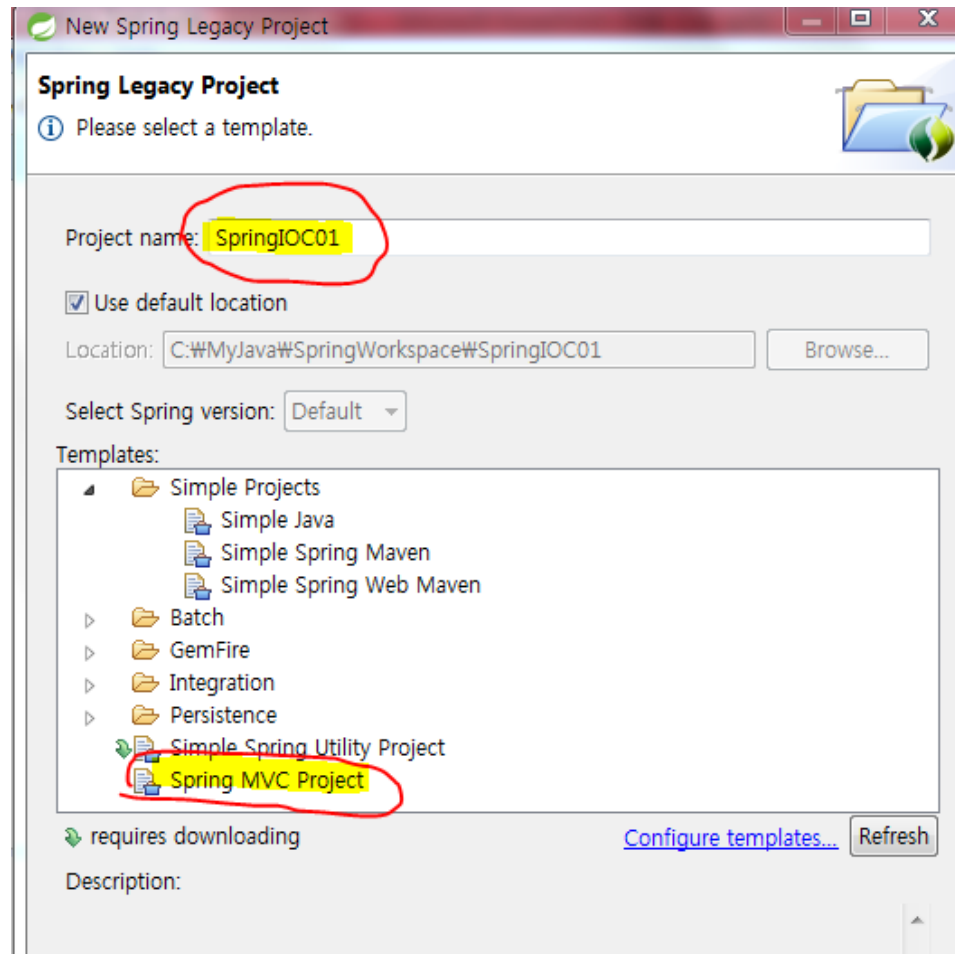


Spring Framework설정

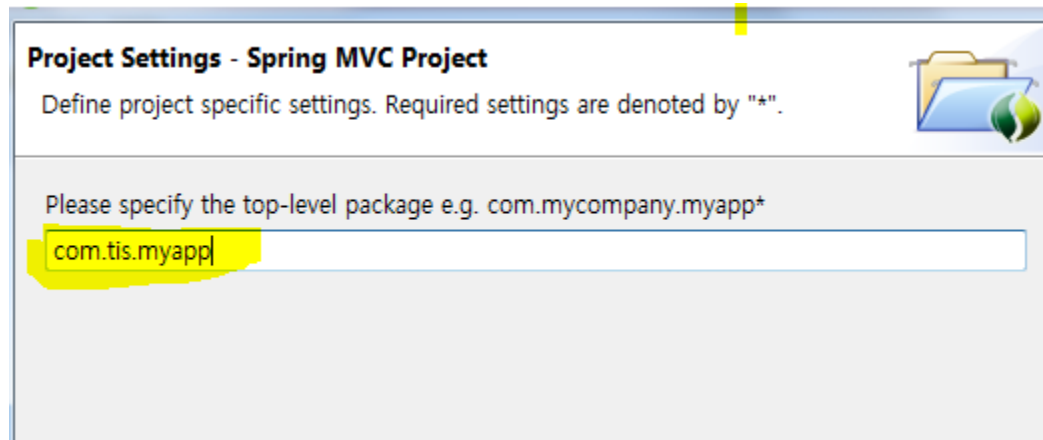
2019-01월 백성애

Spring legacy project 생성

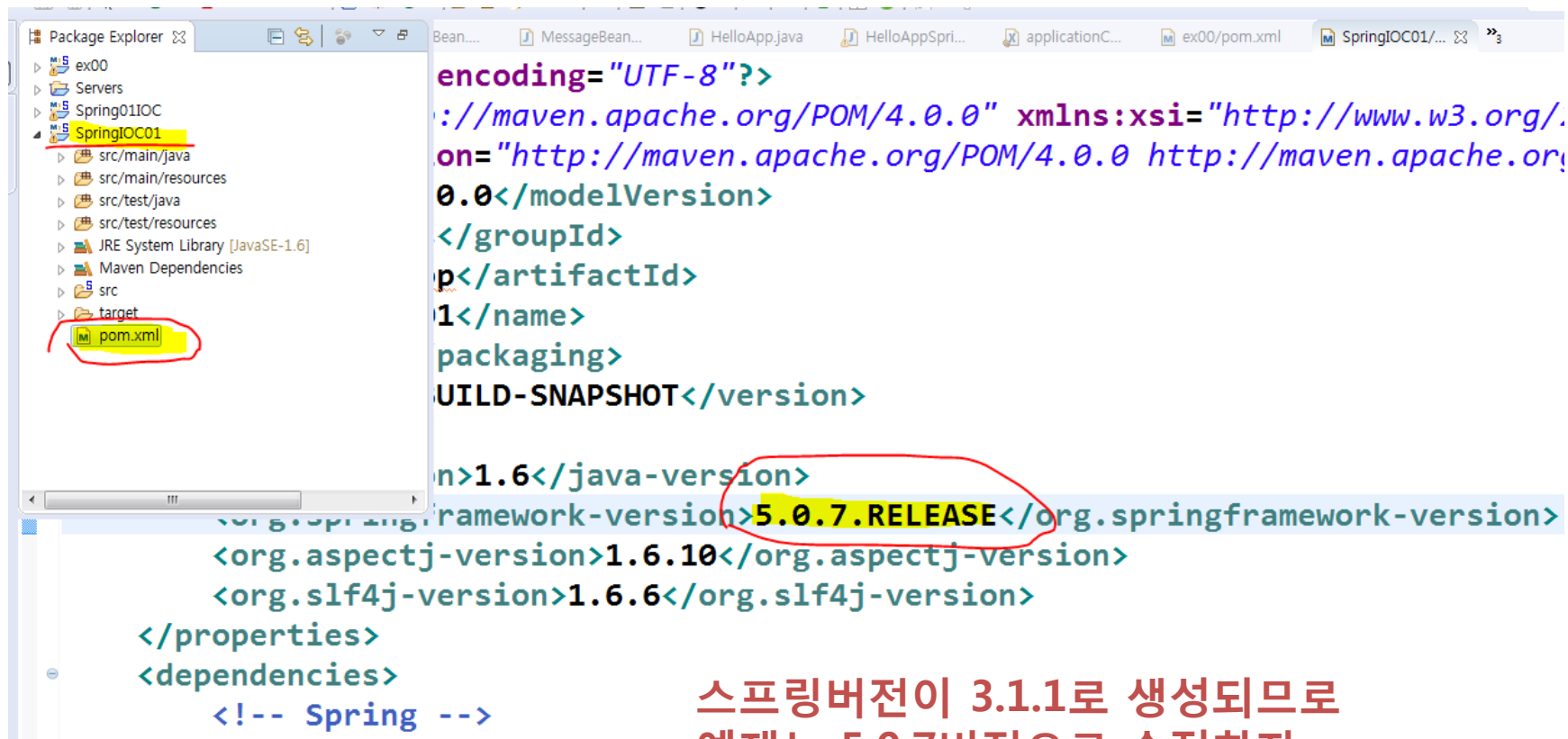


패키지명 기술 (3단계 이상으로)

com.tis.myapp이라는 패키지를 생성하자.



Pom.xml 스프링 Framework 버전 5.0.7로 수정



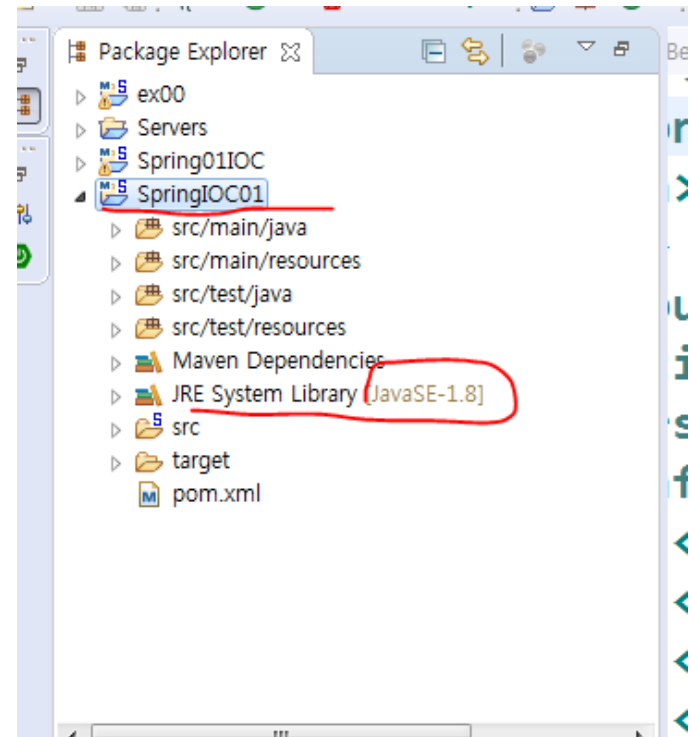
스프링 버전이 3.1.1로 생성되므로
예제는 5.0.7 버전으로 수정하자

Java Version을 1.8로 변경

- 생성된 프로젝트의 jre system library를 보면 JavaSE-1.6버전으로 생성되어 있다.
