#### 12장. 파일 업로드, 다국어 처리

Security - login



		글쓰기
	제목	title
즈	박성자 -	cloud
	내용	content
ē	<b></b> 로드	파일 선택 선택된 파일 없음
		등록 목록



#### ◆ MultipartFile 객체 필드 추가

```
@Setter
@Getter
public class BoardVO implements Serializable {
    private static final long serialVersionUID = 30L;
    private int bno;
    private String title;
    private String writer;
    private String content;
   @DateTimeFormat(pattern="yyyy-MM-dd")
    private Date regDate;
    private int cnt;
    private MultipartFile uploadFile; //파일 업로드
```



#### ◆ FileUpload 라이브러리 추가

- > Macommons-io-2.2.jar C:\Users\U
- > Macrest-core-1.3.jar C:\Users\kiyon\.m2\repository\org\hamcrest\hamcre



#### ◆ MultipartResolver 빈 등록



#### ◆ 파일 업로드 처리

```
@PostMapping("/insertBoard")
@PreAuthorize("isAuthenticated()")
public String insertBoard(BoardVO vo) throws IllegalStateException, IOException{
    //파일 업로드 처리
    MultipartFile uploadFile = vo.getUploadFile();
    if(!uploadFile.isEmpty()) {
         String fileName = uploadFile.getOriginalFilename();
         String filePath = "C:/springwork/SpringBoard2/src/main/webapp/upload/";
         uploadFile.transferTo(new File(filePath + fileName));

√ № > src

    service.insert(vo);
                                                              ∨ 👫 > main
    return "redirect:/board/boardList";
                                                                > 🔄 > resources

√ Ø<sub>?</sub> > upload

                                                                     dish1.jpg

√ № > WEB-INF

                                                                     classes

√ 🔓 > spring

                                                                      A appServlet
                                                                      ₽ message-context.xml
                                                                      > root-context.xml
                                                                      > security-context.xml
```



#### ◆ 글쓰기 페이지

- 업로드 항목 추가
- form 속성 추가 : encType="multipart/form-data"



#### ◆ 다국어 처리

다국어 지원이란 '국제화'라고도 하며 하나의 JSP 페이지를 다양한 언어로 서비스하는 것을 의미한다.

		LOGIN	
			Korean   English
	ID		
	PASSWORD		
		LOG-IN	



#### ◆ 메시지 파일 작성하기

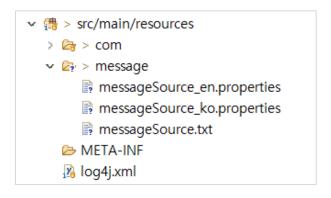
메시지 파일의 확장자는 .properties이며, 파일명은 언어에 해당하는 Locale정보를 결합하여 작성한다.

messageSource\_en.properties

```
#customLogin.jsp

message.user.login.title=LOGIN
message.user.login.id=ID
message.user.login.password=PASSWORD
message.user.login.loginBtn=LOG-IN

message.user.login.language.en=English
message.user.login.language.ko=Korean
```





#### ◆ 메시지 파일 작성하기

#### messageSource.txt

```
message.user.login.title=로그인
message.user.login.id=아이디
message.user.login.password=비밀번호
message.user.login.loginBtn=로그인
message.user.login.language.en=영어
message.user.login.language.ko=한글
```

messageSource\_ko.properties

```
message.user.login.title=\uB85C\uADF8\uC778
message.user.login.id=\uC544\uC774\uB514
message.user.login.password=\uBE44\uBC00\uBC88\uD638
message.user.login.loginBtn=\uB85C\uADF8\uC778

message.user.login.language.en=\uC601\uC5B4
message.user.login.language.ko=\uD55C\uAE00
```

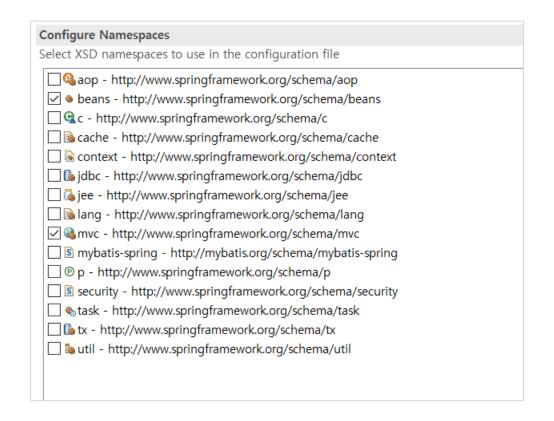


#### ◆ MessageSource 등록

```
<!-- 다국어 관련 설정 -->
                                                              message-context.xml
<!-- MessageSource 등록 -->
<bean id="messageSource"</pre>
      class="org.springframework.context.support.ResourceBundleMessageSource">
    property name="basenames">
        t)
            <value>message.messageSource</value>
        </list>
    </property>
</bean>
<!-- LocaleResolver 등록 -->
<bean id="LocaleResolver"</pre>
      class="org.springframework.web.servlet.i18n.SessionLocaleResolver"/>
<!-- LocaleChangeInteerceptor 등록 -->
<mvc:interceptors>
    <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
        cproperty name="paramName" value="Lang" />
    </bean>
</mvc:interceptors>
```



#### ◆ MessageSource 등록





◆ web-xml에 message-context.mxl 등록



#### ◆ 로그인 화면 처리

```
태그 라이브러리 지시자 선언
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring" %>
<spring:message code= " 메시지 키값" />
```

```
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring" %>
<!DOCTYPE html>
                                                                customLogin.jsp
⊝<html>
⊖ <head>
<meta charset="UTF-8">
<title><spring:message code="message.user.login.title" /></title>
<link rel="stylesheet" href="/resources/css/style.css">
</head>
<jsp:include page="./menu.jsp" />
    <div id="container">
        <section id="login">
            <div class="title">
                 <h2><spring:message code="message.user.login.title" /></h2>
             </div>
```



```
>
   <a href="customLogin?lang=en">
      <spring:message code="message.user.login.language.en" />
   </a>
   <a href="customLogin?lang=ko">
                                                              customLogin.jsp
      <spring:message code="message.user.login.language.ko" />
   </a>
<h3 class="error"><c:out value="${error}" /></h3>
<form action="/login" method="post">
   >
          <spring:message code="message.user.login.id" />
          <input type="text" name="username">
      <spring:message code="message.user.login.password" />
          <input type="password" name="password">
      <input type="submit" value="<spring:message code="message.user.login.loginBtn</pre>
          <input type="hidden" name="${_csrf.parameterName}" value="${ csrf.token}" />
```

