Spring Form Contollers

EHSDI
eBuzima

SimpleFormController

- A Form Controller with features for:
  - Viewing
  - Validating
  - Submitting

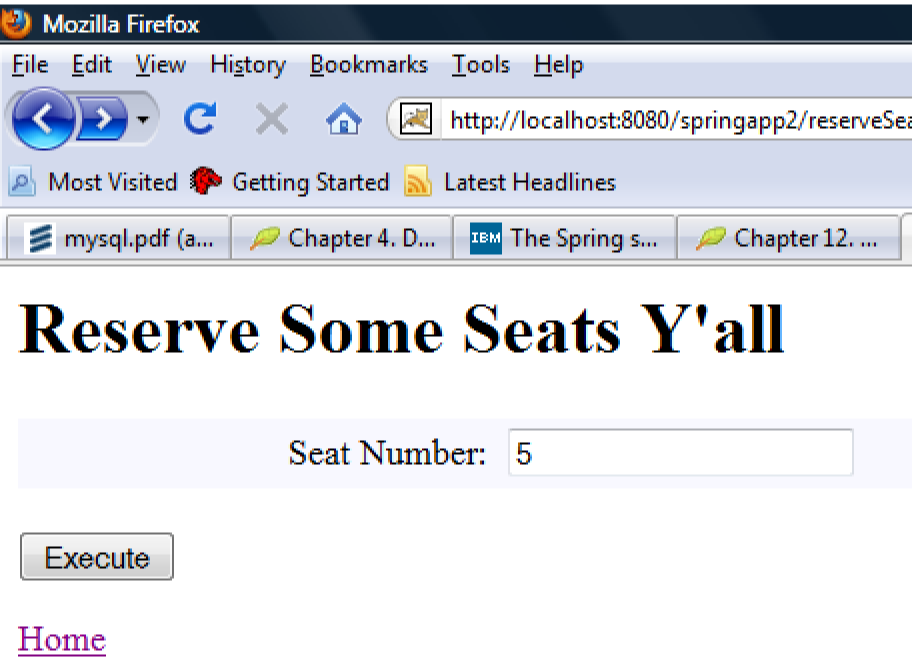
SimpleFormController

- Properties:
  - formView: the logical name for the page containing the form

  - successView: the logical name for the page to show after a successful form submission

FormView and SuccessView

- FormView
- SuccessView

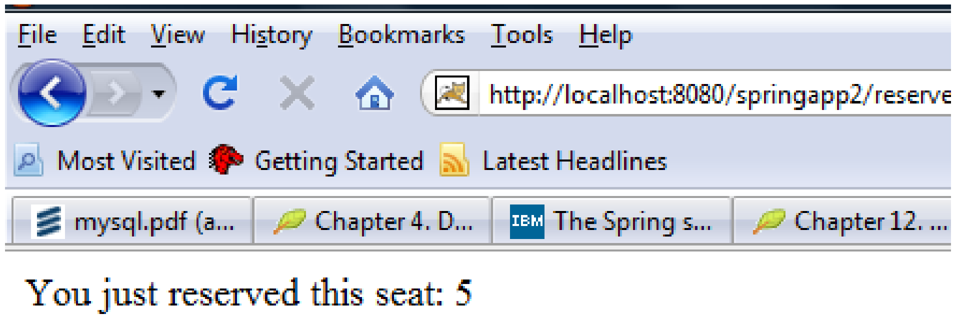

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/springapp2/reserveSea
Most Visited Getting Started Latest Headlines
mysql.pdf (a... Chapter 4. D... The Spring s... Chapter 12. ...

