3장. DI – Junit 테스트

spring

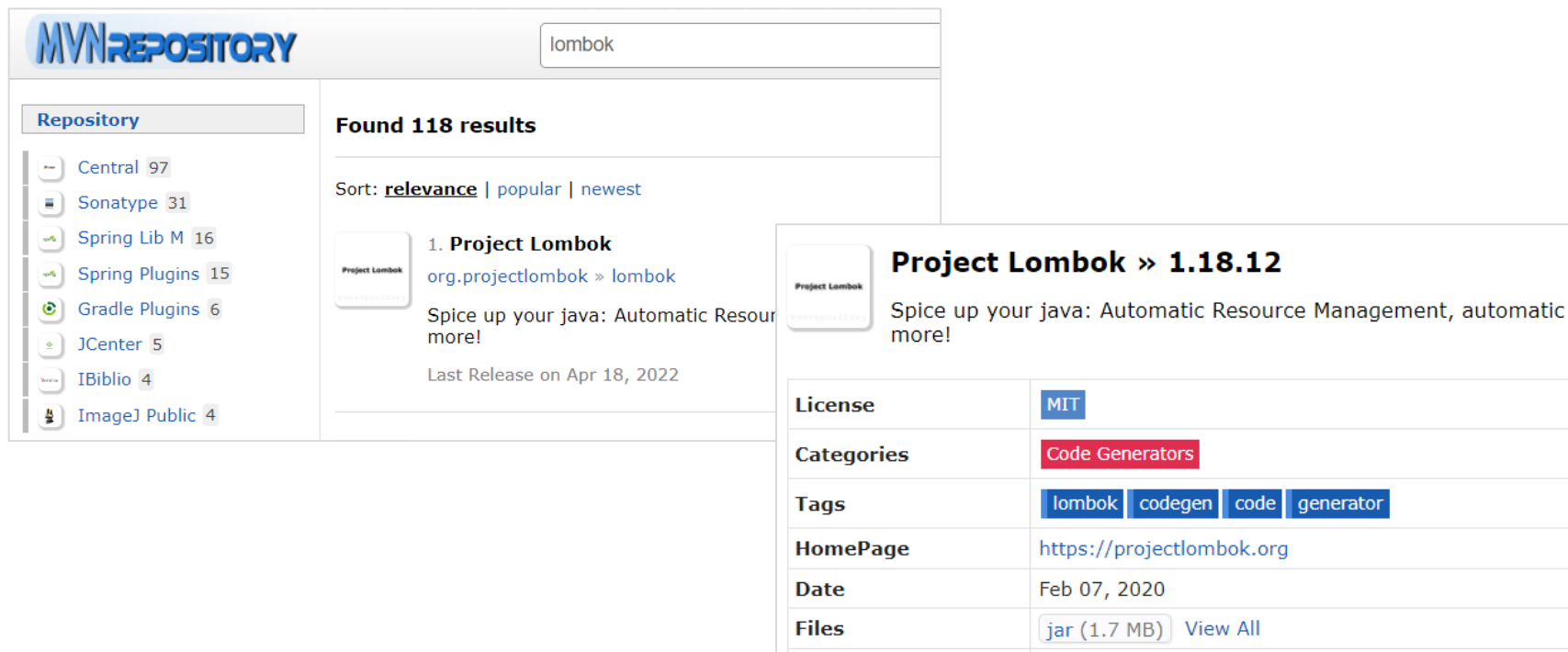# Lombok 라이브러리

◆ Lombok(롬복)

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

MVNRepository >lombok 검색 > 1.18.12 jar 다운로드

Lombok 라이브러리

◆ Lombok(롬복)

lombok.jar파일을 명령 프롬프트에서 실행함.

