORE DOWNLOADS

- Other builds
- Eclipse 2022-06 (4.24)
- Eclipse 2022-03 (4.23)
- Eclipse 2021-12 (4.22)
- Eclipse 2021-09 (4.21)
- Eclipse 2021-06 (4.20)
- Eclipse 2021-03 (4.19)
- Eclipse 2020-12 (4.18)
- Eclipse 2020-09 (4.17)
- Older Versions

### Eclipse IDE 2021-03 R Packages



#### Eclipse IDE for Java Developers

328 MB 399,895 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration



Windows x86\_64  
macOS x86\_64  
Linux x86\_64 | AArch64



#### Eclipse IDE for Enterprise Java and Web Developers

519 MB 397,458 DOWNLOADS

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.



Windows x86\_64  
macOS x86\_64  
Linux x86\_64 | AArch64

[Click here](#) to file a bug against Eclipse Web Tools Platform.  
[Click here](#) to file a bug against Eclipse Platform.  
[Click here](#) to file a bug against Maven integration for web projects.  
[Click here](#) to report an issue against Eclipse Wild Web Developer (incubating).

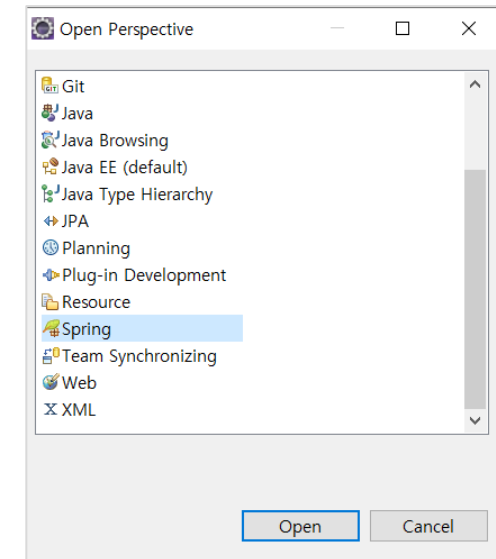
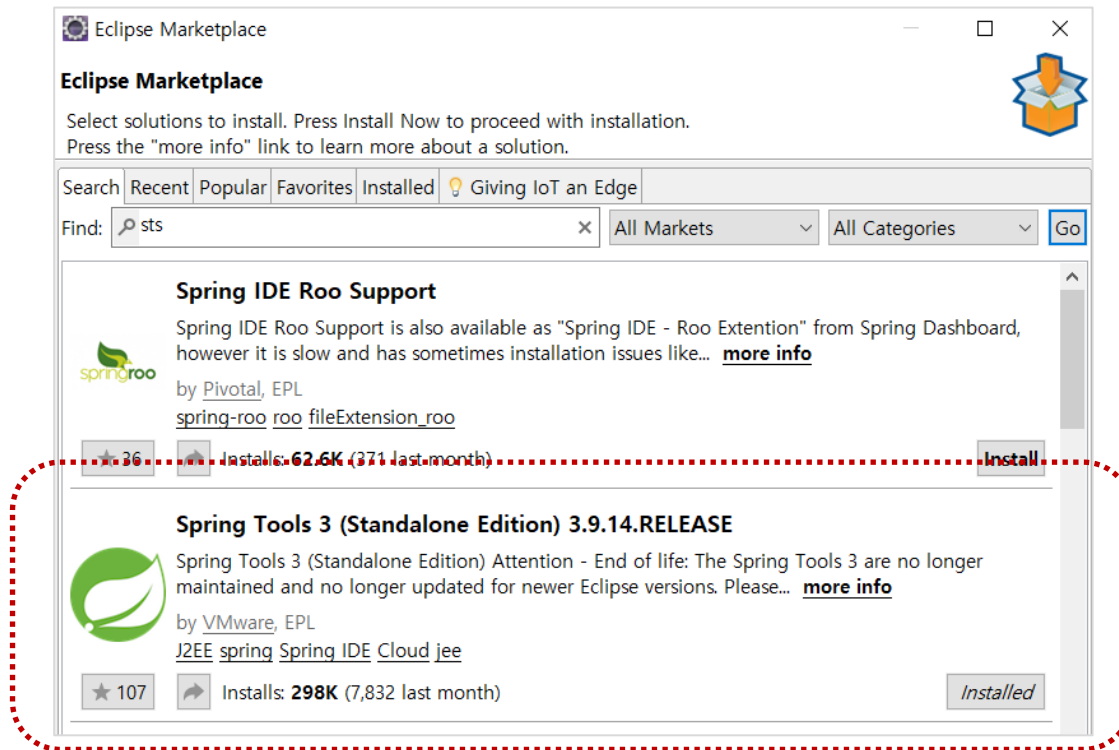