- 스프링 5버전을 이용하고자 하면 JDK1.8을 사용하는 것이 좋다. 따라서 **pom.xml**의 **<plugin>** 태그 중 **maven-compiler-plugin**의 내용은 1.6에서 **1.8로 수정**하자.

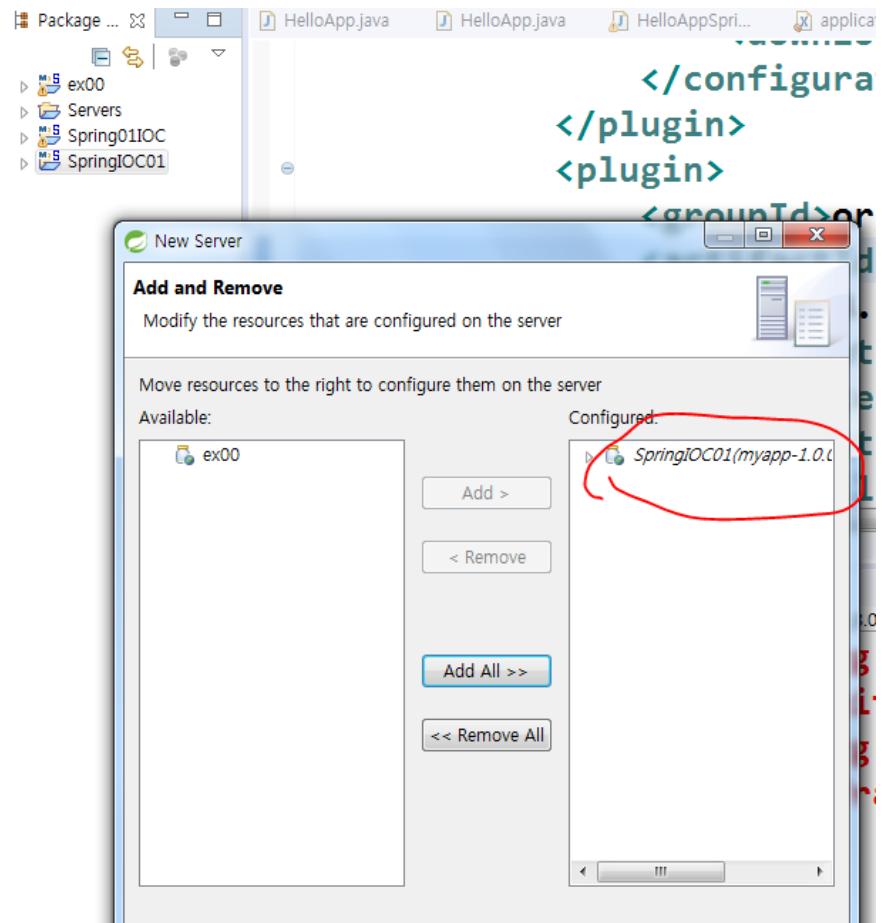
Maven>Update Project를 실행

- 프로젝트를 선택한 상태에서 **Maven>Update Project**를 실행하다.
- 이를 실행하면 컴파일이나 실행환경이 JDK1.8을 기준으로 설정된 것을 확인할 수 있다.



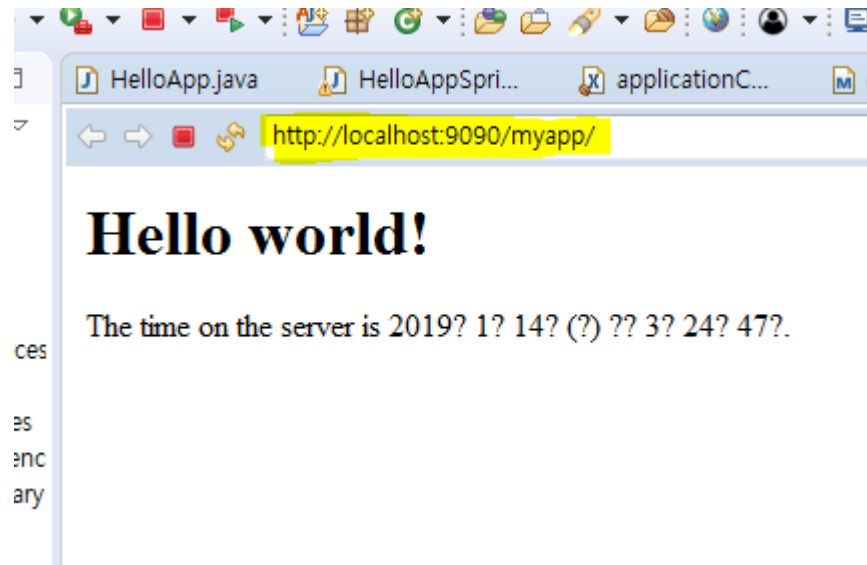
Tomcat에 deploy

- 프로젝트의 Run As> Run on Server를 이용해 실행시키자.



