

2024년 상반기 K-디지털 트레이닝

개발을 위한 준비

[KB] IT's Your Life

1 개발을 위한 준비

✓ workspace 준비

- C:\KB_Fullstack\W08_Spring

1 개발을 위한 준비

New Project

Search:

Generators

- Kotlin
- Groovy
- Empty Project
- Maven Archetype
- Jakarta EE**
- Spring Boot
- JavaFX
- Quarkus
- Micronaut
- Ktor
- Compose for Desktop
- HTML
- React
- Express
- Angular CLI

Name:

Location:

Project will be created in: C:\WKB_FullstackW08_spring\ex01

☐ Create Git repository

Template:

Application server:

Language:

Build system:

Group:

Artifact:

JDK:

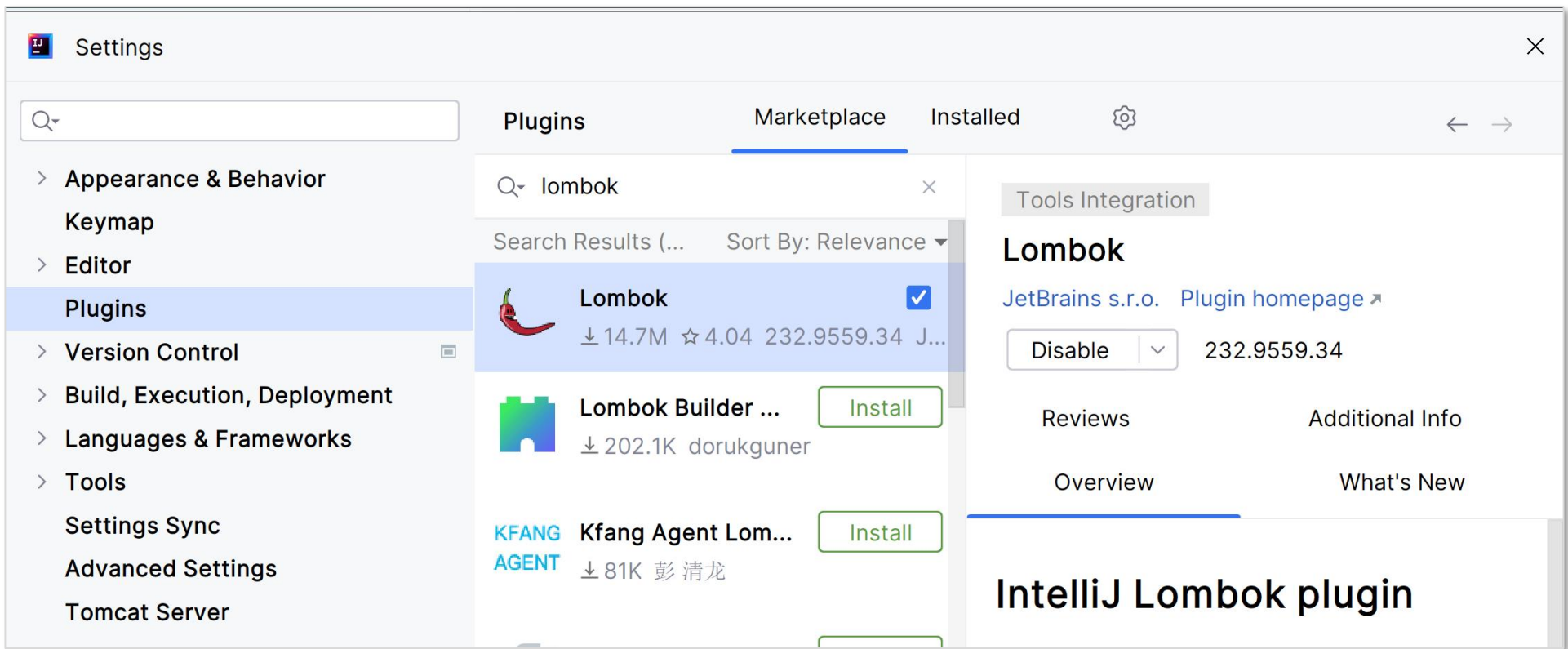
1 개발을 위한 준비

- ✓ **Version: Jakarta EE 8**
 - Servlet 4.0.1

1 개발을 위한 준비

✓ Lombok Plugin 설치

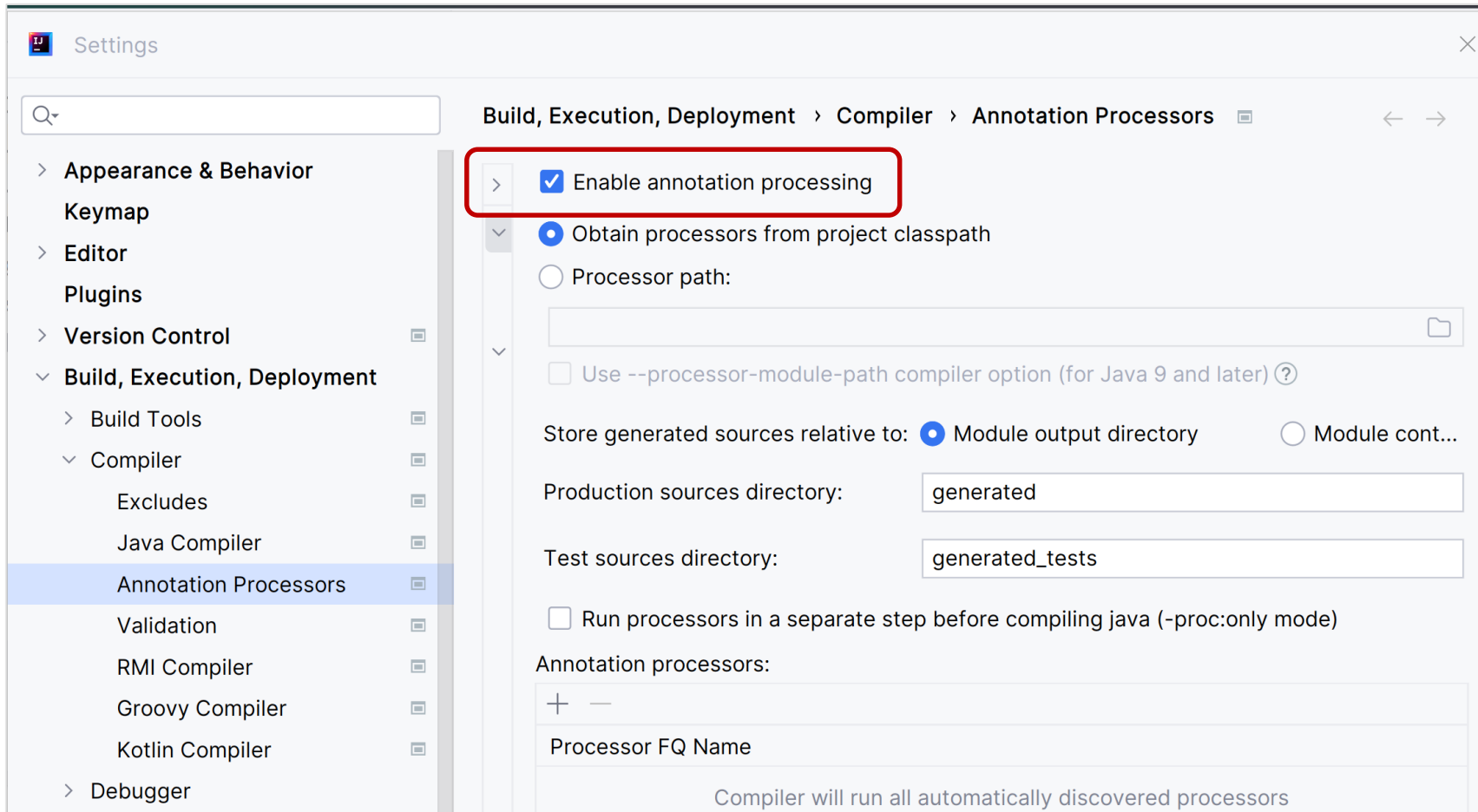
- File > Settings... > Plugins
 - Marketplace에서 Lombok 검색 후 설치



1 개발을 위한 준비

✓ Lombok 설정

- File > Setting... > Build, Execution, Deployment > Compiler > Annotation Processors
 - Enable annotation processing 체크 → 프로젝트 생성시 마다 해줘야 함



1 개발을 위한 준비

build.gradle

```
plugins {  
    id 'java'  
    id 'war'  
}  
  
group 'org.scoula'  
version '1.0-SNAPSHOT'  
  
repositories {  
    mavenCentral()  
}  
  
ext {  
    junitVersion = '5.9.2'  
