

2025년 상반기 K-디지털 트레이닝

백엔드 준비

[KB] IT's Your Life



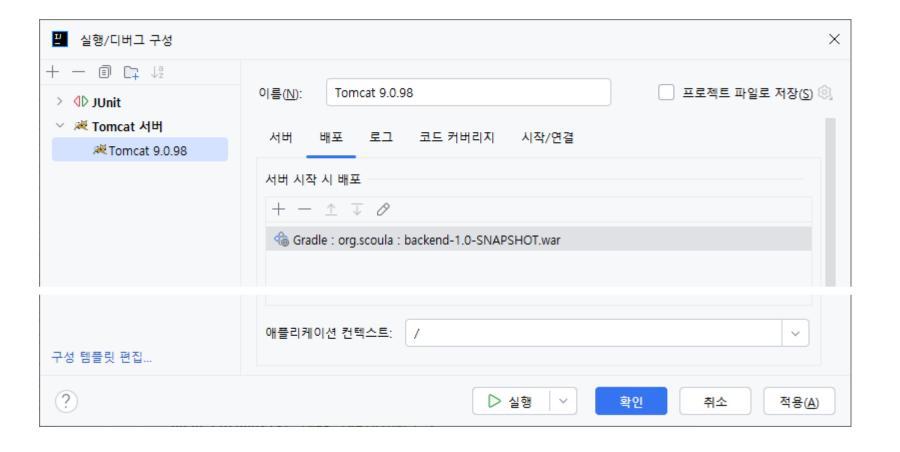
apiserver 프로젝트를 다음 위치로 복사

- C:₩KB_Fullstack₩09_Spring+Vue₩scoula₩backend
- o settings.gradle에서 프로젝트 명 수정 rootProject.name = "backend"
- o build.gradle에 swagger 의존성 추가

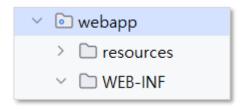
```
// swagger
implementation 'io.springfox:springfox-swagger2:2.9.2'
implementation 'io.springfox:springfox-swagger-ui:2.9.2'
```

→ Gradle Sync

☑ 실행 구성

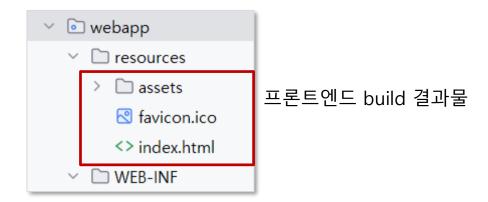


webapp의 하위에 있는 모든 jsp 파일 삭제



webapp/resources

■ 프론트엔드에서 npm run build를 실행하면 배포되는 곳



💟 기본 설정

- /를 요구했을 때 /resources/index.html을 리턴
- 프론트엔드에서 기존 정적 파일은 /assets/**로 기술되어 있음
 - 실제 배치는 /resources/assets/**
 - /assets/**를 요구했을 때 /resources/assets/** 경로로 해석하도록 설정
 - /assets/**는 스프링 처리에서 제외해야 함(resources에 대한 동일한 처리)

o 새로 고침했을 때 해당 url은 백엔드에 없음

- 404에러 발생
- CommonExceptionAdvice에서 index.html로 포워딩
- CommonExceptionAdvice, ApiExceptionAdvice 적용 순서 지정
 - @Order(순번)

config.ServletConfig.java

```
public class ServletConfig implements WebMvcConfigurer {
  @Override
  public void addViewControllers(ViewControllerRegistry registry) {
    registry.addViewController("/")
         .setViewName("forward:/resources/index.html");
  @Override
  public void addResourceHandlers(ResourceHandlerRegistry registry) {
    registry.addResourceHandler("/resources/**")
         .addResourceLocations("/resources/");
    registry.addResourceHandler("/assets/**")
         .addResourceLocations("/resources/assets/");
```

O void configureViewResolvers(ViewResolverRegistry registry) 삭제

CommonExceptionAdvice.java

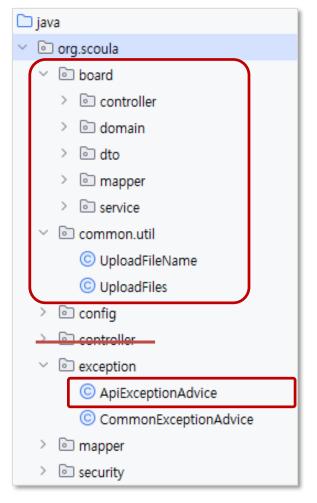
```
package org.scoula.exception;
import lombok.extern.log4j.Log4j2;
import org.springframework.core.annotation.Order;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.NoHandlerFoundException;
@ControllerAdvice
@Log4j2
@Order(1)
public class CommonExceptionAdvice {
   @ExceptionHandler(NoHandlerFoundException.class)
    public String handle404(NoHandlerFoundException ex) {
        return "/resources/index.html";
```