# STS(Spring Tool Suite)

- STS(Spring Tool Suite) 플러그인 설치

이클립스 –[Help] – [Eclipse Marketplace] – sts 검색

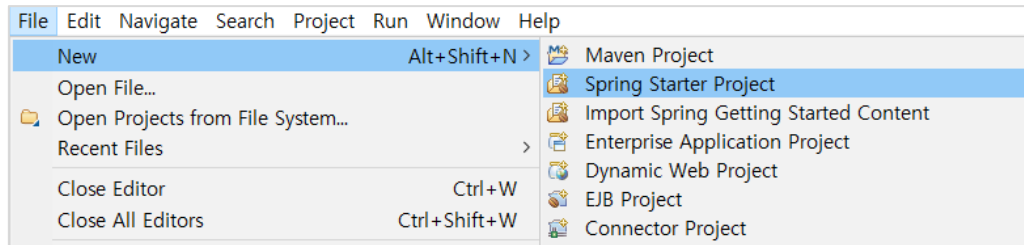
## Spring Tools 3(Standalone Edition) 3.9.14 RELEASE - install



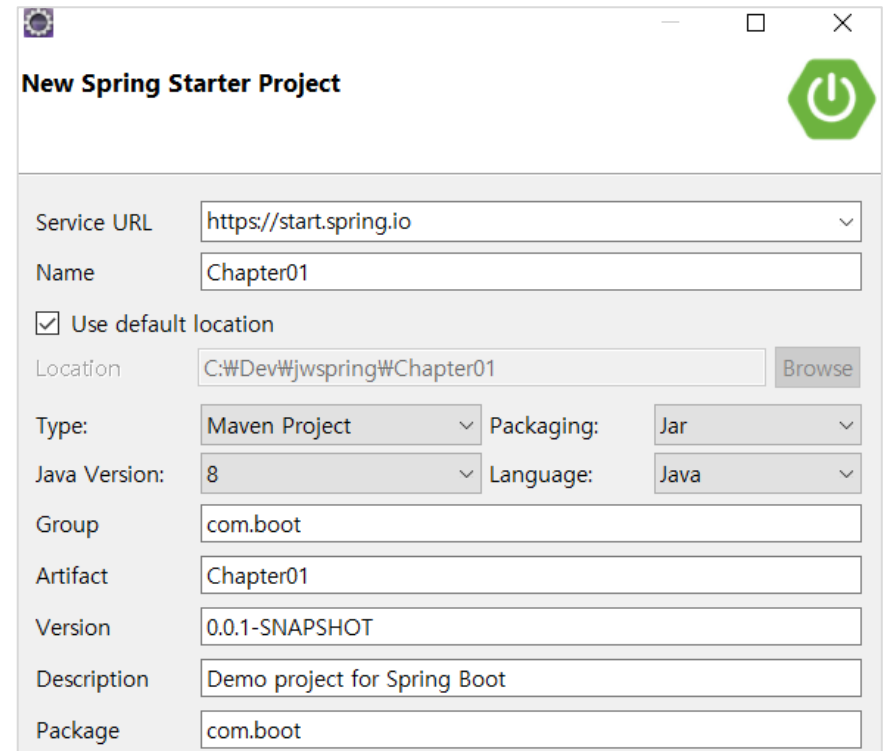
# 스프링 부트(Spring Boot)

- 실습 프로젝트 생성

[File] -> [New] -> [Spring Starter Project]