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

```
c:₩Dev>java -jar lombok.jar
```
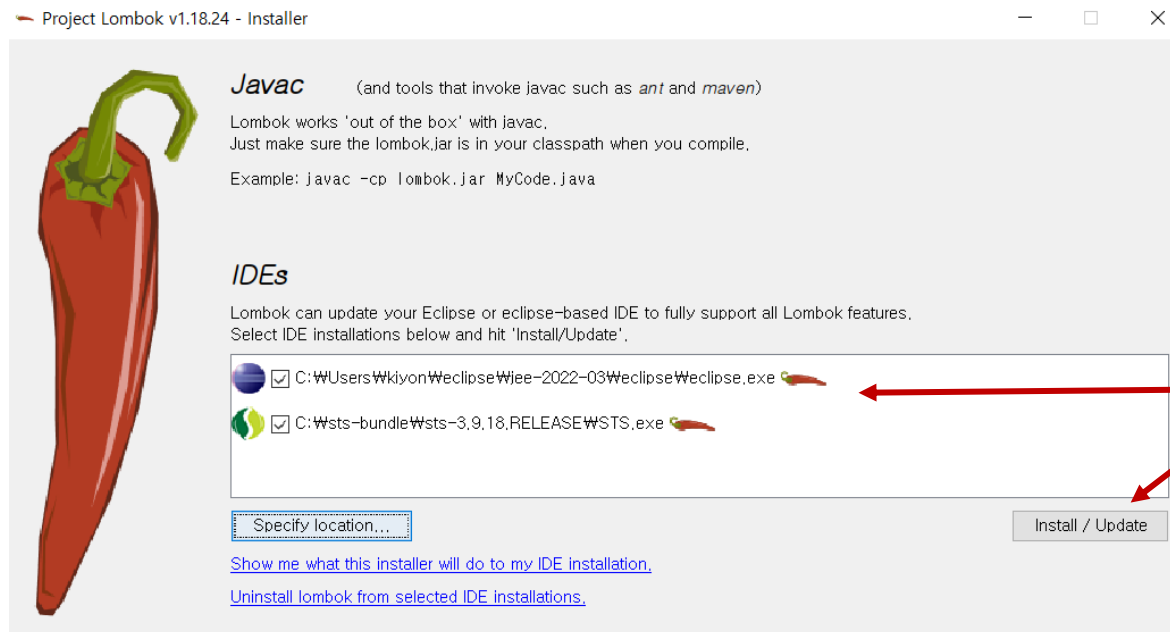

이클립스에 체크후
Install 클릭

Lombok 라이브러리

◆ **Lombok(롬복)**

pom.xml 에 등록


```
Maven   Gradle   Gradle (Short)   Gradle (Kotlin)   SBT   Ivy   Grape
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.12</version>
    <scope>provided</scope>
</dependency>
```

의존성 주입 테스트

◆ **Junit 테스트**

테스트를 위해 라이브러리 pom.xml에 추가하기

```xml
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${org.springframework-version}</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.12</version>
    <scope>provided</scope>
</dependency>
```
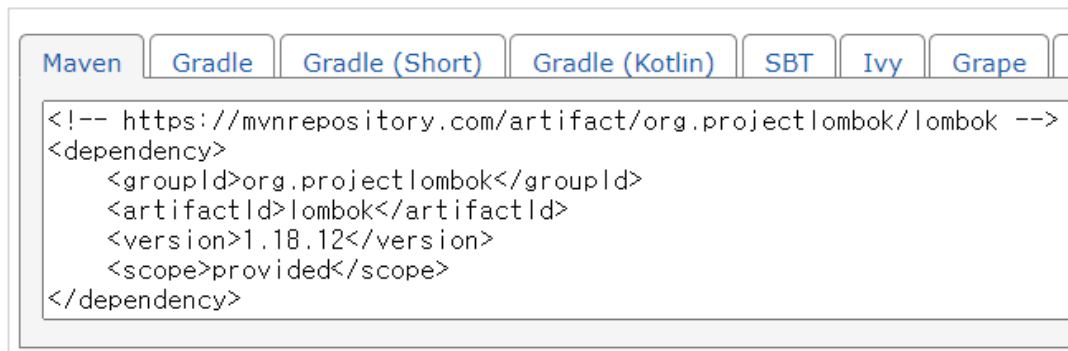
의존성 주입 테스트