    springVersion = '5.3.37'  
    lombokVersion = '1.18.30'  
}  
  
sourceCompatibility = '1.17'  
targetCompatibility = '1.17'  
  
tasks.withType(JavaCompile) {  
    options.encoding = 'UTF-8'  
}
```

1 개발을 위한 준비

build.gradle

```
dependencies {  
    // 스프링  
    implementation ("org.springframework:spring-context:${springVersion}")  
        { exclude group: 'commons-logging', module: 'commons-logging' }  
    implementation "org.springframework:spring-webmvc:${springVersion}"  
    implementation 'javax.inject:javax.inject:1'  
  
    // AOP  
    implementation 'org.aspectj:aspectjrt:1.9.20'  
    implementation 'org.aspectj:aspectjweaver:1.9.20'  
  
    // JSP, SERVLET, JSTL  
    implementation('javax.servlet:javax.servlet-api:4.0.1')  
    compileOnly 'javax.servlet.jsp:jsp-api:2.1'  
    implementation 'javax.servlet:jstl:1.2'  
  
    // Logging  
    implementation 'org.slf4j:slf4j-api:2.0.9'  
    runtimeOnly 'org.slf4j:jcl-over-slf4j:2.0.9'  
    runtimeOnly 'org.slf4j:slf4j-log4j12:2.0.9'  
    implementation 'log4j:log4j:1.2.17'  
  
    // xml내 한글 처리  
    implementation 'xerces:xercesImpl:2.12.2'
```