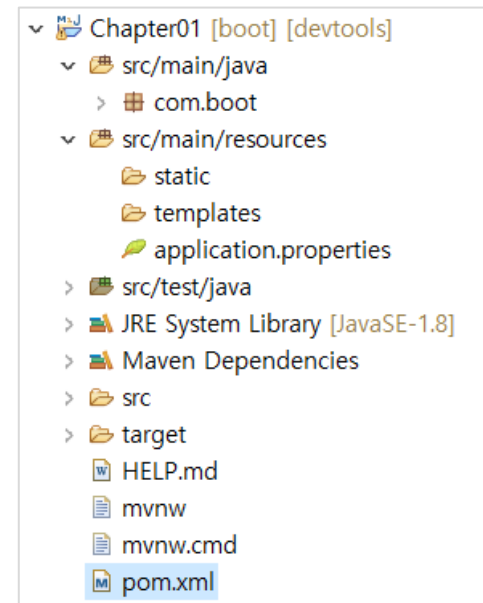
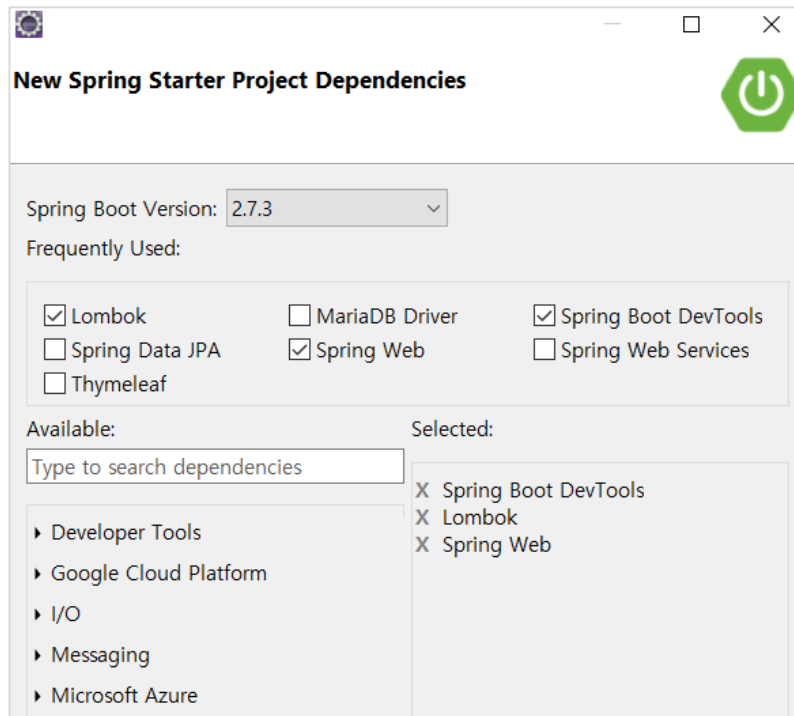
Name – Chapter01,  
Packing – Jar  
Java Version – 8  
Group – com.boot,  
Package – com.boot



# 스프링 부트(Spring Boot)

- 실습 프로젝트 생성

Lombok, Spring Web, Spring Boot DevTools 모듈 추가함





# 스프링 부트(Spring Boot)

- pom.xml

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

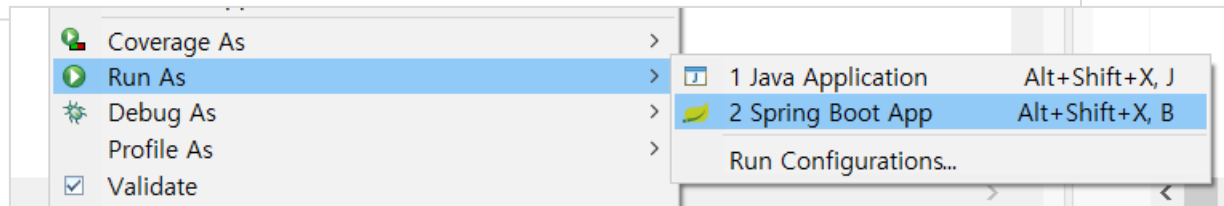


# 스프링 부트(Spring Boot)

- SpringBoot App으로 실행

```
@SpringBootApplication
public class Chapter01Application {

    public static void main(String[] args) {
        //Spring Boot App으로 실행
        SpringApplication.run(Chapter01Application.class, args);
    }
}
```