Reserve Some Seats Y'all

Seat Number: 5

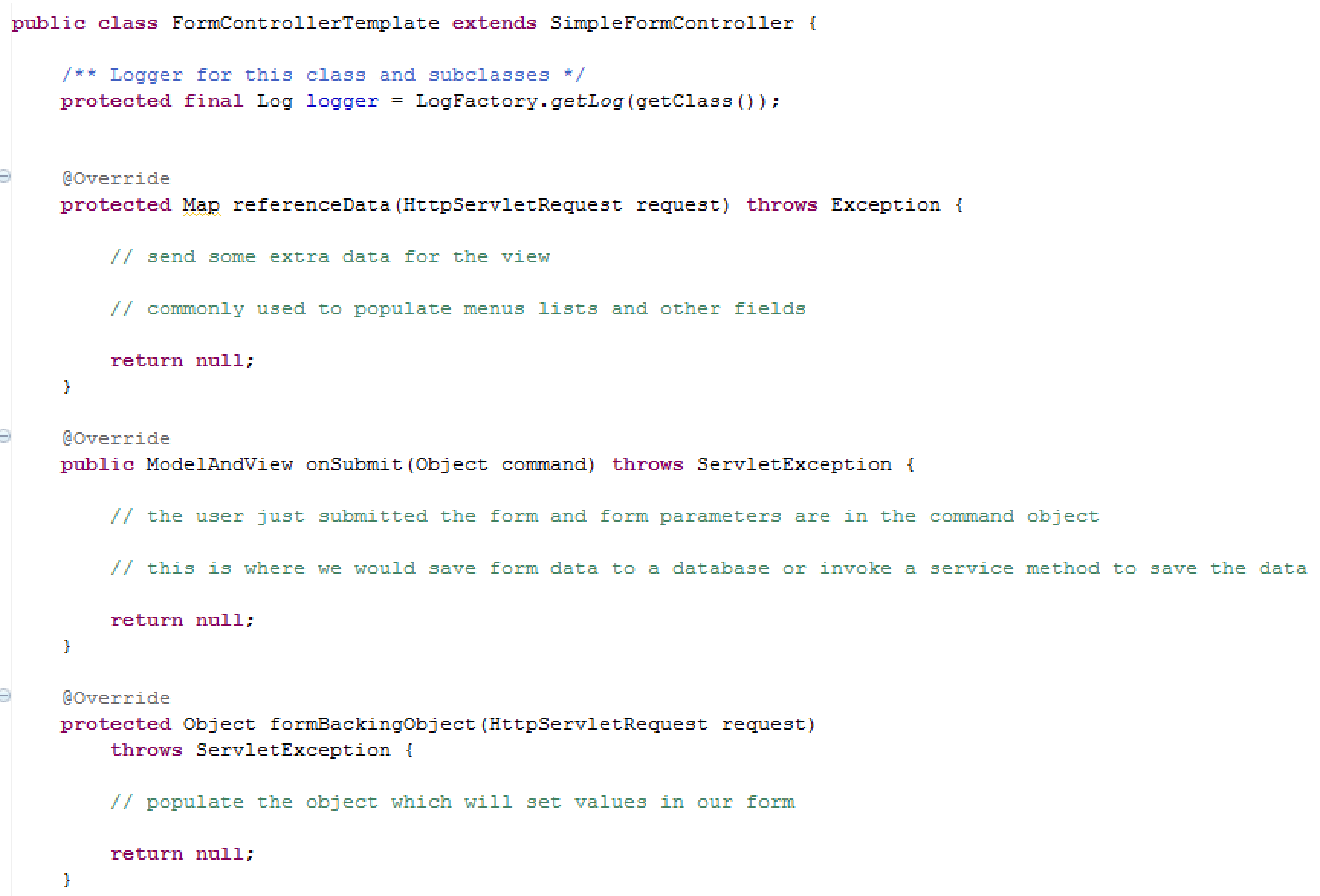
Execute

Home


File Edit View History Bookmarks Tools Help
http://localhost:8080/springapp2/reserve
Most Visited Getting Started Latest Headlines
mysql.pdf (a... Chapter 4. D... The Spring s... Chapter 12. ...

