Your Name
Your Title
1st Line of Your Organization
2nd Line of Your Organization

Spring mvc - step4

Course Name
More Description About the Course

2005-12-31

Lecture Outline

- Rename *HelloController* to *InventroryController* (IC)
- To IC add reference to *ProductManager* class
- Add code to have the controller pass some product information to the view
- *getModelAndView()* returns Map with both date and time, and the products list obtained from manager reference

InventoryController

```
public class InventoryController implements Controller {
    protected final Log logger = LogFactory.getLog(getClass());
    private ProductManager productManager;
    public ModelAndView handleRequest(HttpServletRequest request,
     HttpServletResponse response)
            throws ServletException, IOException {
        String now = (new java.util.Date()).toString();
        logger.info("returning hello view with " + now);
        Map<String, Object> myModel = new HashMap<String, Object>();
        myModel.put("now", now);
        myModel.put("products", this.productManager.getProducts());
        return new ModelAndView("hello", "model", myModel);
    }
    public void setProductManager(ProductManager productManager) {
        this.productManager = productManager;
    }}
```

Message bundle

- hello.jsp
- Adding JSTL <c:forEach/> tag

```
<%@ include file="/WEB-INF/jsp/include.jsp" %>
<html>
  <head><title><fmt:message key="title"/></title></head>
  <body>
    <h1><fmt:message key="heading"/></h1>
    <p><fmt:message key="greeting"/> <c:out value="${model.now}"/></p>
    <h3>Products</h3>
    <c:forEach items="${model.products}" var="prod">
      <c:out value="${prod.description}"/> <i>$<c:out value="${prod.price}"/></i><br><br>
    </c:forEach>
  </body>
</html>
```

Populate business objects

- add a SimpleProductManager to our configuration file and to pass that into the setter of the InventoryController
- Data - bean entries in 'springapp-servlet.xml'
- 'messageSource' bean entry that will pull in the messages resource bundle ('messages.properties')
- Create dir /WEB-INF/classes/
- Create file /WEB-INF/clases/messages.properties with following context
    - *title=SpringApp*
    - *heading=Hello :: SpringApp*
    - *greeting=Greetings, it is now*

Springapp-servlet.xml (1/)

```
<?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">
    <!-- the application context definition for the springapp DispatcherServlet -->
    <bean id="productManager" class="springapp.service.SimpleProductManager">
        <property name="products">
            <list>
                <ref bean="product1"/>
            </list>
        </property>
    </bean>
    <bean id="product1" class="springapp.domain.Product">
        <property name="description" value="Lamp"/>
        <property name="price" value="5.75"/>
    </bean>
```

Springapp-servleet.xml (2/)

```
<bean id="messageSource"
  class="org.springframework.context.support.ResourceBundleMessageSource"
  >
    <property name="basename" value="messages"/>
 </bean>
 <bean name="/hello.htm" class="springapp.web.InventoryController">
    <property name="productManager" ref="productManager"/>
 </bean>
 <bean id="viewResolver"
  class="org.springframework.web.servlet.view.InternalResourceViewResolver"
  >
    <property name="viewClass"
  value="org.springframework.web.servlet.view.JstlView"/>
    <property name="prefix" value="/WEB-INF/jsp/"/>
    <property name="suffix" value=".jsp"/>
 </bean>
</beans>
```

Adding a form

- Expose increase functionality, add form to increase percent value from user
- Used tag library spring-form.tld (copy from spring)
- web.xml will contains

```
<jsp-config>
  <taglib>
    <taglib-uri>/spring</taglib-uri>
    <taglib-location>/WEB-INF/tld/spring-form.tld</taglib-location>
  </taglib>
</jsp-config
```

priceincrease.jsp (1/)

```
<%@ include file="/WEB-INF/jsp/include.jsp" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<html>
<head>
  <title><fmt:message key="title"/></title>
  <style>
    .error { color: red; }
  </style>
</head>
<body>
<h1><fmt:message key="priceincrease.heading"/></h1>
<form:form method="post" commandName="priceIncrease">
```

priceincrease.jsp (2/)

```
  <table width="95%" bgcolor="f8f8ff" border="0" cellspacing="0" cellpadding="5">
   <tr>
     <td align="right" width="20%">Increase (%):</td>
       <td width="20%">
         <form:input path="percentage"/>
       </td>
       <td width="60%">
         <form:errors path="percentage" cssClass="error"/>
       </td>
   </tr>
  </table>
  <br>
  <input type="submit" align="center" value="Execute">
</form:form>
<a href="<c:url value="hello.htm"/>">Home</a>
</body>
</html>
```

Adding a form controller

- define objects to inject into properties for commandClass and validator
- specify two views
  - formView that is used for the form
  - successView that we will go to after successful form processing.
- *onSubmit*(..) method gets control and does some logging before it calls the *increasePrice*(..) method on the *ProductManager* object. It then returns a *ModelAndView* passing in a new instance of a *RedirectView* created using the URL for the success view

Thank you