

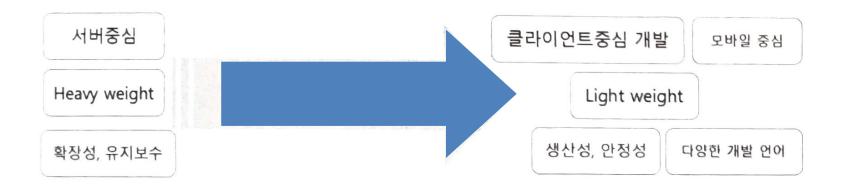
2024년 상반기 K-디지털 트레이닝

스프링의 특징과 의존성 주입

[KB] IT's Your Life



☑ 경량 프레임워크(light-weight Framework)



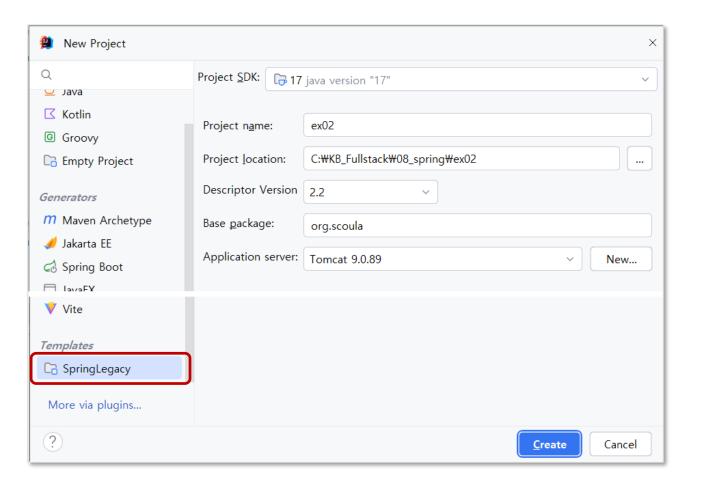
1

🗸 스프링의 주요 특징

- o POJO(Plain Old Java Object) 기반의 구성
- 의존성 주입(Dependency Injection)
- o AOP(Aspect Oriented Programming) 지원
- 편리한 MVC 구조
- WAS에 종속적이지 않은 개발 환경

☑ 프로젝트 만들기

- o File > New > Project...
 - Templates: Spring Legacy



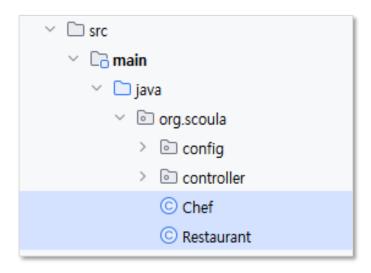
settings.gradle

rootProject.name = "ex02"

gradle sync!

🧿 예제 클래스 생성

- o org.scoula
 - Chef.java
 - Restaurant.java



Chef.java

```
package org.scoula;
import org.springframework.stereotype.Component;

@Component
public class Chef {
}
```

Restaurant.java

```
package org.scoula;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import lombok.Data;
import lombok.Setter;

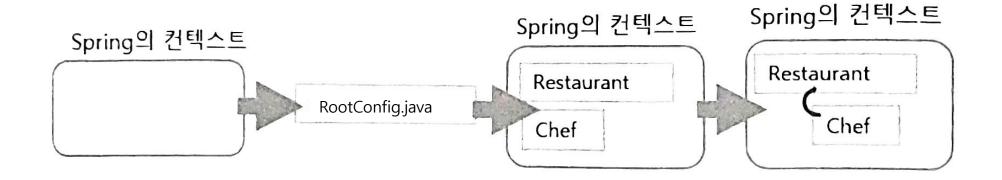
@Component
@Data
public class Restaurant {
    @Autowired
    private Chef chef;
}
```