You just reserved this seat: 5

A First Look At SimpleFormController



```
public class FormControllerTemplate extends SimpleFormController {

    /** Logger for this class and subclasses */
    protected final Log logger = LogFactory.getLog(getClass());


    @Override
    protected Map referenceData(HttpServletRequest request) throws Exception {

        // send some extra data for the view

        // commonly used to populate menus lists and other fields

        return null;
    }

    @Override
    public ModelAndView onSubmit(Object command) throws ServletException {

        // the user just submitted the form and form parameters are in the command object

        // this is where we would save form data to a database or invoke a service method to save the data

        return null;
    }

    @Override
    protected Object formBackingObject(HttpServletRequest request)
        throws ServletException {

        // populate the object which will set values in our form

        return null;
    }
```

A Look at Three Methods

- These methods will commonly be implemented for basic FormController developent:

1. `formBackingObject()`
2. `referenceData()`
3. `onSubmit()`

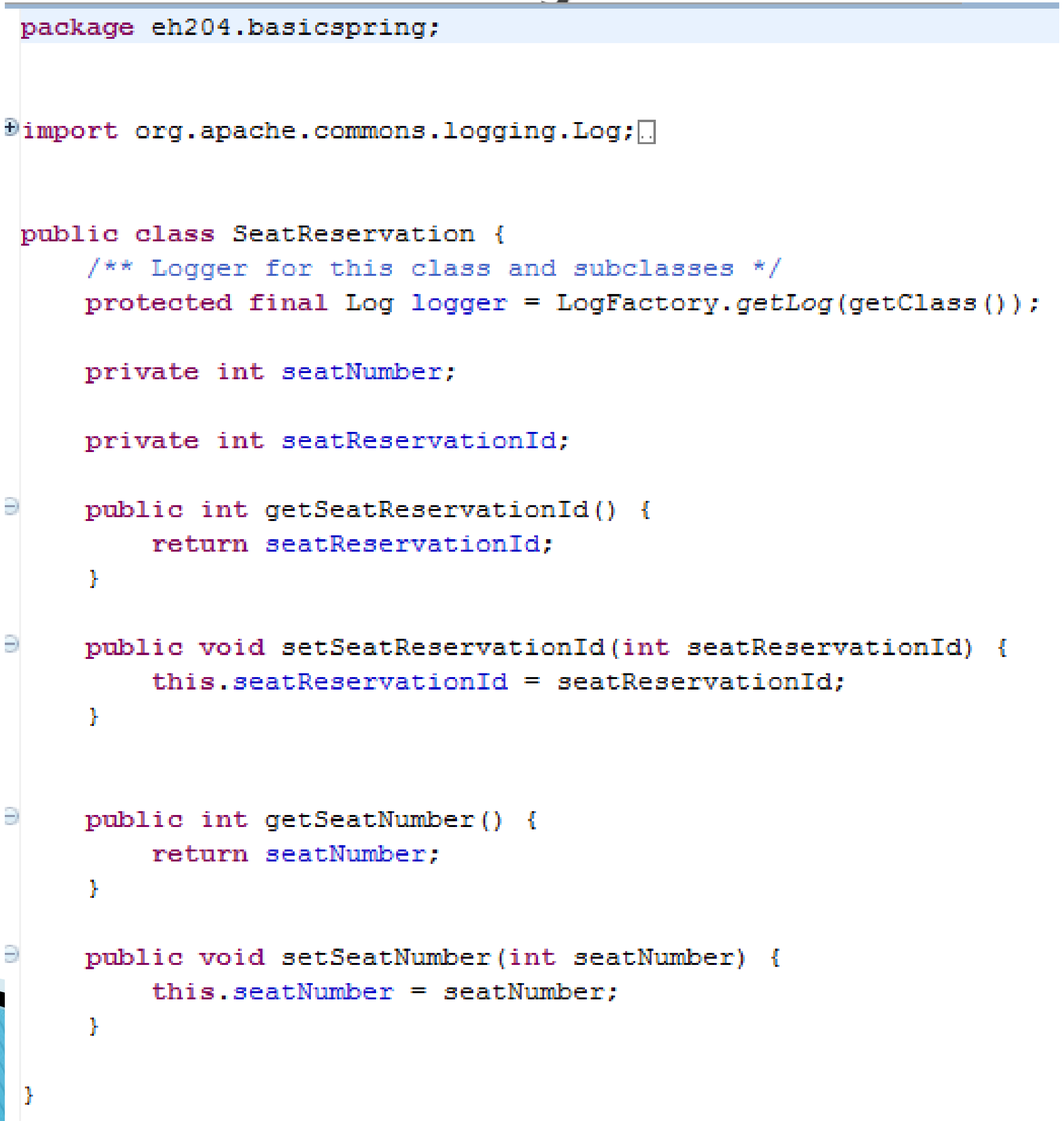
EHSDI
eBuzima

formBackingObject()

- Populate an object which you can reference on a jsp!

```
protected Object formBackingObject(HttpServletRequest request) throws ServletException {
    SeatReservation seatReservation = new SeatReservation();
    seatReservation.setSeatNumber(5);

    return seatReservation;
}
```

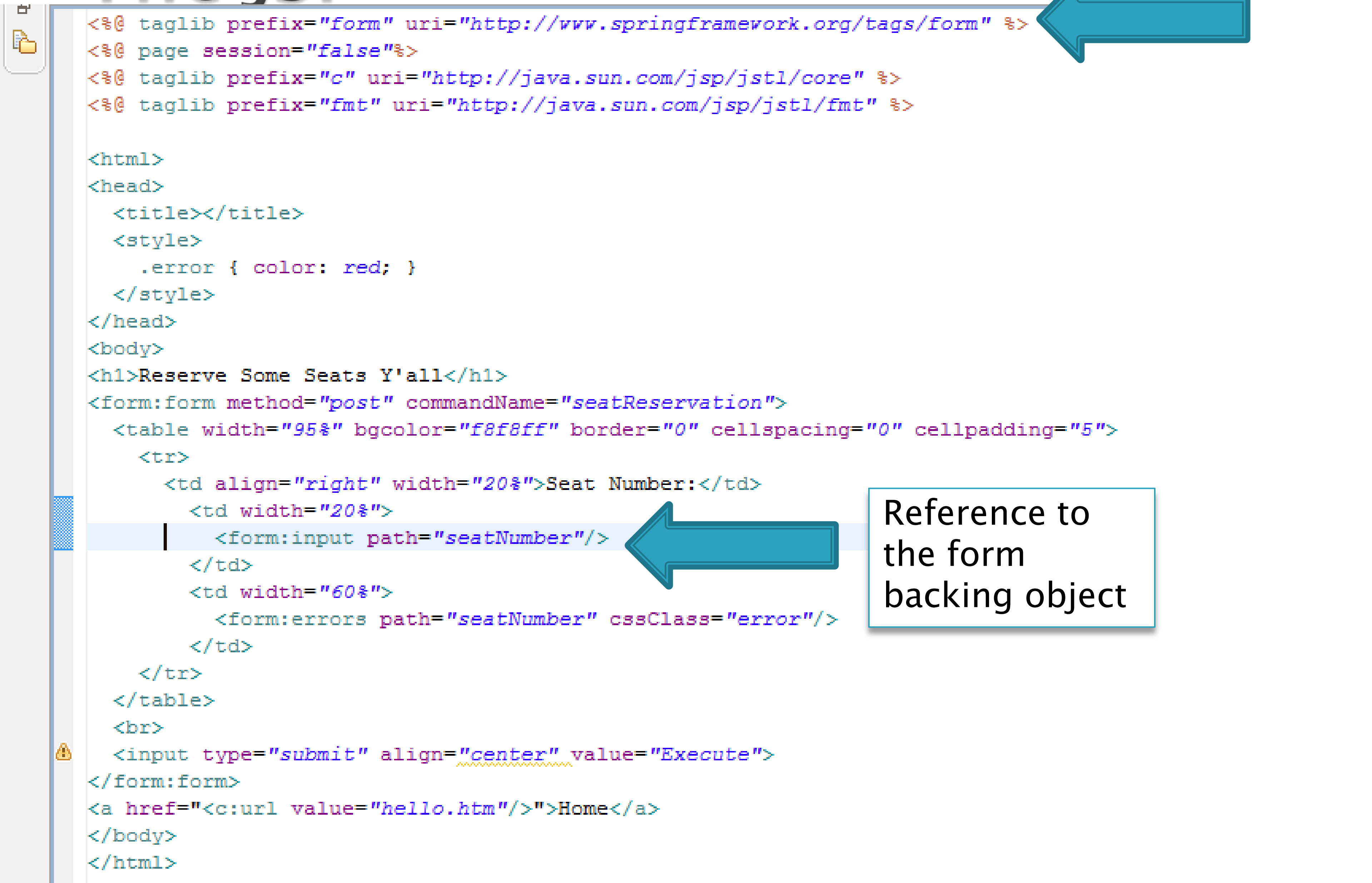
The Command Object: SeatReservation.java



```
package eh204.basicspring;


import org.apache.commons.logging.Log;


public class SeatReservation {
    /** Logger for this class and subclasses */
    protected final Log logger = LogFactory.getLog(getClass());

    private int seatNumber;

    private int seatReservationId;

    public int getSeatReservationId() {
        return seatReservationId;
    }

    public void setSeatReservationId(int seatReservationId) {
        this.seatReservationId = seatReservationId;
    }


    public int getSeatNumber() {
        return seatNumber;
    }

    public void setSeatNumber(int seatNumber) {
        this.seatNumber = seatNumber;
    }

}
```