1 개발을 위한 준비

✎ build.gradle

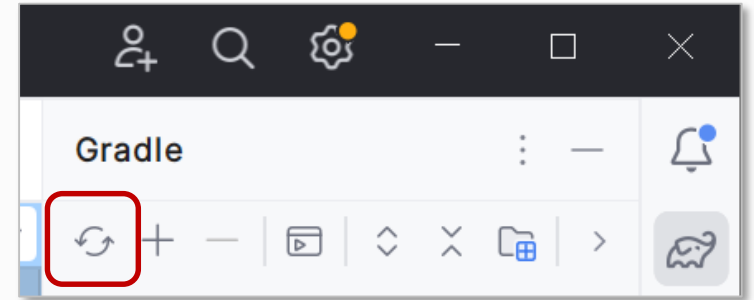
```
// Lombok
compileOnly "org.projectlombok:lombok:${lombokVersion}"
annotationProcessor "org.projectlombok:lombok:${lombokVersion}"

// Jackson - Json 처리
implementation 'com.fasterxml.jackson.core:jackson-databind:2.9.4'

// 테스트
testImplementation "org.springframework:spring-test:${springVersion}"
testCompileOnly "org.projectlombok:lombok:${lombokVersion}"
testAnnotationProcessor "org.projectlombok:lombok:${lombokVersion}"

testImplementation("org.junit.jupiter:junit-jupiter-api:${junitVersion}")
testRuntimeOnly("org.junit.jupiter:junit-jupiter-engine:${junitVersion}")
}

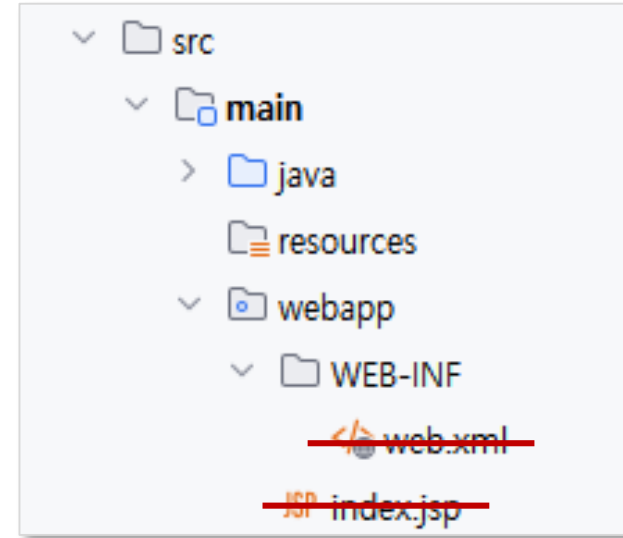
test {
    useJUnitPlatform()
}
```



1 개발을 위한 준비

✓ 디렉토리 구조

- java
 - Java 클래스 배치
- resources
 - Java 클래스를 제외한 모든 파일(설정, xml 등)
- webapp
 - Web URL로 접근하는 root 디렉토리
 - WEB-INF
 - 웹 설정, 스프링 View 배치 디렉토리
 - index.jsp, web.xml 삭제



1 개발을 위한 준비

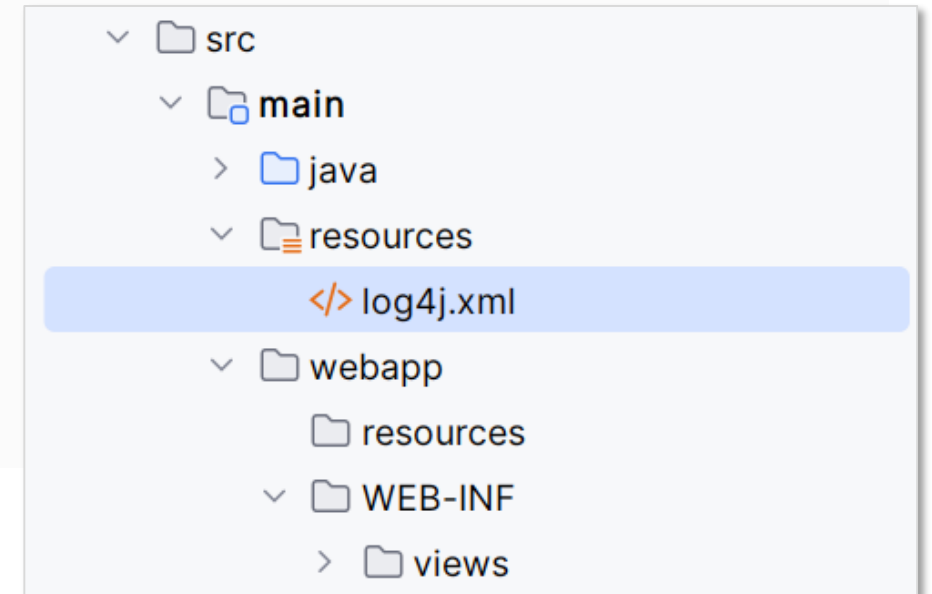
resources/log4j.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <!-- Appenders -->
  <appender name="console" class="org.apache.log4j.ConsoleAppender">
    <param name="Target" value="System.out" />
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%-5p: %c - %m%n" />
    </layout>
  </appender>

