

□ 서비스 개요

- 전자정부 표준 프레임워크에서는 Spring MVC 에서 제공하는 **LocaleResolver**를 이용한다.
- Spring MVC 는 다국어를 지원하기 위하여 아래와 같은 종류의 LocaleResolver 를 제공하고 있다.
 - **CookieLocaleResolver** : 쿠키를 이용한 locale정보 사용
 - **SessionLocaleResolver** : 세션을 이용한 locale정보 사용
 - **AcceptHeaderLocaleResolver** : 클라이언트의 브라우저에 설정된 locale정보 사용

* Bean 설정 파일에 정의하지 않을 경우 **AcceptHeaderLocaleResolver** 가 default 로 적용된다.

❑ CookieLocaleResolver

- CookieLocaleResolver 를 설정하는 경우 사용자의 쿠키에 설정된 Locale 을 읽어들인다.
- Bean 설정정보(xx-servlet.xml)

```
<bean id= "localeResolver"
  class= "org.springframework.web.servlet.i18n.CookieLocaleResolver" >
  <property name= "cookieName" value= "clientlanguage"/>
  <property name= "cookieMaxAge" value= "100000"/>
  <property name= "cookiePath" value= "web/cookie"/>
</bean>
```

- Property 속성

속성	기본값	설명
cookieName	classname + locale	쿠키명
cookieAge	nteger.MAX_INT	-1 로 해두면 브라우저를 닫을 때 없어짐
cookiepath	/	Path 를 지정하면 해당하는 Path와 그 하위 Path 에서만 참조

❑ SessionLocaleResolver

- request가 가지고 있는 **Session**으로 부터 locale 정보를 가져온다.
- Bean 설정정보(xx-servlet.xml)

```
<bean id= 'localeResolver "  
    class="org.springframework.web.servlet.i18n.SessionLocaleResolver" />
```

❑ AcceptHeaderLocaleResolver

- 사용자의 브라우저에서 보내진 request 의 헤더에 accept-language 부분에서 Locale 을 읽어들인다. 사용자 브라우저의 Locale 을 나타낸다.
- Bean 설정정보(xx-servlet.xml)

```
<bean id= 'localeResolver "  
    class="org.springframework.web.servlet.i18n.AcceptHeaderLocaleResolver" />
```

❑ 샘플예제 (SessionLocaleResolver)

– Web Configuration –Filter 설정

- Web 을 통해 들어오는 요청을 Charset UTF-8 적용

```
<filter>
  <filter-name>encoding-filter</filter-name>
  <filter-class>
    org.springframework.web.filter.CharacterEncodingFilter
  </filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>encoding-filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

- Spring Configuration (xx-servlet.xml)
 - 아래와 같이 localeResolver 와 localeChangeInterceptor 를 등록하고 Annotation 기반에서 동작할 수 있도록 **DefaultAnnotationHandlerMapping** 에 interceptor 로 등록을 해준다.
 - SessionLocaleResolver 를 이용하여 위와 같이 하였을 경우 Locale 결정은 **language** Request Parameter 로 넘기게 된다.

```
<bean id="localeResolver"
class="org.springframework.web.servlet.i18n.SessionLocaleResolver" />

<!--
쿠키를 이용한 Locale 이용시 <bean id="localeResolver"
class="org.springframework.web.servlet.i18n.CookieLocaleResolver"/>
-->
<bean id="localeChangeInterceptor"
class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
<property name="paramName" value="language" />
</bean>

<bean id="annotationMapper"
class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping">
<property name="interceptors">
<list>
<ref bean="localeChangeInterceptor" />
</list>
</property>
</bean>
```

– Message Source 설정

- 먼저 다국어를 지원해야 하므로 메시지를 MessageSource 로 추출하여 구현해야 한다.
자세한 MessageSource 내용은 Resource 를 참고하길 바란다.
messageSource는 아래와 같이 설정하였다.

```
<bean id="messageSource"
      class="org.springframework.context.support.ResourceBundleMessageSource">
  <property name="basenames">
    <list>
      <value>classpath:/message/message</value>
    </list>
  </property>
</bean>
```

- message properties 파일은 아래와 같다.
locale에 따라 ko, en 으로 구분하였다.
message_ko.properties 파일 : view.category=카테고리
message_en.properties 파일 : view.category=category

– JSP

- Spring Message Tag 이용 : `<spring:message code="view.category" />`

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>

<form:form commandName="message" >
....
    <table border="1" cellspacing="0" class="boardList" summary="List of Category">
        <thead>
            <tr>
                <th scope="col">No.</th>
                <th scope="col">
                    <input name="checkAll" type="checkbox" class="inputCheck" title="Check All"
onclick="javascript:fncCheckAll();" />
                </th>
                <th scope="col"><spring:message code="view.category" /> ID</th>
                <th scope="col"><spring:message code="view.category" /> 명</th>
                <th scope="col">사용여부</th>
                <th scope="col">Description</th>
                <th scope="col">등록자</th>
            </tr>
        </thead>
    </table>
....
```

– 결과

화면상으로 해당 페이지를 실행해보면 아래와 같다.

