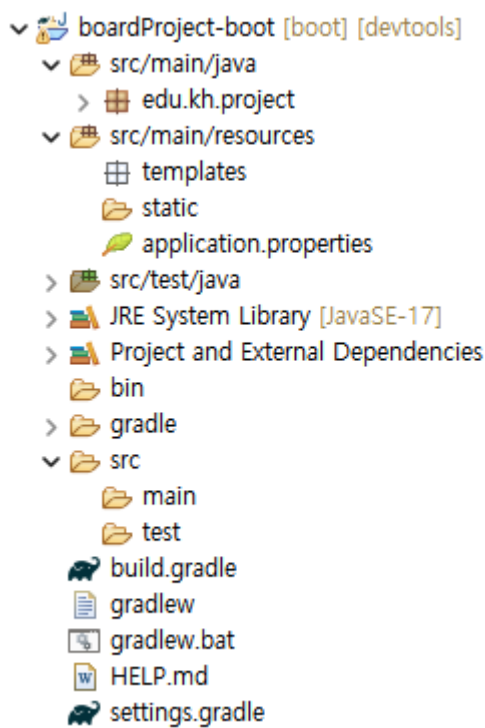




06_Spring Boot 프로젝트, 파일 구조 파악

1. 프로젝트 구조
2. BoardProjectBootApplication.java (XXXAppllication.java)
3. bulid.gradle

1. 프로젝트 구조



- src/main/java : 자바 소스 코드를 모아두는 폴더
- src/main/resources : 자바에서 사용할 자원, 설정 파일을 모아두는 폴더
 - templates : Java 템플릿을 모아두는 폴더(thymeleaf)
 - static : templates에 적용될 정적 자원을 모아두는 폴더(JS, CSS, 이미지)
 - application.properties : Spring Boot 프로젝트 관련 설정을 정의하는 파일 (중요!)
- Project and External Dependencies : build.gradle에 명시된 라이브러리 모음
- build.gradle : gradle(빌드도구 == maven의 pom.xml)
프로젝트에 필요한 라이브러리, 배포, 빌드 관련 설정을 작성하는 파일

2. BoardProjectBootApplication.java (XXXAppllication.java)

```
package edu.kh.project;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration;

@SpringBootApplication(exclude = {SecurityAutoConfiguration.class})
public class BoardProjectBootApplication {

    public static void main(String[] args) {
        SpringApplication.run(BoardProjectBootApplication.class, args);
    }
}
```

```
}
```

- @SpringBootApplication : Spring Boot Application에 필요한 필수 어노테이션을 모아둔 어노테이션
 - Spring Boot 프로젝트로 만든 애플리케이션의 실행을 담당하는 클래스
 - Spring Application을 최소 설정으로 간단하고 빠르게 실행할 수 있게 해줌
 - *** Java 파일을 실행하듯이 Run 버튼을 클릭하면 배포가 시작됨 ***
- (exclude = {SecurityAutoConfiguration.class})
 - Spring Security 자동 설정을 사용하지 않음

```
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Inherited
@SpringBootConfiguration
@EnableAutoConfiguration
@ComponentScan(excludeFilters = { @Filter(type = FilterType.CUSTOM, classes = TypeExcludeFilter.class),
    @Filter(type = FilterType.CUSTOM, classes = AutoConfigurationExcludeFilter.class) })
public @interface SpringBootApplication {
```

@SpringBootApplication 어노테이션이 정의된 인터페이스

- @SpringBootConfiguration
 - Spring설정 파일(설정을 위해 만든 클래스)임을 나타내는 어노테이션
 - @Configuration(해당 클래스는 자바 기반 설정 파일임을 나타내는 어노테이션) 자식 어노테이션
- @ComponentScan
 - 지정된 패키지 이하에 bean 등록을 위한 어노테이션을 인식하는 어노테이션(bean 등록이 수행됨)
 - @SpringBootApplication이 붙은 클래스가 위치하고 있는 패키지 이하가 @ComponentScan 지정 패키지가 됨. (최상위 패키지 바로 아래 클래스가 위치해야함!)
- @EnableAutoConfiguration
 - 자동 설정을 위해 사용되는 어노테이션
 - 등록된 bean과 조건에 따라 자동으로 설정을 적용
 - spring-boot-autoconfigure 라이브러리
→ spring.factories 파일에 자동 설정을 위한 클래스가 등록되어있음

3. bulid.gradle

```

plugins {
    id 'java'
    id 'org.springframework.boot' version '3.2.3'
    id 'io.spring.dependency-management' version '1.1.4'
}

group = 'edu.kh'
version = '0.0.1-SNAPSHOT'

java {
    sourceCompatibility = '17'
}

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {
    //implementation 'org.springframework.boot:spring-boot-starter-jdbc'
    implementation 'org.springframework.boot:spring-boot-starter-mail'
    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-websocket'
    //implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:3.0.3'
    implementation 'org.thymeleaf.extras:thymeleaf-extras-springsecurity6'
    compileOnly 'org.projectlombok:lombok'
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    //runtimeOnly 'com.oracle.database.jdbc:ojdbc11'
    annotationProcessor 'org.springframework.boot:spring-boot-configuration-processor'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    testImplementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter-test:3.0.3'
    testImplementation 'org.springframework.security:spring-security-test'
}

tasks.named('test') {
    useJUnitPlatform()
}

```

- Gradle 빌드 도구를 이용하는 프로젝트를 관리할 때 사용하는 파일
- dependencies에 필요한 라이브러리를 작성하면 mvnrepository에서 자동으로 다운로드가 진행됨.