  <!-- Application Loggers -->
  <logger name="org.scoula">
    <level value="info" />
  </logger>

  <!-- 3rdparty Loggers -->
  <logger name="org.springframework.core">
    <level value="info" />
  </logger>
```



1 개발을 위한 준비

resources/log4j.xml

```
<logger name="org.springframework.beans">
  <level value="info" />
</logger>

<logger name="org.springframework.context">
  <level value="info" />
</logger>

<logger name="org.springframework.web">
  <level value="info" />
</logger>

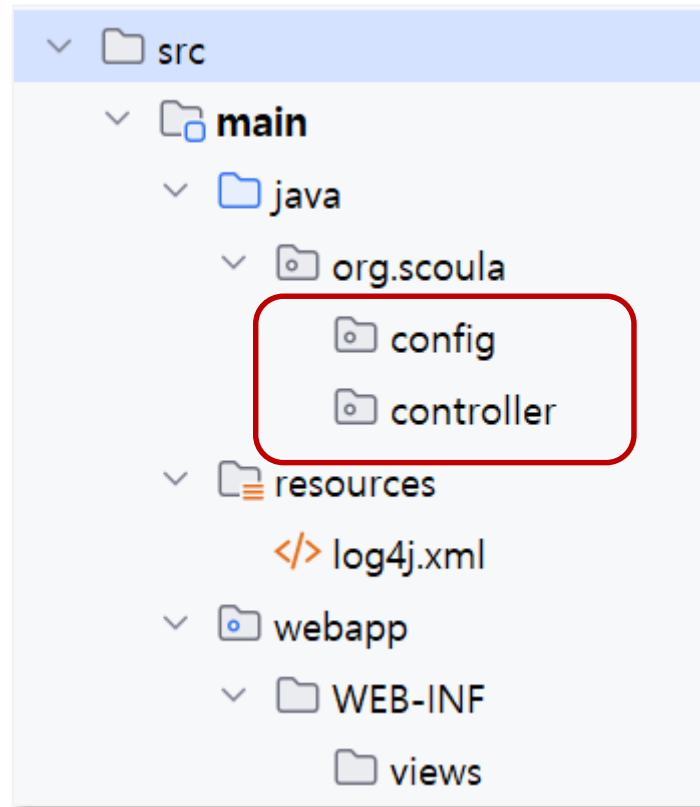
<!-- Root Logger -->
<root>
  <priority value="info" />
  <appender-ref ref="console" />
</root>

</log4j:configuration>
```

1 개발을 위한 준비

✓ 패키지 생성

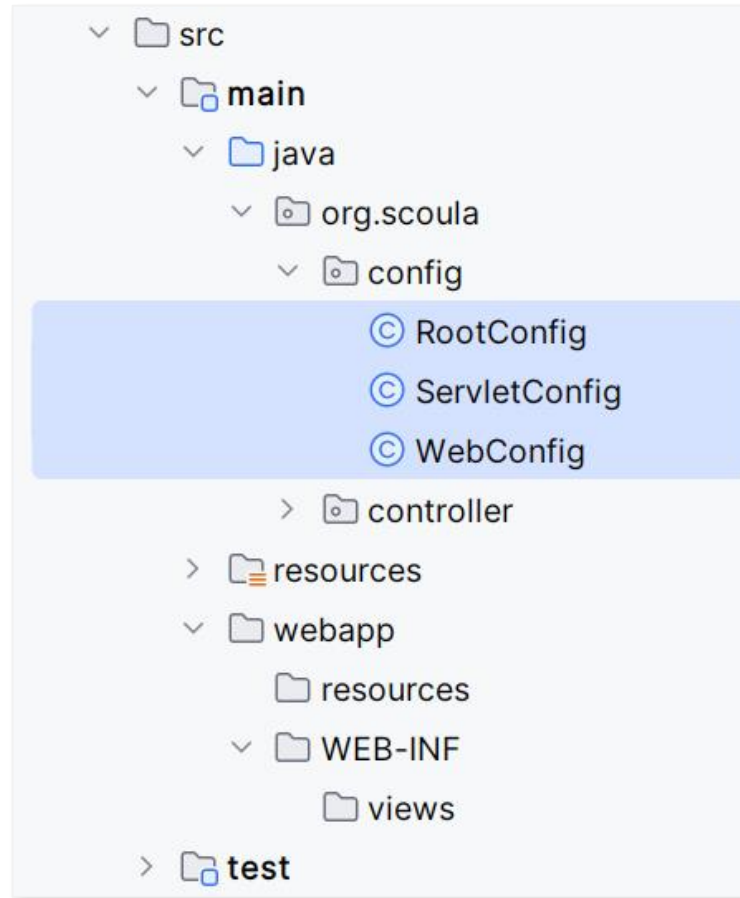
- src/main/java에 패키지 생성
 - org.scoula
 - config : 스프링 설정 패키지
 - controller : 기본 컨트롤러 패키지



1 개발을 위한 준비

✓ 설정 파일 만들기

- org.scoula.config
 - RootConfig.java
 - ServletConfig.java
 - WebConfig.java



1 개발을 위한 준비

RootConfig.java