한글인 경우 :

http://localhost:8080/sample-web/sale/listCategory.do?language=ko

No.	<input type="checkbox"/>	카테고리 ID	카테고리 명	사용여부	
1	<input type="checkbox"/>	0000000001	Sample Test	Y	This is initial test data,
2	<input type="checkbox"/>	0000000002	test Name	Y	test Desc

Manage Checked List All << End < Prev 11 12

영어인 경우 :

http://localhost:8080/sample-web/sale/listCategory.do?language=en

No.	<input type="checkbox"/>	category ID	category 명	사용여부	
1	<input type="checkbox"/>	0000000001	Sample Test	Y	This is
2	<input type="checkbox"/>	0000000002	test Name	Y	test De

Manage Checked List All << Er

LAB 301-mvc 실습 (3)

Step 1-3-1. context-servlet.xml 추가 설정하기

```
<!-- [Exercise 1-3-3] mvc-servlet.xml 추가 설정하기 : SessionLocaleResolver를 이용한 locale 설정 -->
<!-- setting Locale -->
<bean id="localeChangeInterceptor"
      class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"
      p:paramName="lang" />

<bean id="localeResolver" class="org.springframework.web.servlet.i18n.SessionLocaleResolver" />

<!-- Locale Interceptor 설정하기 -->
<bean class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping">
  <property name="interceptors">
    <list>
      <ref bean="localeChangeInterceptor"/>
    </list>
  </property>
</bean>
```

Step 1-3-2. LoginController.java 에 @SessionAttributes 설정 하기

```
@Controller
@SessionAttributes("login")
public class LoginController {
  ...
}
```

Step 1-3-3. LoginController.java 추가 메소드 작성하기

```
@RequestMapping(value = "/memberInfo.do")
public ModelAndView memberInfo(HttpSession httpSession) {
    ModelAndView mav = new ModelAndView("login/memberInfo");

    if (httpSession.getAttribute("login") != null) {
        mav.addObject("login", httpSession.getAttribute("login"));
    }

    return mav;
}

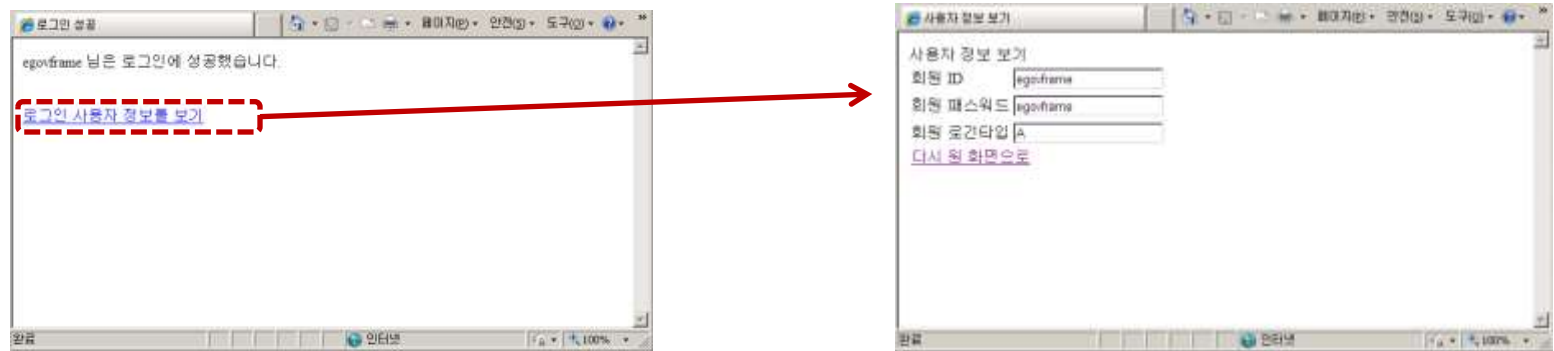
@RequestMapping(value = "/loginOut.do", method = RequestMethod.GET)
public String logOut(SessionStatus sessionStatus) {

    if (!sessionStatus.isComplete())
        sessionStatus.setComplete();

    return getFormView();
}
```

Exercise 1-3-4. . '사용자 정보보기' 예제 실행결과 확인

- 예제 실행 결과 확인

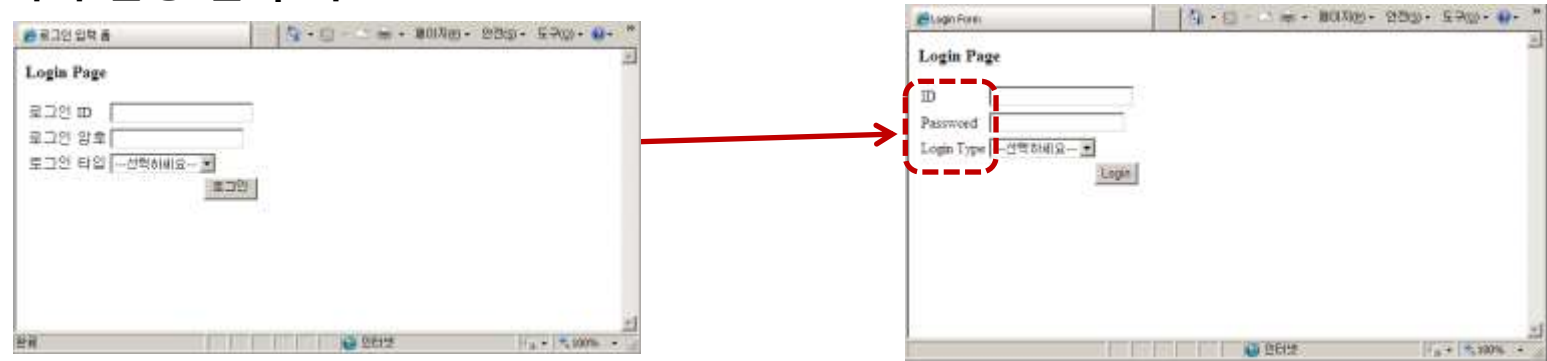


Exercise 1-3-5. 국제화 적용 결과 확인

- 로그인 페이지 locale 변경(en)

(<http://127.0.0.1:8080/lab301-mvc-tutor/loginProcess1.do?lang=en>)

- 예제 실행 결과 확인



❑ The Spring Framework - Reference Documentation

- <http://static.springsource.org/spring/docs/2.5.x/reference/>
- <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/>