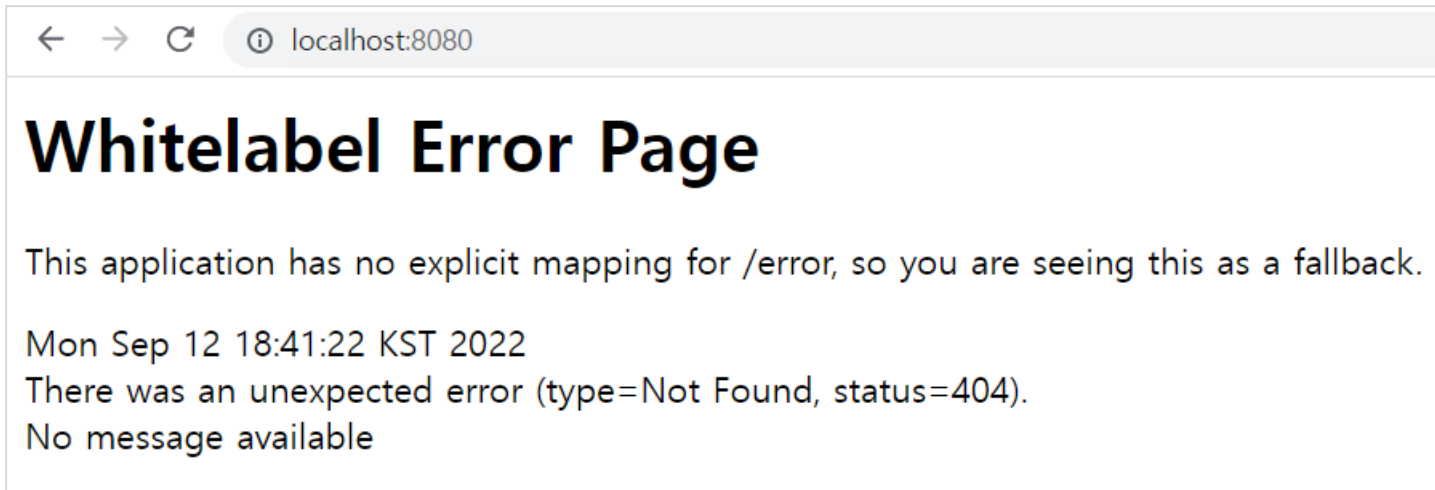


```
restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.65]
restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in
```



# 스프링 부트(Spring Boot)

- <http://localhost:8080>



# 스프링 부트(Spring Boot)

- Java Application과 톰캣 서버로 실행

```
@SpringBootApplication
public class Chapter01Application {

    public static void main(String[] args) {
        //SpringApplication.run(Chapter01Application.class, args); //Spring Boot App으로 실행

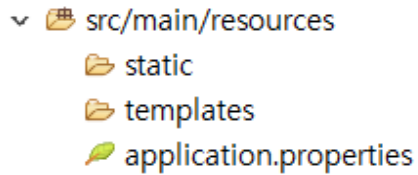
        SpringApplication application =
            new SpringApplication(Chapter01Application.class);
        //application.setWebApplicationType(WebApplicationType.NONE); //자바 애플리케이션으로 실행하기
        application.setWebApplicationType(WebApplicationType.SERVLET); //Spring Boot App으로 실행
        application.run(args);
    }
}
```



# 스프링 부트(Spring Boot)

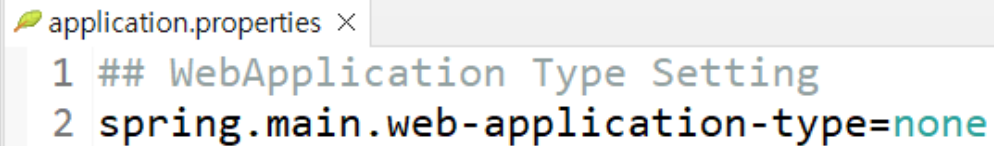
- 프로퍼티(property) 파일 사용하기

application.properties(프로퍼티 파일)은 전체 프로젝트의 프로퍼티 정보를 관리하는 설정 파일이다.



```
src/main/resources
├── static
├── templates
└── application.properties
```

web 서버로 실행 안됨



```
application.properties ×
1 ## WebApplication Type Setting
2 spring.main.web-application-type=none
```



