LAB- Spring Mvc - Static Resource Handling

In this lab you will be understanding how to handle static resource requests efficiently.

STEP -1

In this step, you will be working on **04mvc-staticresourcehandling-start project**

1) Right now, if you observe, resources folder which contains images,css and js files is outside of WEB-INF. we want the resources folder to be inside WEB-INF.

Move resource folder inside WEB-INF

Deploy the application and give a request to home.htm. Observe that page is not displayed properly because css is not applied. Css files cannot be accessed as they are inside WEB-INF .

Now how to map any request mapping /resources/* to /WEB-INF/resources ?

Configure as below inside mvc-config.xml

```
<mvc:resources location="/WEB-INF/resources/" mapping="/resources/**"
        cache-period="600" />
```

Do you remember what is going on in the background once you configure this? if not See the video once again.

Deploy the application and make sure the everything works fine even the resources are inside WEB-INF folder.

2) Configuring < mvc:resources /> as shown above will also send response headers to the browser to cache the static files for 600 seconds as per above configuration.

But what if resource is changed before the cache expiry? How to handle them?

Just open application.properties present inside WEB-INF. It looks like below :

```
application.version=1.0.2
```

configure as shown below in mvc-config.xml

```
<util:properties id="applicationProps" location="/WEB-INF/application.properties" />

        <mvc:resources location="/WEB-INF/resources/"
mapping="/resources-#{applicationProps['application.version']}/**"
        cache-period="600" />
```

Open home.jsp and write as below just above the first <link> tag inside <head> tag

```
<spring:eval expression="@applicationProps['application.version']"
var="applicationVersion"/>

<spring:url value="resources-{applicationVersion}" var="resourceUrl" />
```

Inside every <link> tag's href replace resources with ${resourceUrl} as show below :

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


<link rel="shortcut icon" type="image/x-icon" href="${resourceUrl}/images/favicon.ico" />
```

Similary , changes are already done for you in viewIndividualCourse.jsp and browseCoursesmain.jsp

Deploy the application and make sure everything works as expected.

To Verify whether the server sends some caching headers when a request is given to a static file, give a request to common.css from a rest client on you browser and observe the response headers . (See my video how i am verifying the same)

Congratulations !! You know how to create web application using Spring MVC !!