```
package org.scoula.config;

import org.springframework.context.annotation.Configuration;

@Configuration
public class RootConfig {

}
```

1 개발을 위한 준비

ServletConfig.java

```
package org.scoula.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.ViewResolverRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.view.InternalResourceViewResolver;
import org.springframework.web.servlet.view.JstlView;

@EnableWebMvc
@ComponentScan(basePackages = {"org.scoula.controller"}) // Spring MVC용 컴포넌트 등록을 위한 스캔 패키지
public class ServletConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry
            .addResourceHandler("/resources/**") // url이 /resources/로 시작하는 모든 경로
            .addResourceLocations("/resources/"); // webapp/resources/경로로 매핑
    }
}
```


1 개발을 위한 준비

ServletConfig.java

```
// jsp view resolver 설정
@Override
public void configureViewResolvers(ViewResolverRegistry registry) {
    InternalResourceViewResolver bean = new InternalResourceViewResolver();

    bean.setViewClass(JstlView.class);
    bean.setPrefix("/WEB-INF/views/");
    bean.setSuffix(".jsp");

    registry.viewResolver(bean);
}
}
```

1 개발을 위한 준비

WebConfig.java

```
package org.scoula.config;

import org.springframework.web.filter.CharacterEncodingFilter;
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

import javax.servlet.Filter;

public class WebConfig extends AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return new Class[] { RootConfig.class };
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] { ServletConfig.class };
    }

    // 스프링의 FrontController인 DispatcherServlet이 담당할 Url 매핑 패턴, / : 모든 요청에 대해 매핑
    @Override
    protected String[] getServletMappings() {
        return new String[] { "/" };
    }
}
```

1 개발을 위한 준비

WebConfig.java

```
// POST body 문자 인코딩 필터 설정 - UTF-8 설정
protected Filter[] getServletFilters() {
    CharacterEncodingFilter characterEncodingFilter = new CharacterEncodingFilter();

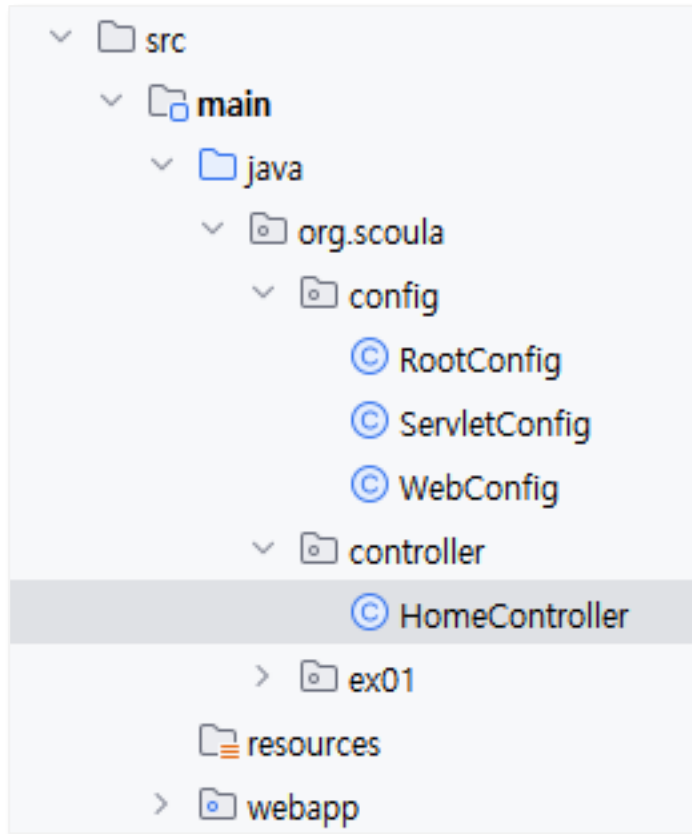
    characterEncodingFilter.setEncoding("UTF-8");
    characterEncodingFilter.setForceEncoding(true);

    return new Filter[] {characterEncodingFilter};
}
```

1 개발을 위한 준비

✓ 컨트롤러 만들기

- org.scoula.controller
 - HomeController.java



1 개발을 위한 준비

HomeController.java

```
package org.scoula.controller;

import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
@Slf4j
public class HomeController {

    @GetMapping("/")
    public String home() {
        log.info("=====> HomController /");
        return "index";    // View의 이름
    }