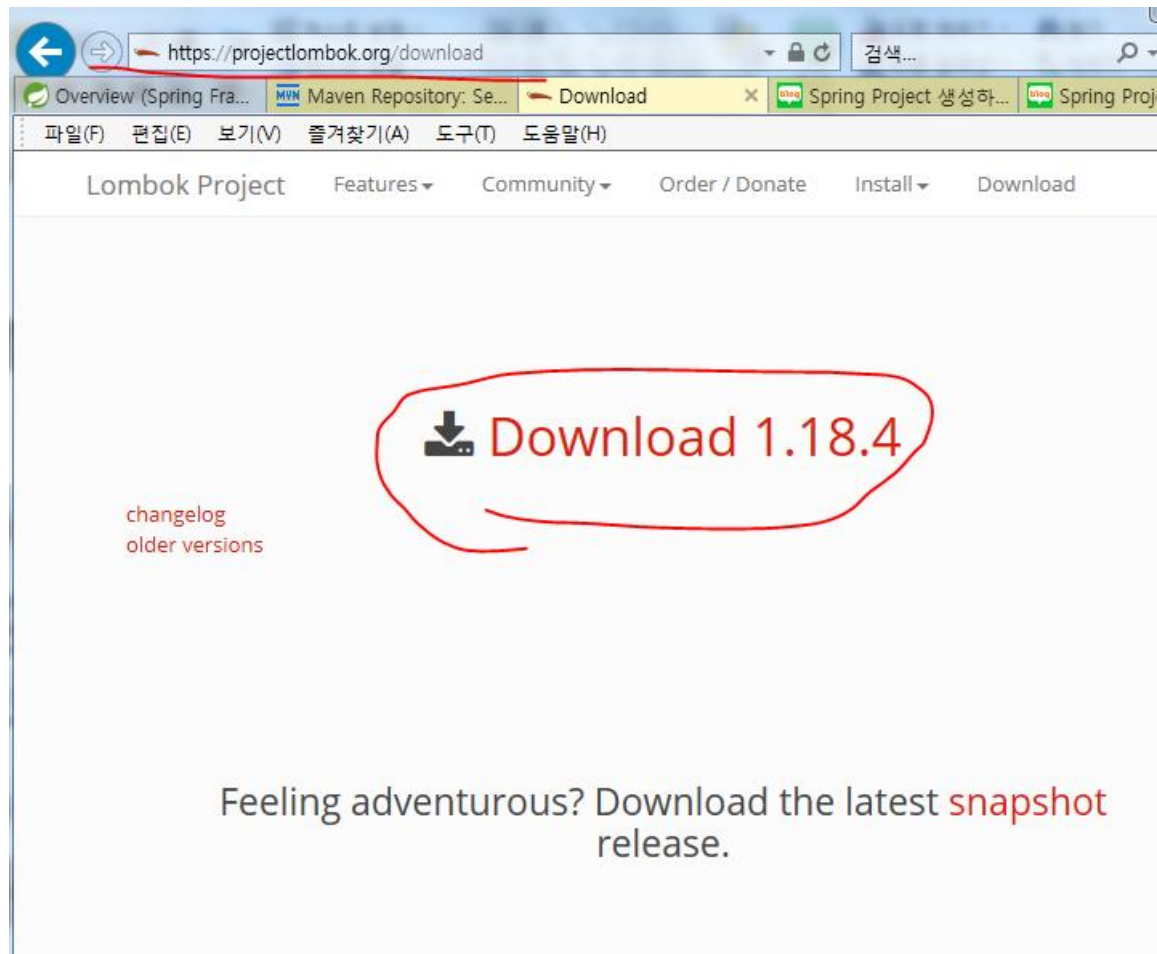
웹브라우저 통해 실행 확인

- 프로젝트를 실행하면 <http://localhost:9090/myapp/>라는 url이 실행된다.

