

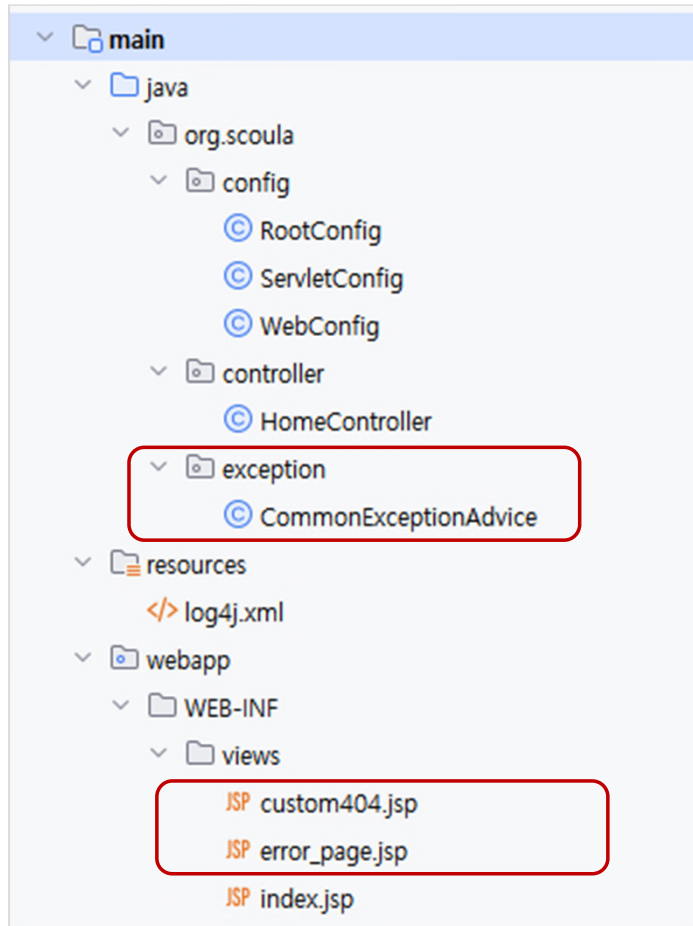
2025년 상반기 K-디지털 트레이닝

SpringLegacy 업데이트

[KB] IT's Your Life

1 SpringLegacy 업데이트

✓ 기존 SpringLegacy 프로젝트에서 작업



예외처리하는 거

1 SpringLegacy 업데이트

ServletConfig.java

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

    //      Servlet 3.0 파일 업로드 사용시
    @Bean
    public MultipartResolver multipartResolver() {
        StandardServletMultipartResolver resolver = new StandardServletMultipartResolver();
        return resolver;
    }
}
```

멀티파트 처리

1 SpringLegacy 업데이트

WebConfig.java

```
...
@Slf4j
@Configuration
public class WebConfig extends AbstractAnnotationConfigDispatcherServletInitializer {
    final String LOCATION = "c:/upload";
    final long MAX_FILE_SIZE = 1024 * 1024 * 10L;
    final long MAX_REQUEST_SIZE = 1024 * 1024 * 20L;
    final int FILE_SIZE_THRESHOLD = 1024 * 1024 * 5;;

    ...

    @Override
    protected void customizeRegistration(ServletRegistration.Dynamic registration) {
        registration.setInitParameter("throwExceptionIfNoHandlerFound", "true");
        MultipartConfigElement multipartConfig =
            new MultipartConfigElement(
                LOCATION, // 업로드 처리 디렉토리 경로
                MAX_FILE_SIZE, // 업로드 가능한 파일 하나의 최대 크기
                MAX_REQUEST_SIZE, // 업로드 가능한 전체 최대 크기(여러 파일 업로드 하는 경우)
                FILE_SIZE_THRESHOLD // 메모리 파일의 최대 크기(이보다 작으면 실제 메모리에서만 작업)
            );
        registration.setMultipartConfig(multipartConfig);
    }
}
```

1 SpringLegacy 업데이트

CommonExceptionAdvice.java

```
package org.scoula.exception;
...

public class CommonExceptionAdvice {
    @ExceptionHandler(Exception.class)
    public String except(Exception ex, Model model) {
        log.error("Exception ..... " + ex.getMessage());
        model.addAttribute("exception", ex);
        log.error(model);
        return "error_page";
    }

    @ExceptionHandler(NoHandlerFoundException.class)
    @ResponseStatus(HttpStatus.NOT_FOUND)
    public String handle404(NoHandlerFoundException ex) {
        log.error(ex);
        model.addAttribute("uri", request.getRequestURI());
        return "custom404";
    }
}
```

1 SpringLegacy 업데이트

views/error_page.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" import="java.util.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h4><c:out value="${exception.getMessage()}"></c:out></h4>

<ul>
<c:forEach items="${exception.getStackTrace()}" var="stack">
<li><c:out value="${stack}"></c:out></li>
</c:forEach>
</ul>

</body>
</html>
```

1 SpringLegacy 업데이트

views/custom404.jsp

```
<!DOCTYPE html>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
  <h1>해당 URL(${uri})은 존재하지 않습니다.</h1>
</body>
</html>
```

1 SpringLegacy 업데이트

✓ SpringLegacy 템플릿 다시 생성

- 기존 템플릿 덮어쓰기

끝!

6/10

스프링에서는 이런 코드를 다 쳤지만
스프링 부트는 다 설정으로 빠졌다.