◆ **Junit 테스트**

Junit 4.7에서 4.12로 update 등록

```xml
<!-- Test -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
```

Log 기록을 위해 1.2.17로 업데이트 등록

```xml
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
```

의존성 주입 테스트
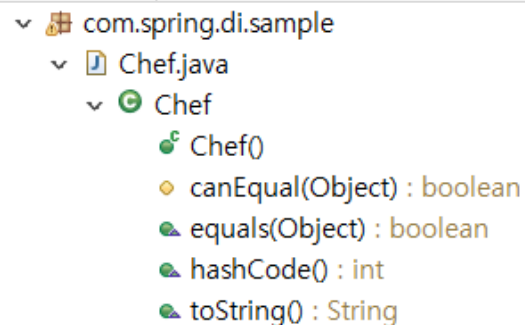
◆ 예제

```java
package com.spring.di.sample;

import org.springframework.stereotype.Component;

import lombok.Data;

@Data
@Component
public class Chef {   //Chef(주방장) 클래스

}
```

```
∨ com.spring.di.sample
    ∨ Chef.java
        ∨ Chef
            Chef()
            canEqual(Object) : boolean
            equals(Object) : boolean
            hashCode() : int
            toString() : String
```

의존성 주입 테스트

◆ 예제

```java
//생성자 주입 - @AllArgsConstructor
//Setter 주입
//필드 주입 - Autowired

@AllArgsConstructor
@Data
@Component
public class Restaurant {

    //@Autowired
    //@Setter(onMethod_ = @Autowired)
    private Chef chef;

}
```

```
∨ 📄 Restaurant.java
  ∨ 🟢 Restaurant
      ▪ chef
      ⚷ Restaurant(Chef)
      ○ canEqual(Object) : boolean
      ▲ equals(Object) : boolean
      ● getChef() : Chef
      ▲ hashCode() : int
      ● setChef(Chef) : void
      ▲ toString() : String
```

의존성 주입 테스트

◆ 예제

```java
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
@Log4j
public class SampleTest {

    @Autowired
    private Restaurant restaurant;

    @Test
    public void test() {
        System.out.println("test1...");
        Log.info(restaurant);
    }
}
```

```
INFO : org.springframework.test.context.support.DefaultTestContextBootstrapper
INFO : org.springframework.test.context.support.DefaultTestContextBootstrapper
test1...
INFO : com.spring.di.sample.SampleTest - Restaurant(chef=Chef())
```

의존성 주입 테스트

◆ 예제

Chef Bean 객체를 주입 하지 않은 경우

```java
@Data
@Component
public class Restaurant {

    //@Autowired
    //@Setter(onMethod_ = @Autowired)
    private Chef chef;
}
```

```
test1...
INFO : com.spring.di.sample.SampleTest - Restaurant(chef=null)
```

의존성 주입 테스트

◆ 예제

Hotel 클래스 생성

```java
//@RequiredArgsConstructor
@Component
@ToString
public class Hotel {

    //private final Chef chef;  //@RequiredArgsConstructor를 주입할 경우

    private Chef chef;

    public Hotel(Chef chef) { //생성자를 직접 만든 경우
        super();
        this.chef = chef;
    }
}
```