# 스프링 부트(Spring Boot)

- 포트 번호 변경하기

```
spring.main.web-application-type=servlet

## Server Setting(포트번호 변경)
server.port = 8181
```

```
SpringApplication application =
    new SpringApplication(Chapter01Application.class);
//application.setWebApplicationType(WebApplicationType.NONE); //Java 어플 실행
application.setWebApplicationType(WebApplicationType.SERVLET); //톰캣 서버. Web 실행
application.run(args);
```

← → ↻ ⓘ localhost:8181

## Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Mon Sep 12 20:54:22 KST 2022

There was an unexpected error (type=Not Found, status=404).

No message available



# 스프링 부트(Spring Boot)

## @RestController 사용하기

JSP같은 View를 별도로 만들지 않는 대신에 컨트롤러 메서드가 리턴한 데이터 자체를 클라이언트로 전달한다.

클라이언트에 전달되는 데이터는 대부분 문자열이거나 VO(Value Object)나 컬렉션 형태의 자바 객체인데, 자바 객체가 전달되는 경우에는 자동으로 JSON으로 변환하여 처리하게 됨

(스프링에서는 jackson-databind를 pom.xml에 등록해야 됨)

```
> jackson-databind-2.13.3.jar - C:\Users\Wkiyon\m2\repository\com\facebook\jackson\jackson-databind\2.13.3\jackson-databind-2.13.3.jar
> jackson-annotations-2.13.3.jar - C:\Users\Wkiyon\m2\repository\com\facebook\jackson\jackson-annotations\2.13.3\jackson-annotations-2.13.3.jar
> jackson-core-2.13.3.jar - C:\Users\Wkiyon\m2\repository\com\facebook\jackson\jackson-core\2.13.3\jackson-core-2.13.3.jar
```



# 스프링 부트(Spring Boot)

## @RestController 사용하기

```
@RestController //view(jsp)를 만들지 않고 문자열이 그대로 출력됨
public class BoardController {

    public BoardController() {
        System.out.println("==> BoardController 생성");
    }

    @GetMapping("/hello")
    public String hello(String name) {
        return "Hello : " + name;
    }
}
```

← → ↺ localhost:8181/hello?name=장그래

Hello : 장그래





# 스프링 부트(Spring Boot)

## @RestController 빈(Been) 등록

스프링에서는 xml에서 <component-sacan>으로 빈(객체)을 생성하는데,  
스프링부트는 @SpringBootApplication내의 @ComponentScan이 처리함.

```
@SpringBootApplication
public class Chapter01Application {
```

```
    @AliasFor(annotation = ComponentScan.class, attribute = "basePackageClasses")
    Class<?>[] scanBasePackageClasses() default {};
```



# 스프링 부트(Spring Boot)

- VO 객체 리턴하기

컨트롤러 메서드가 단순히 문자열을 리턴하지 않고 VO(Value Object) 객체를 리턴하는 경우에는 VO 객체를 JSON 데이터로 변환하여 응답 프로토콜 바디에 출력한다.

```
package com.boot.domain;

import java.util.Date;

@Getter
@Setter
public class BoardVO {
    private int seq;
    private String title;
    private String writer;
    private String content;
    private Date createDate;
    private int cnt;
}
```

```
//vo 객체 리턴
@GetMapping("/getBoard")
public BoardVO getBoard() {
    BoardVO board = new BoardVO();
    board.setSeq(1);
    board.setTitle("테스트 제목...");
    board.setWriter("테스터");
    board.setContent("테스트 내용입니다...");
    board.setCreateDate(new Date());
    board.setCnt(0);
    return board;
}
```

← → ↺ ⓘ localhost:8181/getBoard

{"seq":1,"title":"테스트 제목...","writer":"테스터","content":"테스트 내용입니다."}



# 스프링 부트(Spring Boot)

- 컬렉션(리스트) 객체 리턴하기

```
//컬렉션(리스트) 리턴
@GetMapping("/getBoardList")
public List<BoardVO> getBoardList(){
    List<BoardVO> boardList = new ArrayList<>();
    for(int i=1; i<=10; i++) {
        BoardVO board = new BoardVO();
        board.setSeq(i);
        board.setTitle("제목" + i);
        board.setWriter("테스터");
        board.setContent(i + "번 내용입니다.");
        board.setCreateDate(new Date());
        board.setCnt(0);
        boardList.add(board);
    }
    return boardList;
}
```