}
```

1 개발을 위한 준비

✓ Jsp 파일 만들기

○ src/main/webapp

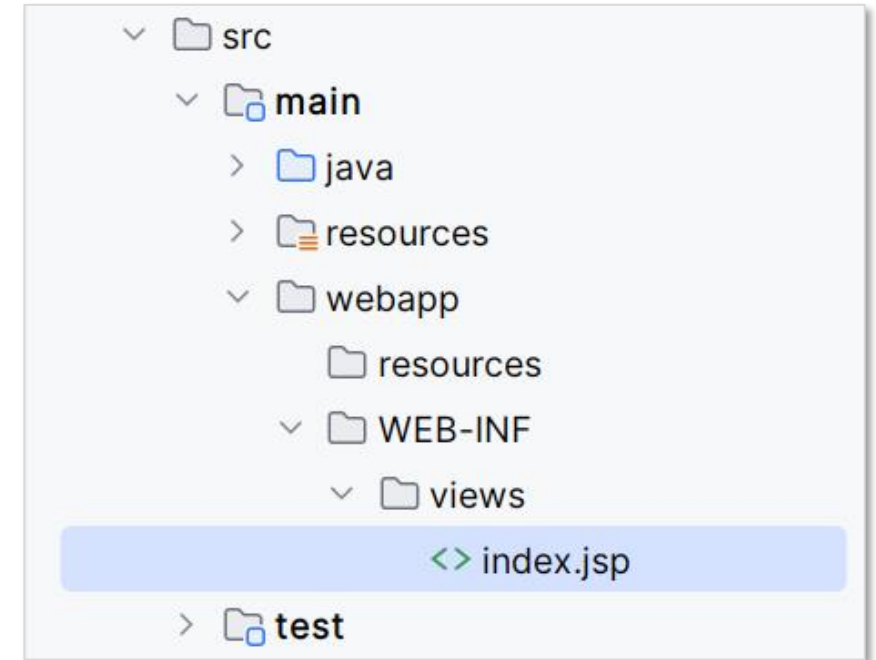
- WEB-INF/views/index.jsp

```
@Override
public void configureViewResolvers(ViewResolverRegistry registry) {
    InternalResourceViewResolver bean = new InternalResourceViewResolver();

    bean.setViewClass(JstlView.class);
    bean.setPrefix("/WEB-INF/views/");
    bean.setSuffix(".jsp");