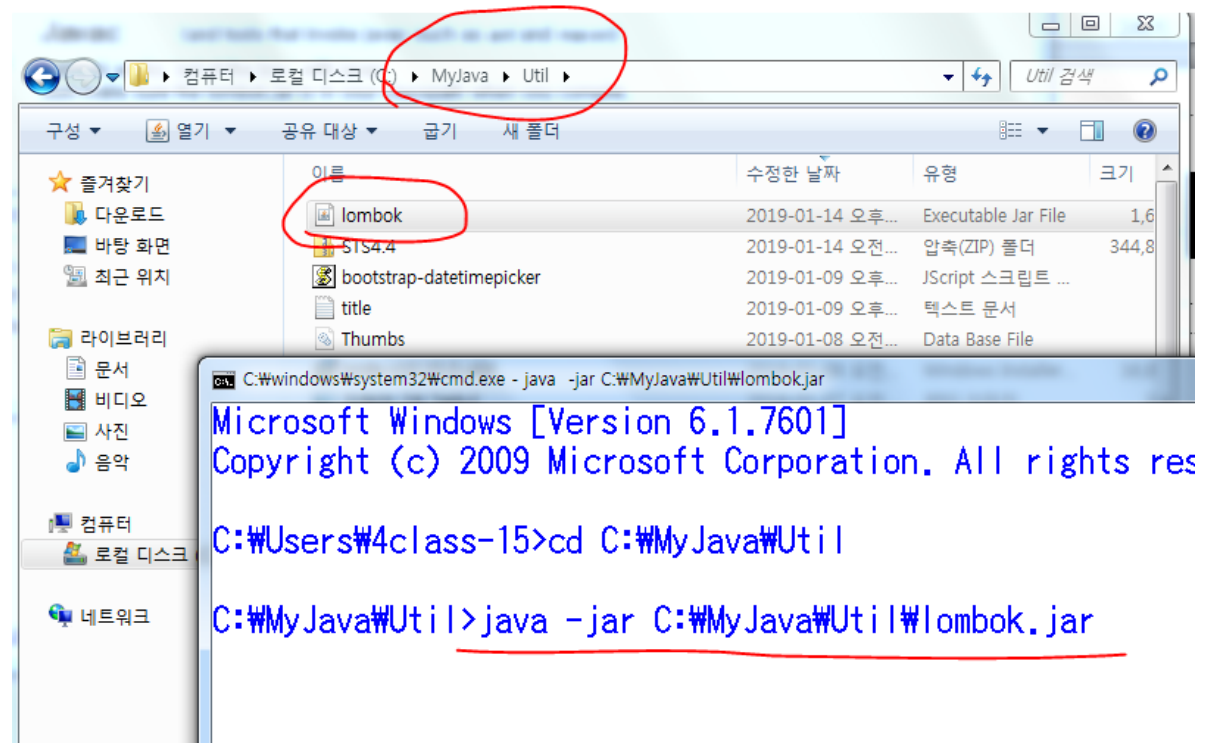


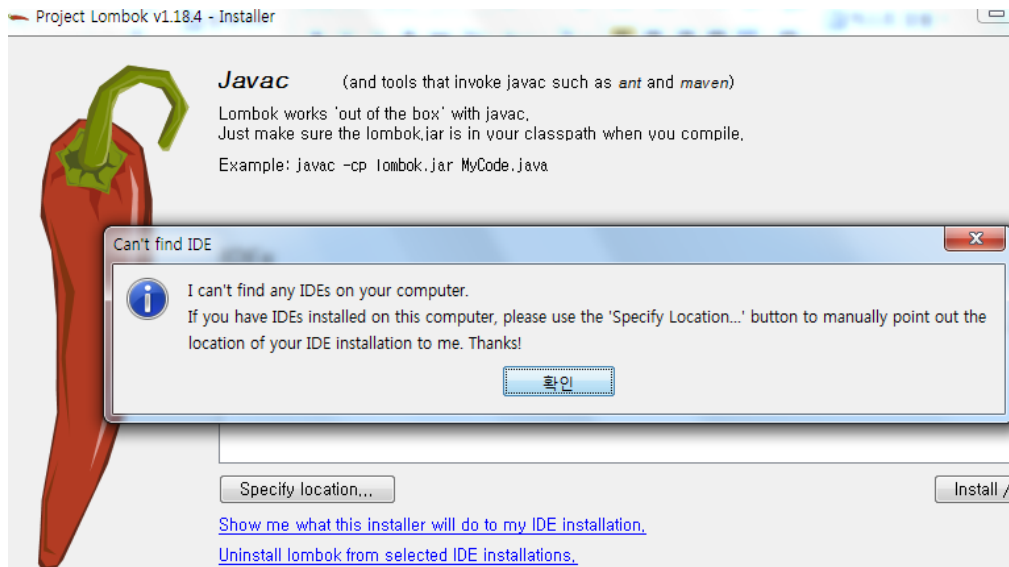
Lombok라이브러리 설치

- Lombok라이브러리를 사용하면 setter/getter, toString(), Constructor등을 자동으로 생성해주므로 편리하다.
- Lombok은 Eclipse에디터내에서도 사용되어야 하기 때문에 별도로 설치하자.
- 다운로드:
- <https://projectlombok.org/download>
- 위 경로에서 jar파일을 다운로드 받는다.

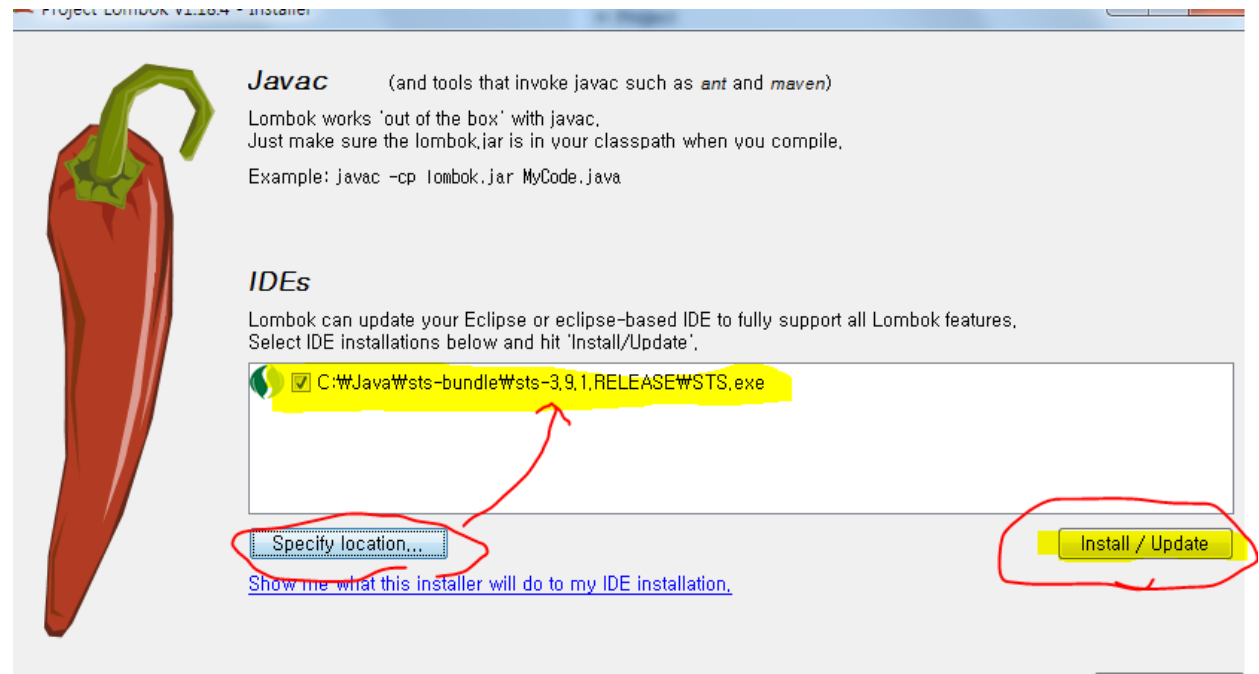


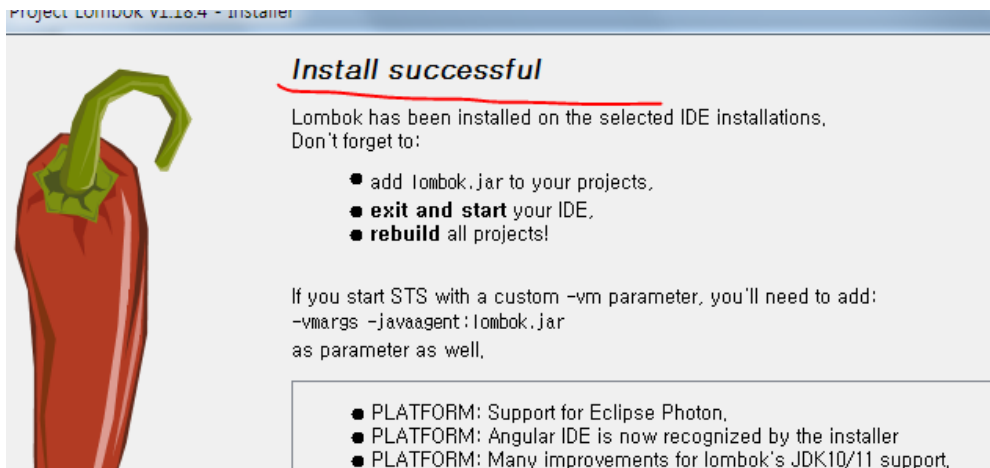
다운로드된
lombok.jar파일
이 있는 곳에서
도스창을 열고 오
른쪽 과 같이 실행
시키자.