RootConfig.java

```
package org.scoula.config;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

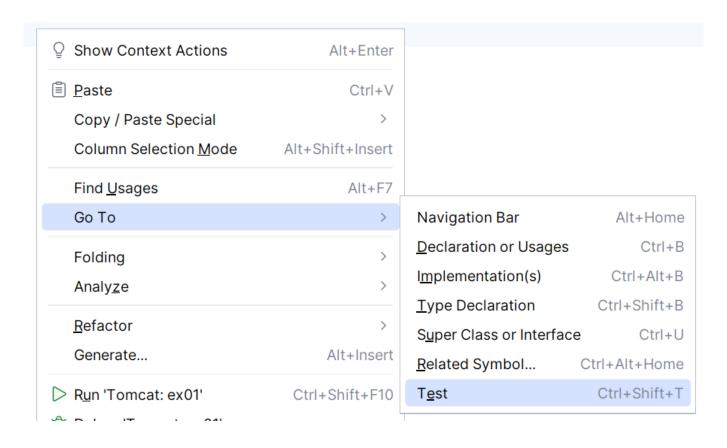
@Configuration
@ComponentScan(basePackages = {"org.scoula"})
public class RootConfig {
}
```

Sprng Context의 구성



테스트 코드를 통한 확인

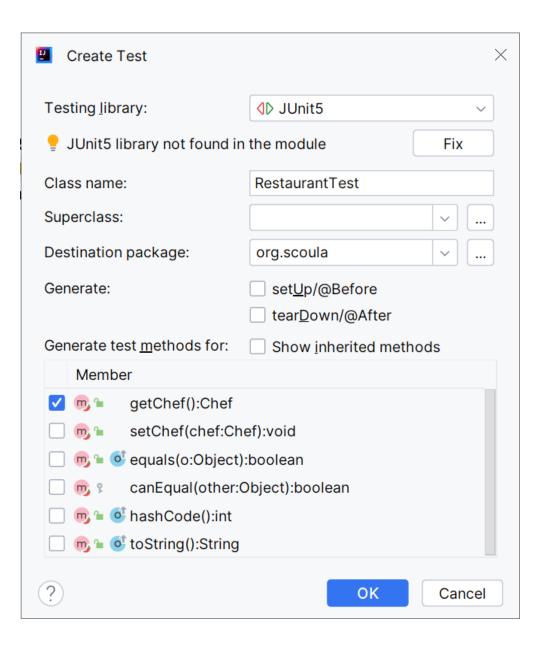
- 테스트 클래스 만들기
 - Restaurant.java 편집창 >> Go To > Test





테스트 코드를 통한 확인

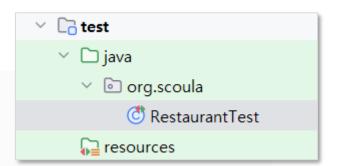
테스트 클래스 만들기



의존성 주입

test :: RestaurantTest.java

```
package org.scoula;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
class RestaurantTest {
      @Test
      void getChef() {
       }
}
```




```
import static org.junit.jupiter.api.Assertions.assertNotNull;
@ExtendWith(SpringExtension.class)
@ContextConfiguration(classes = {RootConfig.class})
@Log4j
class RestaurantTest {
    @Autowired
                                                                         19
    private Restaurant restaurant;
                                                                         20
                                                                                   @Test
    @Test
                                                                              Run 'RestaurantTest.getChef'
                                                                                                          Ctrl+Shift+F10
                                                                         22
    void getChef() {
                                                                              Debug 'RestaurantTest.getChef'
                                                                         23
        assertNotNull(restaurant);
                                                                              Run 'RestaurantTest.getChef' with Coverage
                                                                         24
        log.info(restaurant);
                                                                         25
                                                                                Modify Run Configuration...
        log.info("-----");
        log.info(restaurant.getChef());
```

```
INFO : org.scoula.RestaurantTest - Restaurant(chef=org.scoula.Chef@25230246)
INFO : org.scoula.RestaurantTest - ------
INFO : org.scoula.RestaurantTest - org.scoula.Chef@25230246
```

@Log4j

- Lombok을 이용해서 로그를 기록하는 Logger 변수 생성
- 로그 설정 파일
 - src/main/resources/log4j.xml
 - 실제 운영을 위한 로거 설정
 - src/test/resources/log4j.xml
 - 테스트를 위한 로거 설정
 - 없으면 운영을 위한 로거 설정이 적용