← → ↺ ⓘ localhost:8181/getBoardList

```
[{"seq":1,"title":"제목1","writer":"테스터","content":"1번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":2,"title":"제목2","writer":"테스터","content":"2번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":3,"title":"제목3","writer":"테스터","content":"3번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":4,"title":"제목4","writer":"테스터","content":"4번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":5,"title":"제목5","writer":"테스터","content":"5번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":6,"title":"제목6","writer":"테스터","content":"6번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":7,"title":"제목7","writer":"테스터","content":"7번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":8,"title":"제목8","writer":"테스터","content":"8번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":9,"title":"제목9","writer":"테스터","content":"9번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}, {"seq":10,"title":"제목10","writer":"테스터","content":"10번 내용입니다.","createDate":"2022-09-12T12:50:09.534+00:00","cnt":0}]
```



# 스프링 DevTools

- 스프링 DevTools 사용하기

프로젝트 생성시 체크한 DevTools는 컨트롤러(java)를 수정할 때마다 매번 애플리케이션을 중지, 재실행하지 않고도 사용할 수 있게 해준다.

```
restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http
restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0
restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplication
restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization
```



# Lombok 라이브러리

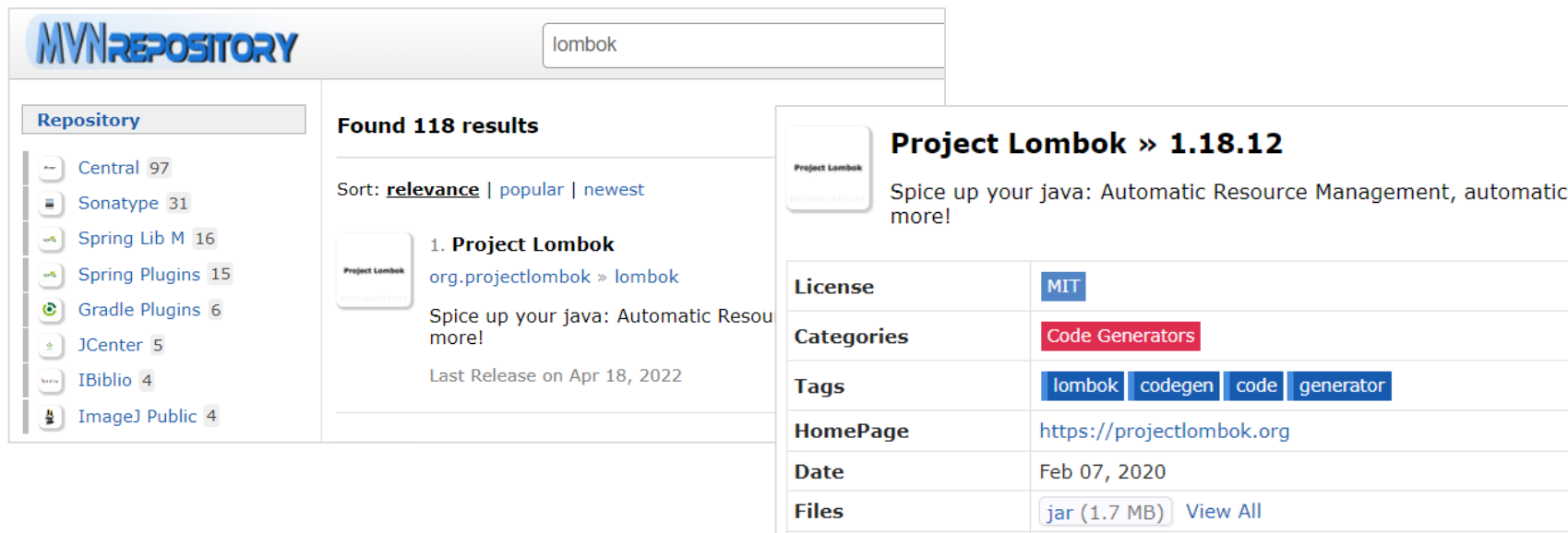
## ◆ Lombok(롬복)

Java 개발시 사용하는 VO클래스의 생성자, getter/setter, toString()을 자동으로 생성해 준다.


pom.xml에 <dependency>를 추가하면 자동으로 jar파일이 다운로드 된다.