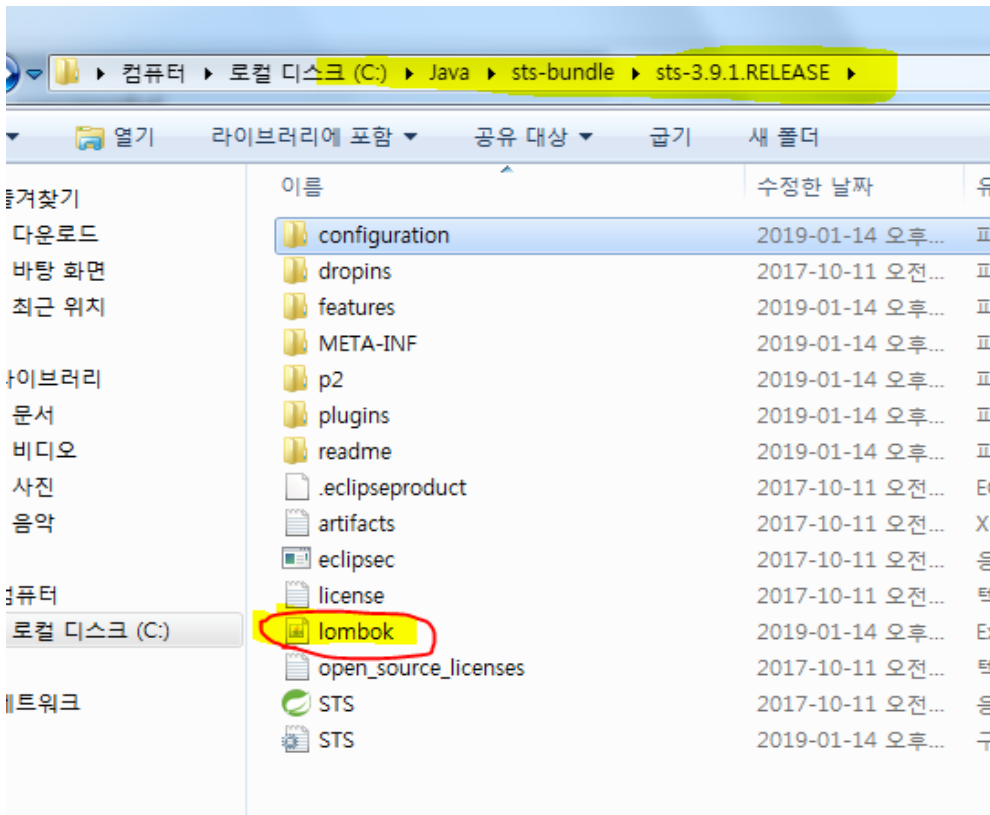


필요한 IDE를 선택하라고 나오면 Eclipse(STS)나 STS설치 경로를 찾아서 아래와 같이 설정해주자.





설치가 완료되면 STS 실행 경로에 lombok.jar 파일이 추가된 것을 확인할 수 있다.



Pom.xml 추가 설정

```
File Edit Source Navigate Search Project Run Window Help
Restaurant.java SpringIOC01/... SampleTests... appContext3.xml Dao.java DaoOracle.java DaoMySQL.java appC...

</dependency>

<!-- Test -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
<!-- Add Lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.0</version>
  <scope>provided</scope>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <artifactId>maven-eclipse-plugin</artifactId>
```

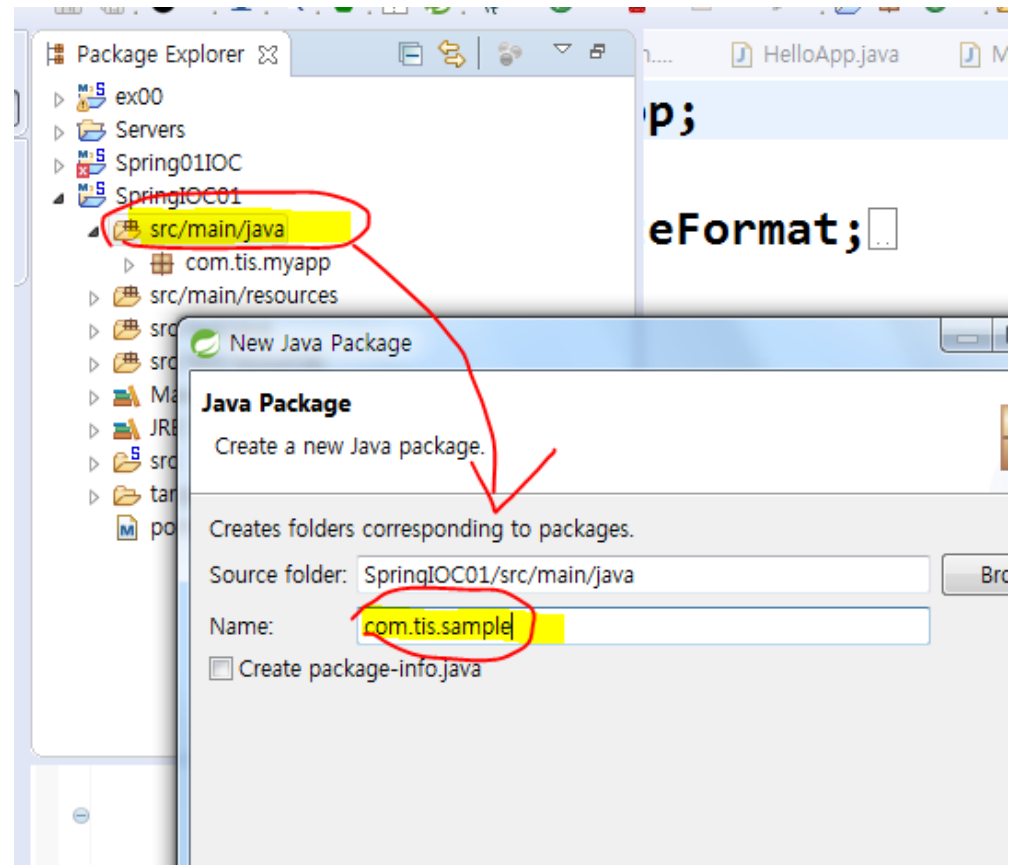
Lombok라이브러리 설정 추가와 함께 단위 테스트를 진행하기 위해 junit은 4.10이상을 사용해야 하므로 버전이 그 이하일 경우는 4.1이상으로 수정하자.

우리는 4.12로 사용할 예정입니다

예제 클래스 작성

새로운 패키지를 생성하자.

com.tis.sample패키지를 생성해서 Chef클래스와 Restaurant클래스를 생성하자.