    registry.viewResolver(bean);
}
```

- controller에서 index라는 view 이름을 리턴한 경우
/WEB-INF/views/ + view 이름 + .jsp
→ /WEB-INF/views/index.jsp



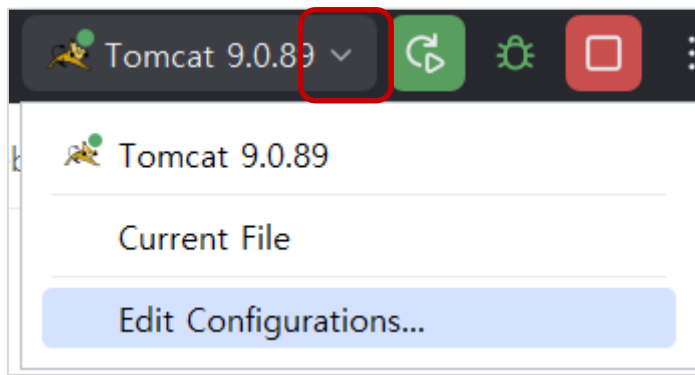
1 개발을 위한 준비

views/index.jsp

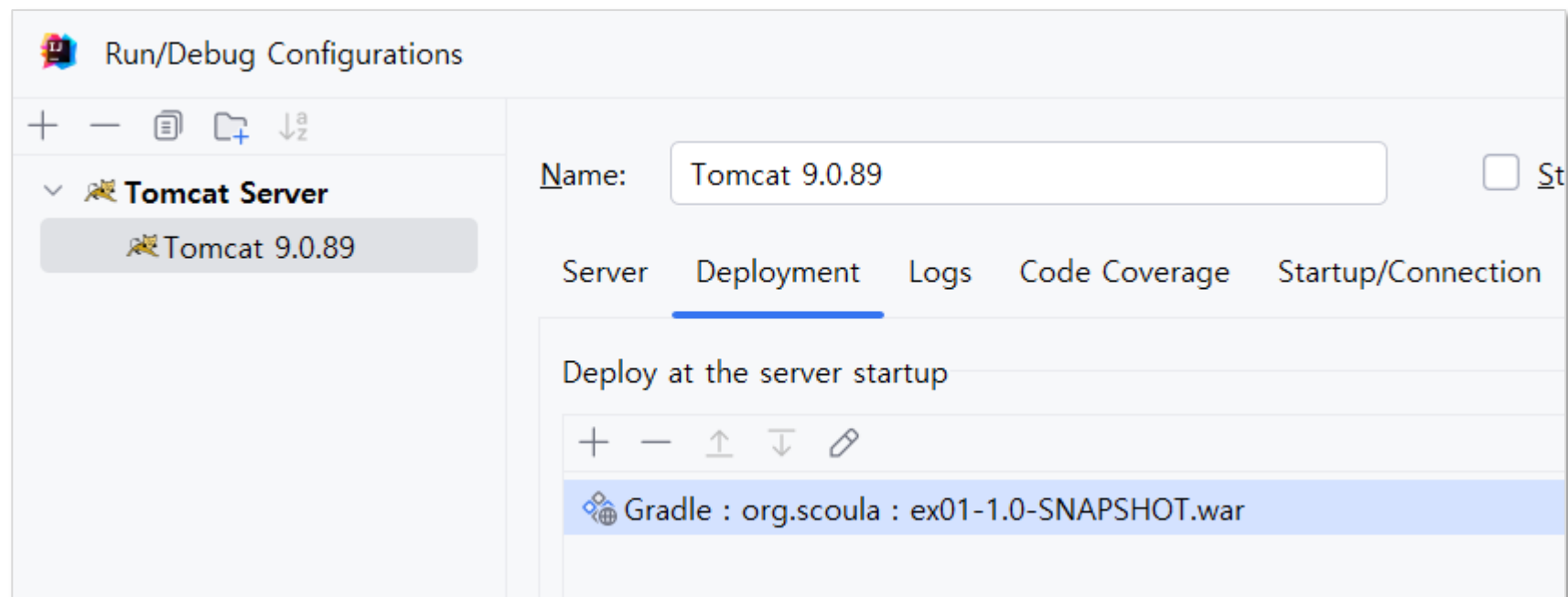
```
<!DOCTYPE html>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>환영합니다.</h1>
</body>
</html>
```

1 개발을 위한 준비

✓ 톰캣 설정



○ Application context: /



1 개발을 위한 준비

✓ 애플리케이션 실행



Run ex00 x

Console Tomcat Localhost Log Tomcat Access Log

31-Oct-2023 11:37:48.461 정보 [main] org.apache.jasper.servlet.JspScanner.scanJars 식어도 하나의 JAR가 ILD들을 찾기 위해 스캔되었는

INFO : org.springframework.web.context.ContextLoader - Root WebApplicationContext: initialization started

INFO : org.springframework.web.context.ContextLoader - Root WebApplicationContext initialized in 555 ms

INFO : org.springframework.web.servlet.DispatcherServlet - Initializing Servlet 'dispatcher'

INFO : org.springframework.web.servlet.DispatcherServlet - Completed initialization in 645 ms

31-Oct-2023 11:37:50.138 정보 [main] org.apache.catalina.startup.HostConfig.deployDescriptor 배치 descriptor [C:\Users\student\.Sm

31-Oct-2023 11:37:50.140 정보 [main] org.apache.coyote.AbstractProtocol.start 프로토콜 핸들러 ["http-nio-8080"]을(를) 시작합니다.

31-Oct-2023 11:37:50.170 정보 [main] org.apache.catalina.startup.Catalina.start 서버가 [3518] 밀리초 내에 시작되었습니다.

<http://localhost:8080/>

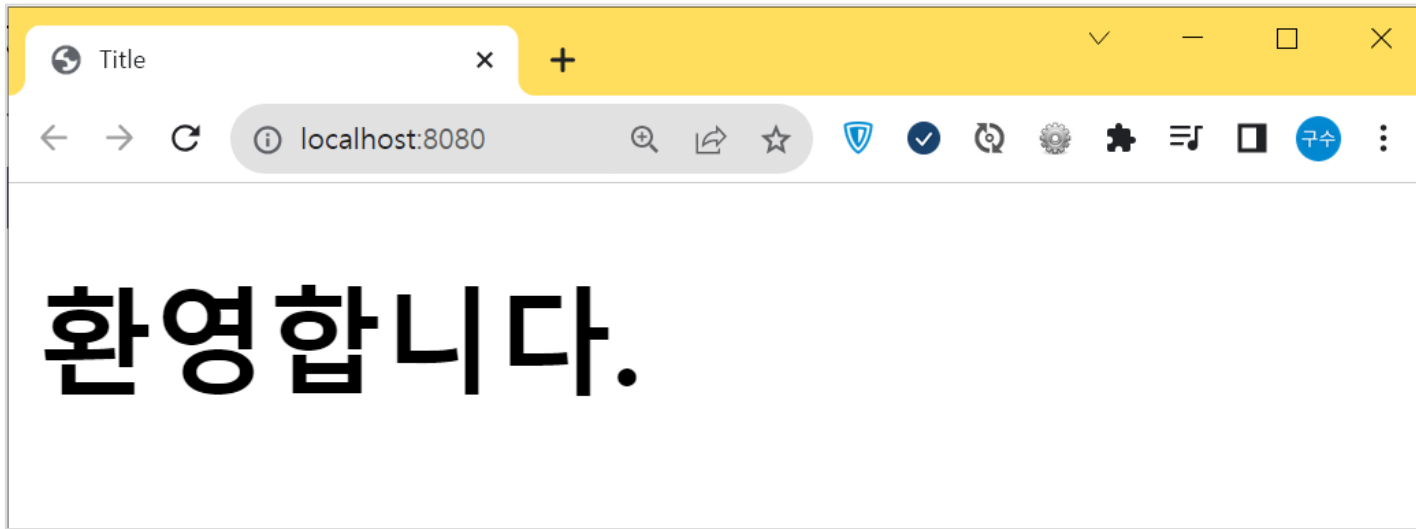
ex00 > src > main > webapp > WEB-INF > views > index.jsp

1 개발을 위한 준비

✓ 확인

- <http://localhost:8080/>

