

J2EE

Day5

Recap Step1

- We installed Spring framework
- We created a configuration for our application
- We added oracle drivers
- We installed mysql database
- We told Spring plugin about spring related config files
- We learnt about Beans
- We saw how Eclipse created icons for Beans

Recap Step2

- We saw that `src/main/webapp` contains Spring related files
- We created a `css` folder under `src/main/webapp`
- We created a folder called `flows` under `src/main/webapp`
- We created a Spring Web Flow Definition file called `main-flow.xml` under `src/main/webapp/flows`

Recap Step3

- We configured main-flow.xml
- We created a welcome.xhtml view file in src/main/webapp/flows/main/
- We added the following namespaces to welcome.xhtml

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui"
    template="/WEB-INF/templates/general.xhtml">
```

Recap Step4

- We created a template under `src/main/webapp/WEB-INF` under a folder called `templates`
- We created a template file called `general.xhtml` under `src/main/webapp/WEB-INF`
- We created a `welcome.xhtml` file under `src/main/webapp/flows/main/welcome.xhtml`

Recap Step5

We setup a re-direct from index.jsp to app/main as follows :

```
<head>
```

```
<meta http-equiv="Refresh" content="2; URL=app/main" />
```

```
<title>Insert title here</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Setup MySQL DB in Eclipse

Download MySQL Connector/J from

<http://dev.mysql.com/downloads/connector/j/>

Extract the zip to your home folder.

Setup Database Connection in MySQL

Go to Window -> Show View -> Data Source Explorer

Right click on Database Connections -> New

Select the Database to MySQL under New Connection Profile

Give it the name LocalMySQL and click Next

Specify a driver

Click the New Driver Definition icon next to the Drivers listbox

On the popup window, select the Database version in Name/Type tab : 5.1

Switch to the Jar List tab and click Clear All to clear the default false jar file location

Point to Database jar file

Locate the jar file inside the folder that you extracted from the downloaded MySQL Connector/J zip file

Modify the database connection setting according to your MySQL Database configuration and click OK

Specify Connection Details

Change database name to j2eedemo_schema

The JDBC URL format for MySQL Connector /J is as follows :

```
jdbc:mysql://[host:port],[host:port].../  
[database][?propertyName1]=[propertyValue1]  
[&propertyName2]=[propertyValue2]...
```

Change URL to jdbc:mysql://localhost:
3306/j2eedemo_schema

and click on Test Connection

Finish Database Configuration

Click Finish to close the setup wizard.

You should see a Database icon in the Database Connections folder.

Configure datasource-config

Comment out old dataSource block and add the following block to datasource-config :

```
<bean name="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://localhost:3306/j2eedemo_schema" />
    <property name="username" value="root" />
    <property name="password" value="" />
</bean>
```

mysql-connector-java

Switch to the connectors/mysql-connector-java-5.1.26 which has the mysql-connector-java.jar and run the following command :

```
mvn install:install-file -Dfile=mysql-connector-java-5.1.26-bin.jar -DgroupId=mysql -DartifactId=mysql-connector-java -Dversion=5.1.26-bin -Dpackaging=jar
```

Add the following dependency to pom.xml of the project :

```
<!-- MySql 5.5 Connector -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.26</version>
</dependency>
```


Maven clean, install and deploy

Deploy to make sure the application is able to be built and deployed

Step 1

UIs consist of 2 parts :

- Templates

- Compositions

general.xhtml is the template

and

welcome.xhtml is the composition which uses the above template

Creating UIs with Facelets, Templates, JSF, Primefaces and Webflow

Step2

Clean up welcome.xhtml by removing the following line :

```
<h:outputText value="JSF Works Great" />
```

Clean up general.xhtml by removing the following :

```
<h1>Webflow and Facelets +  
and </h1>
```

Step3

A view can be constructed from several templates

We have the concept of multiple inheritance in views, i.e. a view can inherit from several templates

Creating UIs with Facelets, Templates, JSF, Primefaces and Webflow

Step4

Add the following above `</h:head>` in general.xhtml

```
<link rel="stylesheet" type="text/css" href="${request.contextPath}/resources/css/style.css" />
```

Create a file called style.css under src/main/webapp/css

Any content or container with the tag `<ui:insert>` can addressed from the compositions

Creating UIs with Facelets, Templates, JSF, Primefaces and Webflow

Step5

Remove `<ui:insert name="body" />`

and

Add the following above `</h:body>` in `general.xhtml` :

```
<div id="container">
    <div id="header">
        <ui:insert name="header" />
    </div>
    <div id="content">
        <ui:insert name="content" />
    </div>
    <div id="footer">
        <ui:insert name="footer" />
    </div>
</div>
```

Step6

Add the following style to style.css :

```
#container
{
    width: 80%;
    height: 450px;
    margin: 50px auto;
    padding: 20px 20px;
    box-shadow : 2px 2px 10px #888;
}
```

Creating UIs with Facelets, Templates, JSF, Primefaces and Webflow

Step7

Copy images from tempfiles/css/images and paste it under src/main/webapp/css/

Then modify style.css as follows :

Clean, install and deploy

Creating UIs with Facelets, Templates, JSF, Primefaces and Webflow

Step8

Don't Add `<h1>Please Sign In</h1>` below `<div id="header">` in `general.xhtml`

Add the following style to `style.css` :

```
padding-bottom: 50px;
```

```
}
```

```
#header
```

```
{
```

```
    height:60px;
```

```
    text-align: center;
```

```
    background-color: #525252;
```

```
    color: white;
```

```
    padding: 5px;
```

```
}
```

Clean, install and deploy