Chef.java 파일 작성

```
package Spring01IOC/pom.xml>.sample;

import org.springframework.stereotype.Component;

import lombok.Data;

@Component
@Data

//setter/getter와 생성자, toString()
//을 자동으로 생성하도록 @Data 어노테이션을 붙인다.
public class Chef {

}
```


Restaurant.java파일 작성

@Setter를 붙이면 자동으로 setChef()를 컴파일시 생성한다. 생성된 코드는 이클립스 패키지에서 확인할 수 있다.

```
package com.tis.sample;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

import lombok.Data;
import lombok.Setter;

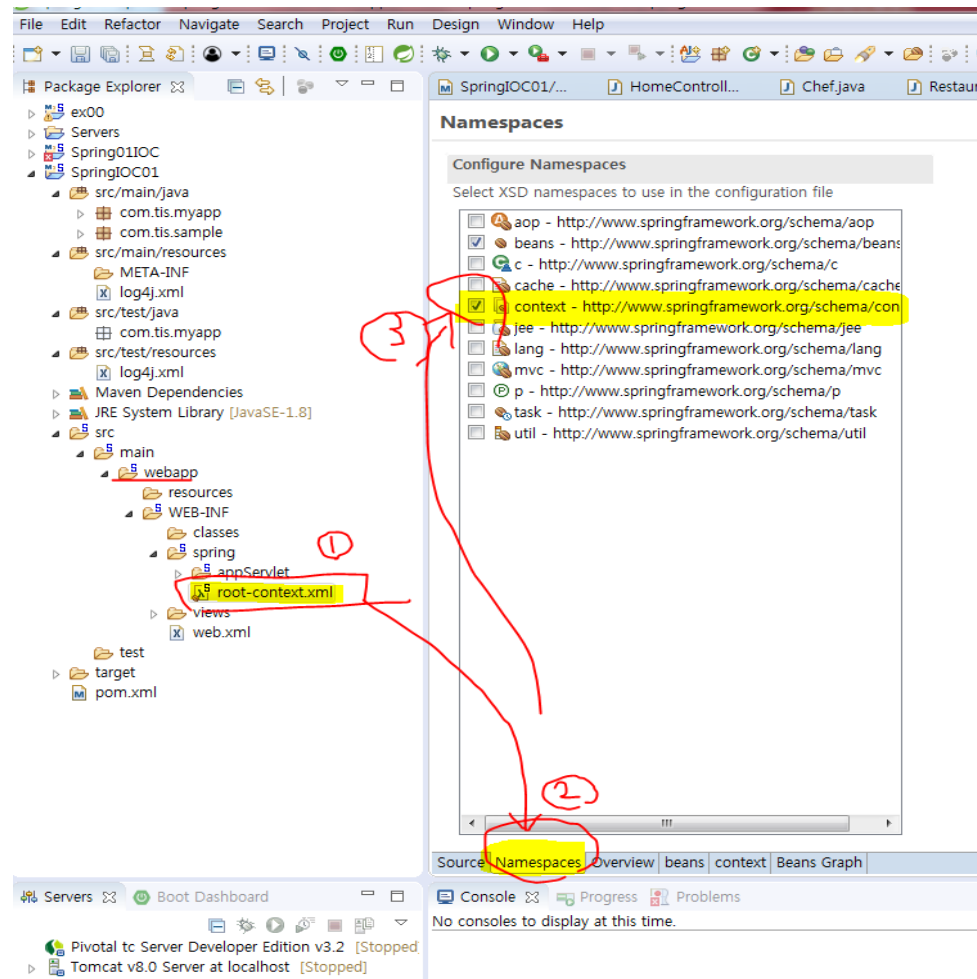
@Component
@Data
public class Restaurant {

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

}
```

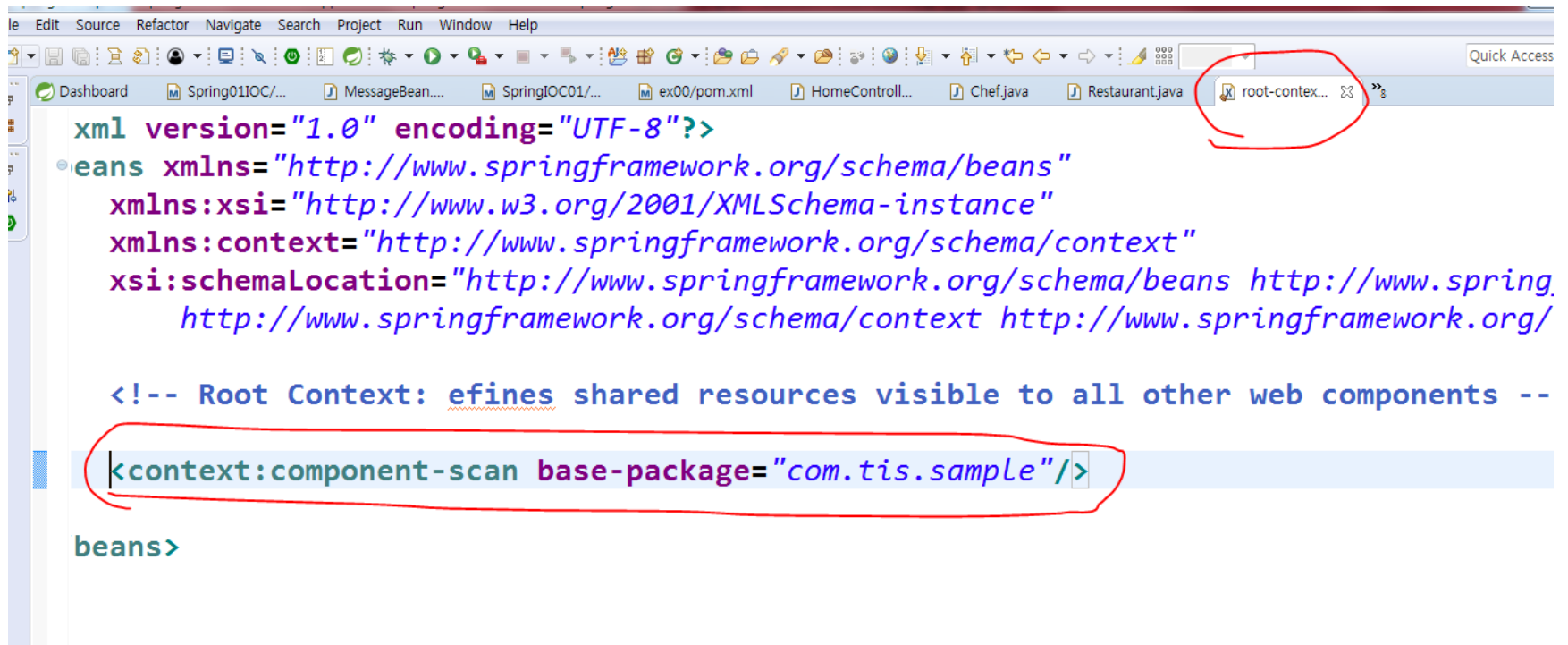
XML을 이용한 DI설정

- root-context.xml에 namespace 'context'를 추가한다.



root-context.xml에 설정 추가

<context:component-scan base-package="com.tis.sample">
를 추가하자.



