Module 3-3

Introduction to MVC - Views

Spring

The Spring Framework is a very popular framework that abstracts various aspects of application development in order to create robust web applications faster.

A strong grasp of basic Java is needed to fully take advantage of Spring!

Technology Check

Anatomy of a Spring Project

- POM.xml: The pom.xml file specifies classes that the project might need from an internet repository.

- The src/main folder: This is where the vast majority of your code will go, and it's further subdivided into several sections:

    - The **java** folder - this is where all the Java code will go.
    - The **webapp/WEB-INF folder** - this is where your JSP code will go to (more on this later today!)

Model View Controller Definition

Model View Controller (MVC) is a philosophy of application development that divides the application based on responsibilities.

There are many frameworks capable of delivering a MVC application.

Model View Controller Definition

Model

- application state and business logic
- only part of the application that talks to the database
- Models could be classes

View

- presents data to user
- accepts input from user
- Views might be a desktop display, a mobile display, a file output based on a model

Controller

- Takes input from the view and passes it to the appropriate model objects
- Grabs all necessary building blocks and organizes them for the output
- Takes results from the model and passes it to the appropriate view

Model View Controller Definition

In the coming days we will be learning to implement MVC styled web application, but it need not be only for web applications - the MVC philosophy can be applied to mobile apps and desktop apps as well!

Model View Controller, why we need views

Amazon has 606 million products on sale, does it make sense to maintain the same number of HTML files?

Views: The JSP

Java Server Pages (JSP) are a way to dynamically generate HTML. There are two types of JSP components:

- **Expression Language (EL)**: allows for the use of Java expressions within a JSP file, must be enclosed in ${...}
- **Java Standard Tag Library (JSTL) Tag**s: Specialized tags that make implementing dynamic behavior easier.

A functional JSP page could contain a combination of JSTL, EL, and regular HTML.

Views: Common JSTL Tags

- **<c:set />**
  - used for setting "scoped variables"
- **<c:if />**
  - like an `if` statement from Java but without `else`
- **<c:choose />**
  - used when we have `if ... else if ... else` logic (i.e. when there are multiple decision branches)
- **<c:foreach />**
  - `begin` and `end` attributes can be used to iterate over a series of numbers
  - can also be used to iterate over a collection

Let’s do some coding!

JSTL Example: <c:set/>

Consider the following example:

```
<c:set var="var1" value="a" />
<c:set var="var2" value="79" />
<c:set var="var3" value="10" />

${var1}
${var2}
${var3}
${var2 + var3}
```

On the rendered HTML page, the output will be:
a 79 10 89

JSTL Example: <c:if/>

Consider the following example:

```
<c:set var = "scale" value="F"/>

<c:if test = "${scale == 'F'}" >
        <p>Human body temperature is 98.6</p>
</c:if>
```

On the rendered HTML page, the output will be:
Human body temperature is 98.6

JSTL Example: <c:choose/>

Consider the following example:

```
<c:set var = "scale" value="O"/>
<c:choose>

        <c:when test="${scale == 'F'}" >
                <p>Human body temperature is 98.6 F.</p>
        </c:when>

        <c:when test="${scale == 'C'}" >
                <p>Human body temperature is 37.0 C.</p>
        </c:when>

        <c:when test="${scale == 'K'}" >
                <p>Human body temperature is 310.15 K.</p>
        </c:when>

        <c:otherwise>
                <p>Bad Scale</p>
        </c:otherwise>
</c:choose>
```

On the rendered HTML page, the output will be:
Bad Scale

JSTL Example: <c:forEach/>

Consider the following example:

```
<c:set var="var3" value="10" />

<c:forEach var="i" begin="1" end="${var3}">
    <c:out value="${i}" />
</c:forEach>
```

On the rendered HTML page, the output will be:
1 2 3 4 5 6 7 8 9 10