그리고 별도로 이클립스 설치 폴더에 롬복 라이브러리를 추가해야 한다.

MVNRepository > lombok 검색 > 1.18.12 jar 다운로드



The screenshot shows the Maven Repository website with a search for 'lombok'. The search results show 118 results, sorted by relevance. The first result is 'Project Lombok' version 1.18.12, published on Feb 07, 2020. The project description is 'Spice up your java: Automatic Resource Management, automatic more!'. The license is MIT, and the categories are Code Generators. The tags are lombok, codegen, code, and generator. The homepage is https://projectlombok.org. The file size is 1.7 MB.

Repository	Found 118 results	Sort: <a href="#">relevance</a>   <a href="#">popular</a>   <a href="#">newest</a>
<a href="#">Central</a> 97	 <b>1. Project Lombok</b>	
<a href="#">Sonatype</a> 31	<a href="#">org.projectlombok » lombok</a>	
<a href="#">Spring Lib M</a> 16	Spice up your java: Automatic Resou more!	
<a href="#">Spring Plugins</a> 15	Last Release on Apr 18, 2022	
<a href="#">Gradle Plugins</a> 6		
<a href="#">JCenter</a> 5		
<a href="#">IBiblio</a> 4		
<a href="#">ImageJ Public</a> 4		

Project Lombok » 1.18.12	
Spice up your java: Automatic Resource Management, automatic more!	
License	<a href="#">MIT</a>
Categories	<a href="#">Code Generators</a>
Tags	<a href="#">lombok</a> <a href="#">codegen</a> <a href="#">code</a> <a href="#">generator</a>
HomePage	<a href="https://projectlombok.org">https://projectlombok.org</a>
Date	Feb 07, 2020
Files	<a href="#">jar (1.7 MB)</a> <a href="#">View All</a>



# Lombok 라이브러리

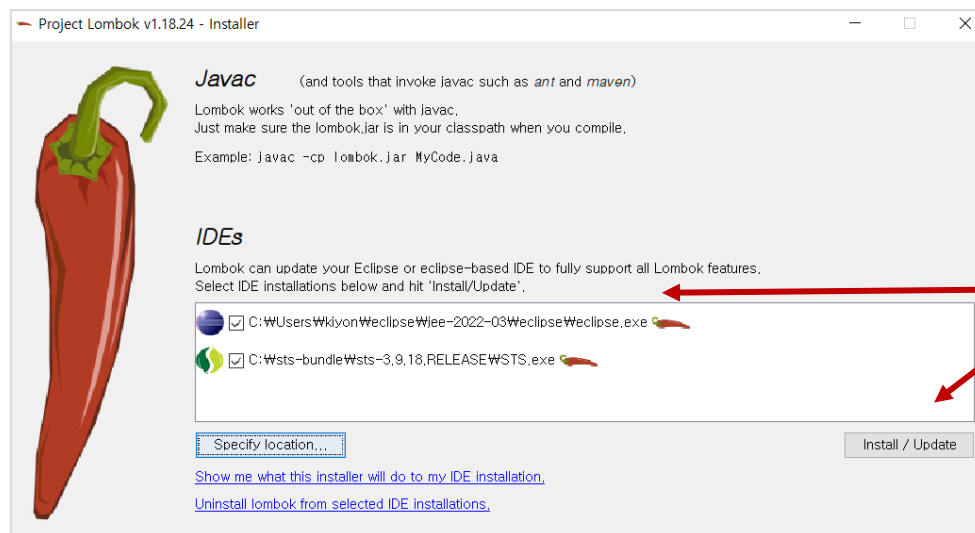
## ◆ Lombok(롬복)

lombok.jar파일을 명령 프롬프트에서 실행함.

 lombok.jar	2022-05-01 오후 7:34	ALZip JAR File	1,926KB
--	--------------------	----------------	---------

cmd(명령프롬프트)에서 실행

```
c:\WDev>java -jar lombok.jar
```



이클립스에 체크후  
Install 클릭