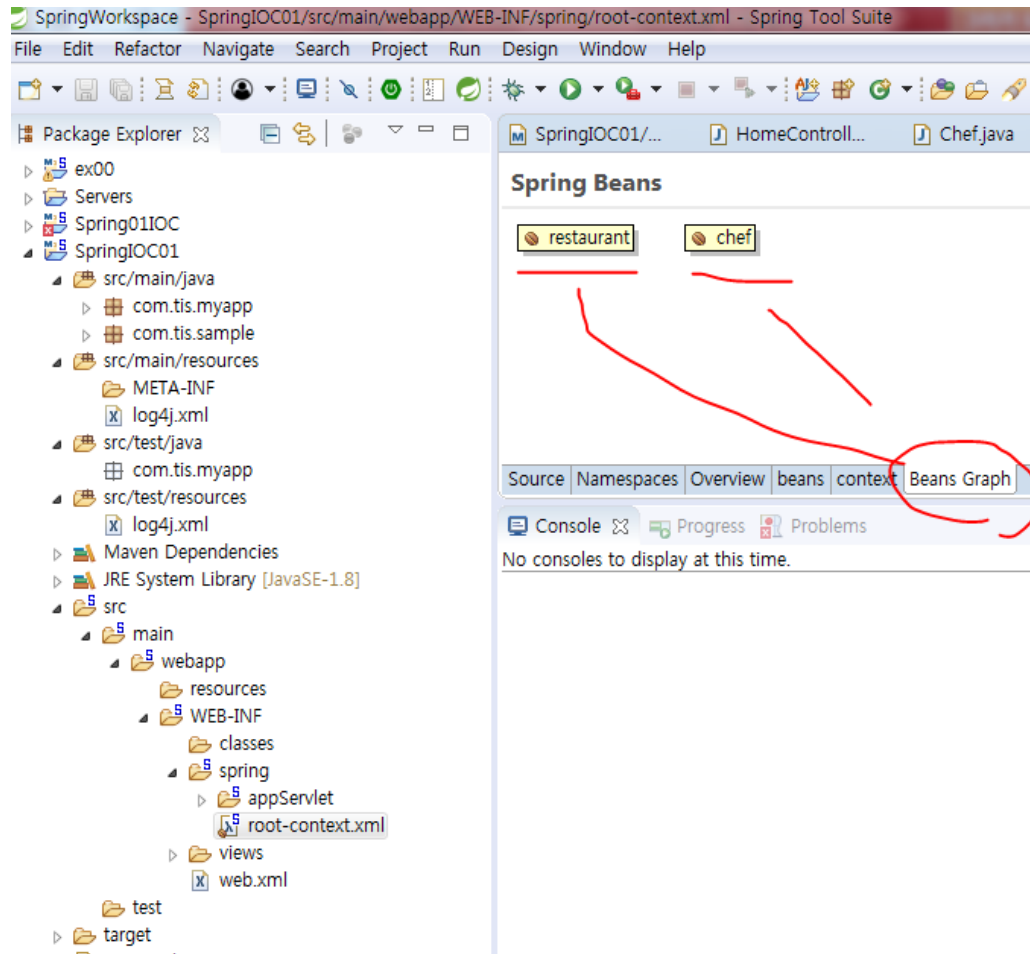
The screenshot shows an IDE window with the file 'root-context.xml' open. The XML content is as follows:

```
xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.spring
                           http://www.springframework.org/schema/context http://www.springframework.org/

<!-- Root Context: defines shared resources visible to all other web components -->
<context:component-scan base-package="com.tis.sample"/>
```

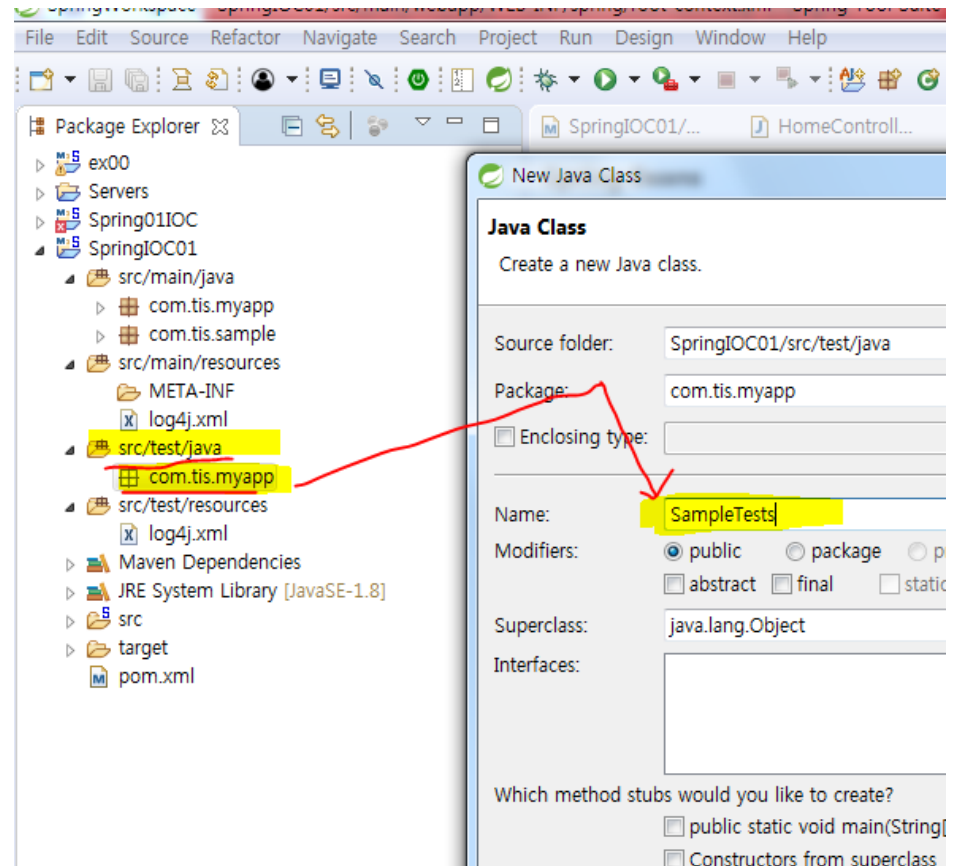
The line `<context:component-scan base-package="com.tis.sample"/>` is highlighted with a red circle. The IDE's toolbar and project explorer are also visible.

Beans Graph에서 확인해보자.



