



Spring-mvc, step1

Your Name

Your Title

1st Line of Your Organization

2nd Line of Your Organization

2005-12-31

Course Name

More Description About the Course

Spring base information

- Rod Johnson
- Inversion of Control (IoC)
- Aspect-Oriented Programming (AOP)
- From version 2.5 is supported annotations
- Setter, constructor, interface injection
- Advantages
 - Easy to use
 - Making simple tasks easy, hard things possible
 - POJOs (Plain Old Java Objects)
 - Everything should be POJOs
- Links
 - <http://weblog.morosystems.cz/spring>

DispatcherServlet

- Also know as Front Controller (define where are requests are routed base on information, which come later)
- DispatcherServlet* - Icontrol where all our requests are routed
- This servlet definition also has an attendant `<servlet-mapping/>` entry that maps to the URL patterns that we will be using
 - Any URL with an 'htm' extensions be routed to springapp
 - `<servlet-mapping>`
 - `<servlet-name>springapp</servlet-name>`
 - `<url-pattern>*.htm</url-pattern>`
 - `</servlet-mapping>`
- PageController (Fowler)
- Default *HandlerMapping* that the *DispatcherServlet* uses is the *BeanNameUrlHandlerMapping*
- *-servlet.xml contains beans definitons

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" >
  <servlet>
    <servlet-name>springapp</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>springapp</servlet-name>
    <url-pattern>*.htm</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>
      index.jsp
    </welcome-file>
  </welcome-file-list>
</web-app>
```


springapp-servlet.xml

- Contains bean definitions (POJOs) used by DispatcherServlet, WebApplicationContext - all web-related component go there
- Name is determined by the value of the <servlet-name/> element from the 'web.xml', with '-servlet' appended to it

```
<?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-2.5.xsd">
```

```
    <bean name="/hello.htm" class="springapp.web.HelloController"/>
```

```
</beans>
```


HelloController.java (1/2)

- defines the controller that our application will be using to service a request with the corresponding URL mapping of '/hello.htm'
- uses an implementation class of the interface called *HandlerMapping* to define the mapping between a request URL and the object that is going to handle that request (the handler)
- HelloController is responsible for handling a request for a particular page of the website (Fowler)
- Controller handles the request and returns a ModelAndView - in this case, one named '*hello.jsp*'

HelloController.java (2/2)

```
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;

public class HelloController implements Controller {
    protected final Log logger = LogFactory.getLog(getClass());

    public ModelAndView handleRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        logger.info("Returning hello view");
        return new ModelAndView("hello.jsp");
    }
}
```


Test for HelloController

- Agile development
- During development, not after
- Test packages

```
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("hello.jsp", modelAndView.getViewName());  
    }  
}
```


🔴 Copy to directory /WEB-INF/

```
<%@ page import="java.util.Map" %>
<html>
  <head>
    <title>Spring example</title>
  </head>
  <body>
    <h1><%=((Map) (request.getAttribute("model"))).get("pozdrav")%></h1>
  </body>
</html>
```

🔴 Run <http://localhost:8080/springapp/hello.htm>

Thank you