WAY2LEARN

LAB- Spring Mvc - Using Themes

In this lab you will be understanding how to create an internatinalized spring mvc application.

STEP -1

In this step, you will be working on **03mvc-themes-start** project

The code Provided to you is same as the code in **07mvc-tiles-solution**

Firstly, let us try to understand what we want to achieve.

We want the user to be able to customize look and feel of the application according to his taste.

When we say "Look and feel" , hope you know that we can change look and feel just by changing the css applied.

So, we want to change the applied css dynamically based on user selection.

Go to /WEB-INF/resources/css folder and observe that there are 2 files with name **layout.css** and **layout-red.css**

Open them and observe that there is only one simple difference in both files in line number 5. In layout-red.css, we are applying **banner-repeat-red.png (red)** and in layout.css, we are applying **banner-repeat.png (blue)**

Now we want to apply either layout.css or layout-red.css dynamically based on user selection.

If you want to support 2 themes, u need to write 2 ".properties " files under classpath.
The file names should be same as the theme names you want to use.

So, observe blue.properties and red.properties under src/main/resources.

We want the theme names to be blue and red. So, we named them accordingly. Open them and observe that the key "layout.css" is having different values.

ThemeResolver will be used to resolve a theme. So, complete TODO-1 inside mvc-config.xml by configuring a CookieTheme resolver as shown below :

```
	<bean id="themeResolver"
class="org.springframework.web.servlet.theme.CookieThemeResolver">
		<property name="defaultThemeName" value="blue"/>
		<property name="cookieName" value="clientTheme" />
		<property name="cookieMaxAge" value="36000" />
	</bean>
```

Observe that the default the value is given as **blue.**

So, the above themeResolver will look for a cookie with name "clientTheme" and will determine the theme name based on its value.
If no cookie with name "clientTheme", the default theme is "blue".

Open baseLayout.jsp and observe that we are adding link to layout.css as shown below :

```
<spring:theme var="layoutCSS" code="layout.css"/>

<link rel="stylesheet" type="text/css" href=" ${layoutCSS}">
```

<spring:theme> tag handler will identify the current theme using the theme resolver and look into the corresponding ".properties" file for the code " *layout.css* " based on the theme name.

That value is used as the url for css.

So, the CSS URL value is fetched either from blue.properties or red.properties dynamically.

Deploy the application and give a request to /home.htm and observe that the blue theme is displayed by default as there will be no cookie initially.

Change the default theme name to red and observe that the default red theme only is displayed.

Who will actually create a cookie initially?

So, Configure w ThemeChangeInterceptor as shown below :

```
    <mvc:interceptors>
		<bean
class="org.springframework.web.servlet.theme.ThemeChangeInterceptor"/>
	</mvc:interceptors>
```

This ThemeChange interceptor will create a Cookie based on the value of a request parameter

with name "theme".
Open topMenu.jsp and observe that there are links for red and blue.

If you click on "red " link, a request parameter theme=red will be passed. Similarly for blue.

Now deploy the application and observe that initially the default theme "blue" is displayed because there are no cookies in the request.

Click on red Link and observe that red theme is applied.

Are you able to visualize how the themes are getting changed? If not, go and see my video once again !!

Congratulations !! You know how to apply themes to mvc application