단위 테스트 해보기

- `src/test/java` 폴더 내에 `com.tis.myapp` 패키지
에 **SampleTests**
- 클래스를 추가



```
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.tis.sample.Restaurant;

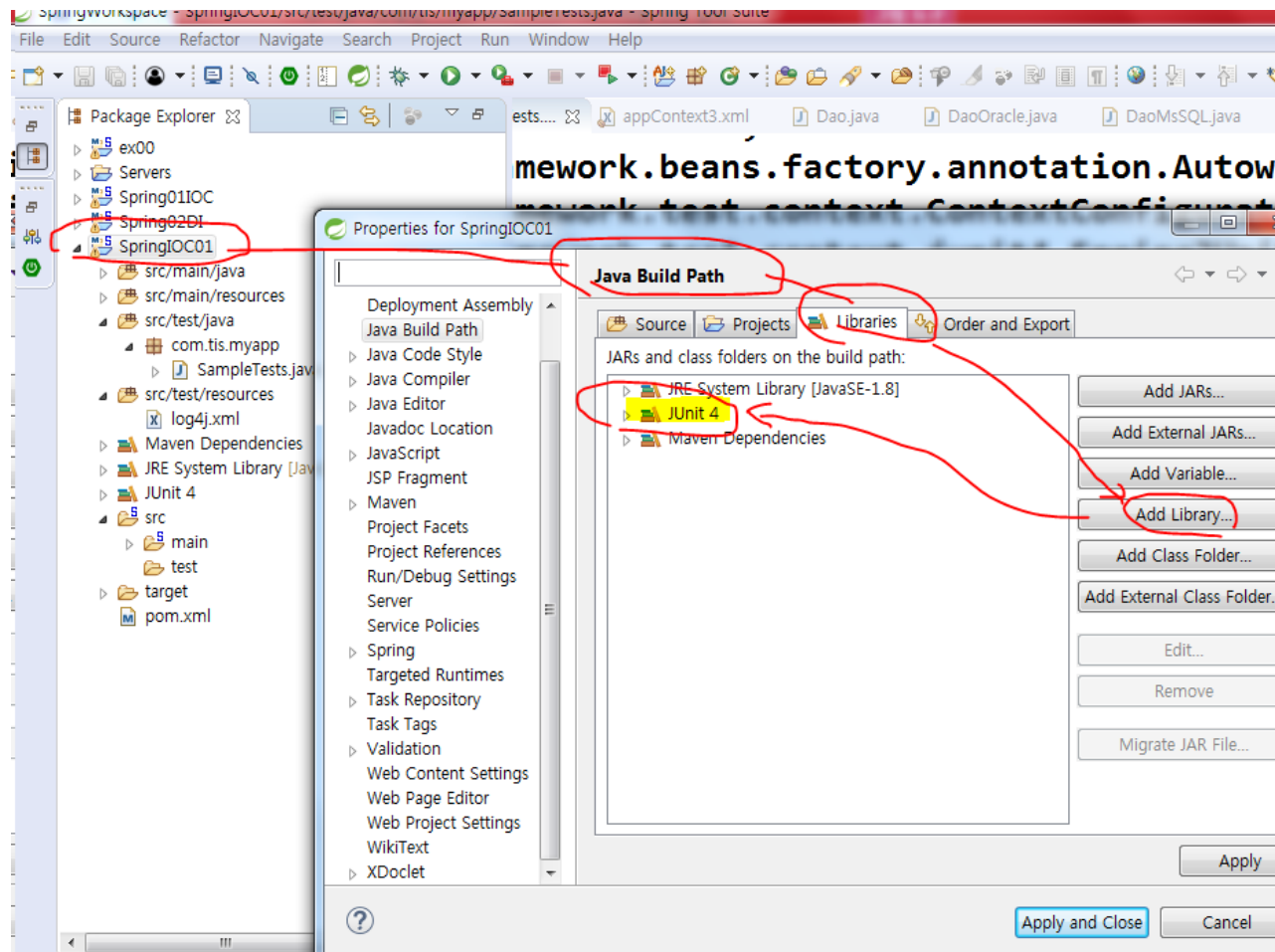
import lombok.Setter;
import lombok.extern.log4j.Log4j;

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

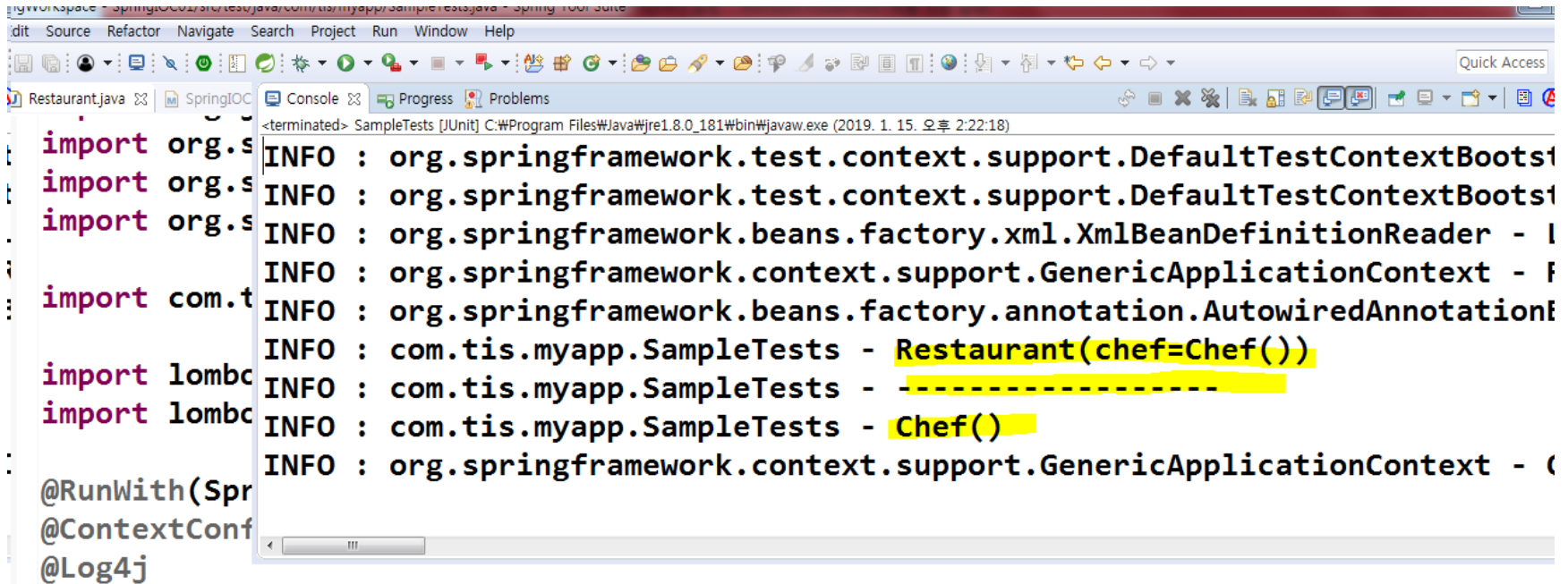
    @Setter(onMethod_ = {@Autowired})
    private Restaurant rest;

    // @Test: 테스트 대상을 표시하는 어노테이션
    @Test
    public void testExist() {
        // rest가 null이 아니어야만 테스트가 성공한다는 것을 의미
        assertNotNull(rest);
        log.info(rest);
        log.info("-----");
        log.info(rest.getChef());
    }
}
```

단위 테스트위해 프로젝트에 junit4라이브러리 추가



Run as> Junit Test를 선택해서 실행



The screenshot shows an IDE window with a Java file named `Restaurant.java` and a console window. The Java file contains the following code:

```
import org.springframework.test.context.support.DefaultTestContextBoots1
import org.springframework.test.context.support.DefaultTestContextBoots1
import org.springframework.beans.factory.xml.XmlBeanDefinitionReader - l
import org.springframework.context.support.GenericApplicationContext - f
import com.tis.myapp.SampleTests - Restaurant(chef=Chef())
import lombok
import lombok

@RunWith(SpringRunner.class)
@ContextConfiguration(classes = {SampleTests.class})
@Log4j
```

The console window shows the output of the JUnit test execution. The output is as follows:

```
<terminated> SampleTests [JUnit] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2019. 1. 15. 오후 2:22:18)
INFO : org.springframework.test.context.support.DefaultTestContextBoots1
INFO : org.springframework.test.context.support.DefaultTestContextBoots1
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - l
INFO : org.springframework.context.support.GenericApplicationContext - f
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationMethodPostProcessor - f
INFO : com.tis.myapp.SampleTests - Restaurant(chef=Chef())
INFO : com.tis.myapp.SampleTests - -----
INFO : com.tis.myapp.SampleTests - Chef()
INFO : org.springframework.context.support.GenericApplicationContext - (
```


테스트 코드 의미

- **@RunWith(SpringJUnit4ClassRunner.class)**
- => 현재 테스트 코드가 스프링을 실행하는 역할을 할 것이라는 것을 표시함
- **@ContextConfiguration** : 스프링 설정파일을 참조해 빈으로 등록함
- **@Log4j** : Lombok을 이용해서 로그를 기록하는 Logger를 변수로 생성함. 별도의 Logger객체 선언 없이도 Log4j라이브러리와 설정이 존재하면 바로 사용 가능