의존성 주입 테스트

◆ 예제

```java
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
@Log4j
public class SampleTest {

    @Autowired
    private Restaurant restaurant;

    @Autowired
    private Hotel hotel;

    @Test
    public void testHotel() {
        Log.info(hotel);
    }
}
```

```
INFO : com.spring.di.sample.SampleTest - Hotel(chef=Chef())
```
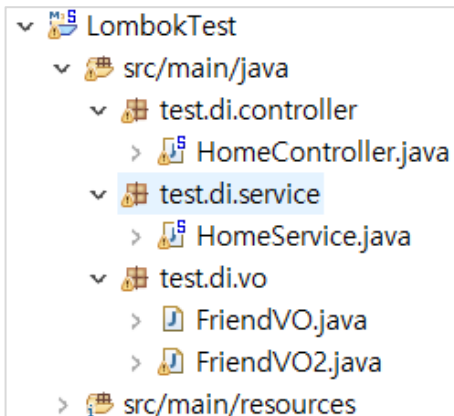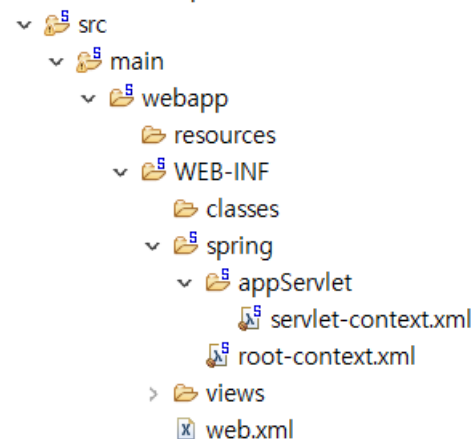
의존성 주입 테스트

◆ **Lombok – DI 실습**

servlet-context.xml 설정

```xml
<!-- Resolves views selected for rendering by @Controllers to .
<beans:bean class="org.springframework.web.servlet.view.Interna
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>

<context:component-scan base-package="test.di" />
```

의존성 주입 테스트

◆ **Lombok – DI 실습**

```java
package test.di.service;

import org.springframework.stereotype.Service;

@Service
public class HomeService {

    public void sayHello() {
        System.out.println("안녕하세요~");
    }
}
```

의존성 주입 테스트

◆ **Lombok – DI 실습**

Hello world!

The time on the server is 2022? 8? 20? ?? 10? 12? 14? KST.

```java
@Controller
@Slf4j
@RequiredArgsConstructor
public class HomeController {

    //private static final Logger logger = LoggerFactory.getLogger(Home(

    /*@Autowired    //필드 주입
    private HomeService homeService;*/

    private final HomeService homeService;   //생성자 주입

    /*@Autowired  //setter 주입
    public void setHomeService(HomeService homeService) {
        this.homeService = homeService;
    }*/
```

의존성 주입 테스트

◆ **Lombok – DI 실습**

```java
@RequestMapping(value = "/", method = RequestMethod.GET)
public String home(Locale locale, Model model) {
    Log.info("Welcome home! The client locale is {}.", locale);

    //시간 설정 - home.jsp
    Date date = new Date();
    DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, D

    String formattedDate = dateFormat.format(date);

    model.addAttribute("serverTime", formattedDate );

    //sayHello() 메서드 호출
    homeService.sayHello();

    return "home";
}
```

```
INFO: 이름이 []인 컨텍스트를 다시 로드하는 것을 완료했습니다.
INFO : test.di.controller.HomeController - Welcome home! The client locale is ko_KR.
안녕하세요~
```

의존성 주입 테스트

◆ **Setter 주입**

```java
@Getter
@Setter
public class FriendVO {
    private String name;
    private int age;

    /*
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }*/
}
```

의존성 주입 테스트

◆ 생성자 주입

```java
@Getter
@AllArgsConstructor
public class FriendVO2 {
    private String name;
    private int age;

    /*public FriendVO2(String name, int age) {
        super();
        this.name = name;
        this.age = age;
    }*/
}
```

의존성 주입 테스트

◆ **Lombok – DI 실습**

```java
@Service
public class HomeService {

    public void sayHello() {
        //System.out.println("안녕하세요~");
        FriendVO vo1 = new FriendVO();
        vo1.setName("안산");
        vo1.setAge(21);

        FriendVO2 vo2 = new FriendVO2("손흥민", 30);

        System.out.println(vo1.getName() + "님 안녕하세요~");
        System.out.println(vo2.getName() + "님 안녕하세요~");
    }
}
```

```
INFO : test.di.controller.HomeController - Welcome home!
안산님  안녕하세요~
손흥민님  안녕하세요~
```