Your Name
Your Title
Your Organization Line 1
Your Organization Line 2

Spring-mvc, step2

Course Title
Course Description

JSTL

- will be using the JSP Standard Tag Library (JSTL)
- Header file, included in all jsp files
- WEB-INF/, views accessed via the controller, not directly
- /WEB-INF/jsp/include.jsp
  - <%@ page session="false"%>
  - <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
  - <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
- Until JSTL is ready => using tags (redirect)
- All requests for index.jsp will go throught our application framework
  - <%@ include file="/WEB-INF/jsp/include.jsp" %>
  - <%-- Redirected because we can't set the welcome page to a virtual URL. --%>
  - <c:redirect url="/hello.htm"/>

hello.jsp

```
<%@ include file="/WEB-INF/jsp/include.jsp" %>
<html>
  <head><title>Hello :: Spring Application</title></head>
  <body>
    <h1>Hello - Spring Application</h1>
    <p>Greetings, it is now <c:out value="${now}"/></p>
  </body>
</html>
```

HelloControllerTest.java update

```
package springapp.web;
import org.springframework.web.servlet.ModelAndView;
import springapp.web.HelloController;
import junit.framework.TestCase;
public class HelloControllerTests extends TestCase {
    public void testHandleRequestView() throws Exception{
        HelloController controller = new HelloController();
        ModelAndView modelAndView = controller.handleRequest(null, null);
        assertEquals("WEB-INF/jsp/hello.jsp", modelAndView.getViewName());
        assertNotNull(modelAndView.getModel());
        String nowValue = (String) modelAndView.getModel().get("now");
        assertNotNull(nowValue);
    }
}
```

HelloController.java

```
package springapp.web;
import org.springframework.web.servlet.mvc.Controller;
import org.springframework.web.servlet.ModelAndView;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import java.io.IOException;
import java.util.Date;
public class HelloController implements Controller {
    protected final Log logger = LogFactory.getLog(getClass());
    public ModelAndView handleRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String now = (new Date()).toString();
        logger.info("Returning hello view with " + now);
        return new ModelAndView("WEB-INF/jsp/hello.jsp", "now", now);
    }}
```

Decouple view from controller

- Map to the view using a logical name, allowing us to switch the view without having to change the controller
- Use *ResourceBundleViewResolver* and a *SimpleUrlHandlerMapping*
- To map view to location *InternalResourceViewResolver, set prefix, suffix*
- Internatialization - *JstlView*, it will enable us to use JSTL in combination with message resource bundles, support for internationalization

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">
    <!-- the application context definition for the springapp DispatcherServlet -->
    <bean name="/hello.htm" class="springapp.web.HelloController"/>
    <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"></property>
        <property name="prefix" value="/WEB-INF/jsp/"></property>
        <property name="suffix" value=".jsp"></property>
    </bean>
```