```
31-Oct-2023 11:40:34.738 INFO [main] org.apache.catalina.startup.Catalina.start() 서비스 시작 [3]  
http://localhost:8080/  
INFO : org.scoula.book.controller.HomeController - =====> HomeController /
```

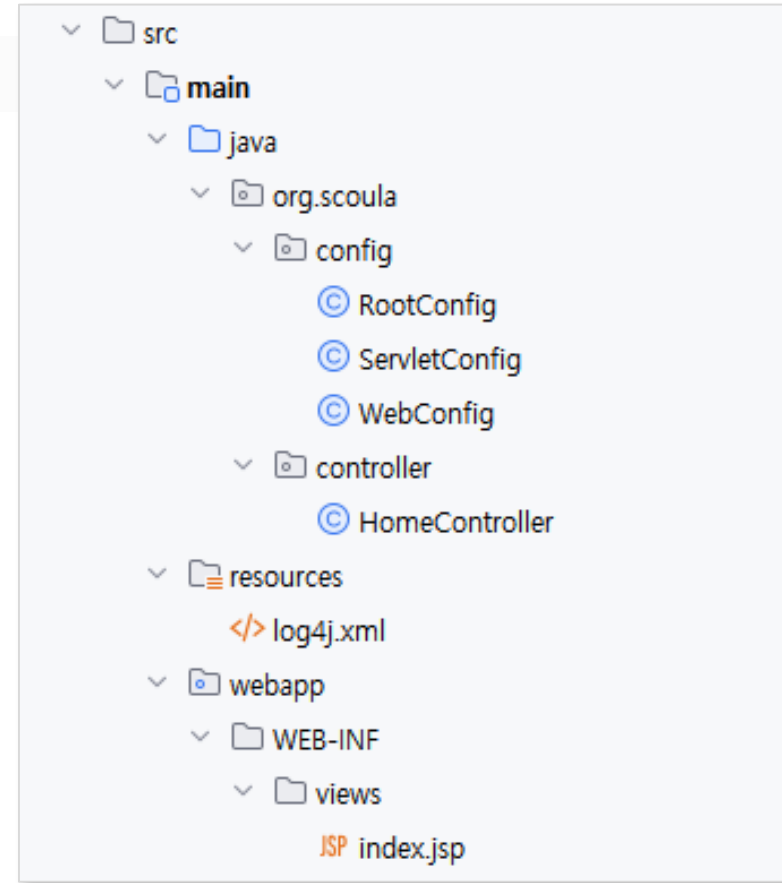


✓ 프로젝트 템플릿 만들기

- 탐색기에서 ex01 프로젝트 폴더를 SpringLegacy이름으로 복사 후 SpringLegacy 프로젝트 열기
- SpringLegacy/settings.gradle

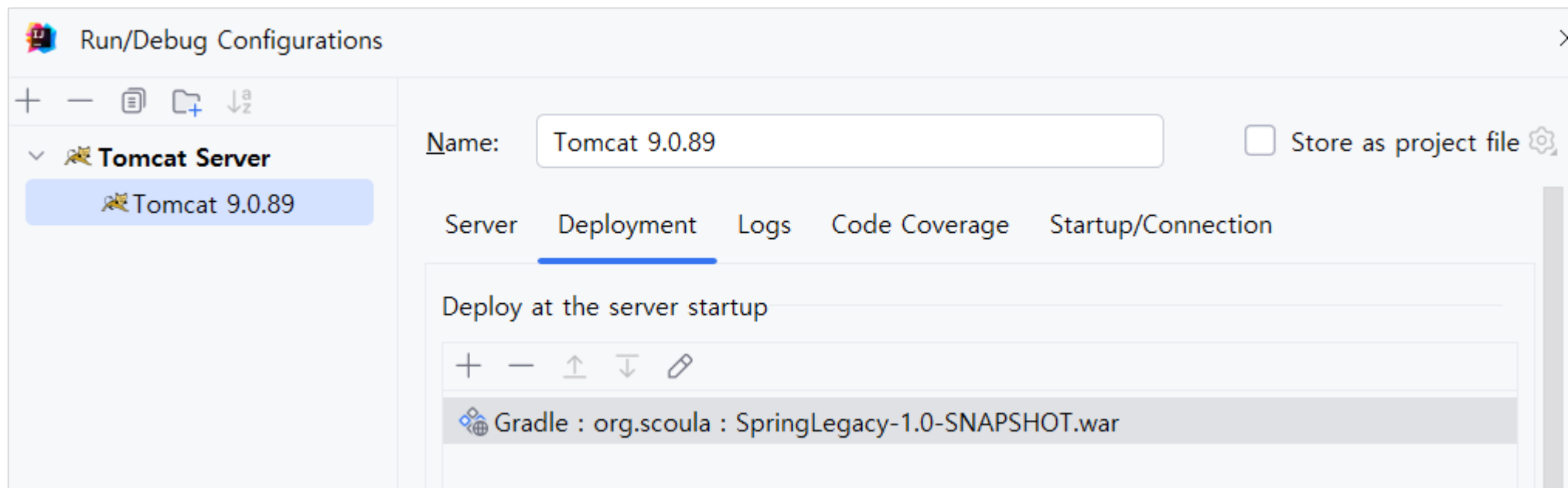
```
rootProject.name = "SpringLegacy"
```

- 수정 후 Sync



✓ 톰캣 설정 수정 → ArtifactId 변경

```
rootProject.name = "SpringLegacy"
```



1 개발을 위한 준비

✓ 프로젝트 템플릿 만들기

- File > New Projects Setup > Save Project as Template...