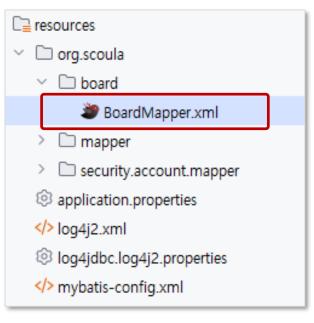
ApiExceptionAdvice.java

```
package org.scoula.exception;
import org.springframework.core.annotation.Order;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.RestControllerAdvice;
import java.util.NoSuchElementException;
@RestControllerAdvice
@order(2)
public class ApiExceptionAdvice {
```

게시판(board) api 준비

- o boardapi 프로젝트에서 org.scoula.board 패키지, common 패키지 복사
- o exception.ApiExceptionAdvice.java, BoardMapper.xml 복사





게시판(board) api 준비

- o SwaggerConfig.java 복사
 - JWT 인증 처리 추가



config.SwaggerConfig.java

```
public class SwaggerConfig {
   @Bean
    public Docket api() {
        return new Docket(DocumentationType.SWAGGER_2)
               // JWT 인증
                .securityContexts(List.of(this.securityContext())) // SecurityContext 설정
                .securitySchemes(List.of(this.apiKey())) // ApiKey 설정
                .select()
                .apis(RequestHandlerSelectors.withClassAnnotation(RestController.class))
                .paths(PathSelectors.any())
                .build()
                .apiInfo(apiInfo());
```

config.SwaggerConfig.java

```
// JWT SecurityContext 구성
private SecurityContext securityContext() {
    return SecurityContext.builder()
            .securityReferences(defaultAuth())
            .build();
private List<SecurityReference> defaultAuth() {
    AuthorizationScope authorizationScope = new AuthorizationScope("global", "accessEverything");
    AuthorizationScope[] authorizationScopes = new AuthorizationScope[1];
    authorizationScopes[0] = authorizationScope;
    return List.of(new SecurityReference("Authorization", authorizationScopes));
// ApiKey 정의
private ApiKey apiKey() {
    return new ApiKey("Authorization", "Authorization", "header");
```

config.ServletConfig.java

```
public class ServletConfig implements WebMvcConfigurer {
   @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
       // Swagger UI 리소스를 위한 핸들러 설정
        registry.addResourceHandler("/swagger-ui.html")
                .addResourceLocations("classpath:/META-INF/resources/");
       // Swagger WebJar 리소스 설정
        registry.addResourceHandler("/webjars/**")
                .addResourceLocations("classpath:/META-INF/resources/webjars/");
       // Swagger 리소스 설정
        registry.addResourceHandler("/swagger-resources/**")
                .addResourceLocations("classpath:/META-INF/resources/");
        registry.addResourceHandler("/v2/api-docs")
                .addResourceLocations("classpath:/META-INF/resources/");
```

config.WebConfig.java

✓ resoruces::/mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration</pre>
        PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <settings>
        <setting name="mapUnderscoreToCamelCase" value="true" />
    </settings>
    <typeAliases>
        <package name="org.scoula.security.account.domain" />
        <package name="org.scoula.board.domain" />
    </typeAliases>
</configuration>
```

config.RootConfig.java

```
@Configuration
@PropertySource({"classpath:/application.properties"})
@MapperScan(basePackages = {"org.scoula.board.mapper"})
@ComponentScan(basePackages = {"org.scoula.board.service"})
@Log4j2
@EnableTransactionManagement
public class RootConfig {
    ...
}
```

config.ServletConfig.java

```
@EnablewebMvc
@ComponentScan(basePackages = {
         "org.scoula.controller",
         "org.scoula.exception",
         "org.scoula.board.controller"
})
public class ServletConfig implements WebMvcConfigurer {
         ...
}
```

security.config.SecurityConfig.java

security.config.SecurityConfig.java

```
@Override
public void configure(HttpSecurity http) throws Exception {
   http
       .authorizeRequests()
       .antMatchers(HttpMethod.OPTIONS).permitAll()
        // 일단 모든 접근 허용
       .anyRequest().permitAll();
   http.httpBasic().disable() // 기본 HTTP 인증 비활성화
       .csrf().disable()
                        // CSRF 비활성화
       .formLogin().disable() // formLogin 비활성화 관련 필터 해제
       .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS); // 세션 생성 모드 설정
```