EHSDI
eBuzima

The JSP



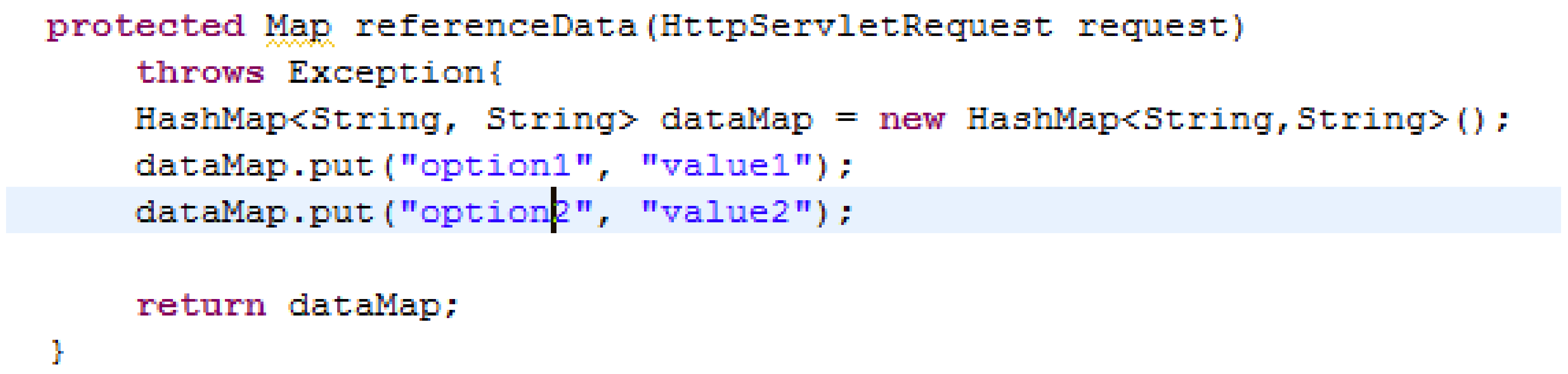
```
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ page session="false"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>

<html>
<head>
  <title></title>
  <style>
    .error { color: red; }
  </style>
</head>
<body>
<h1>Reserve Some Seats Y'all</h1>
<form:form method="post" commandName="seatReservation">
  <table width="95%" bgcolor="f8f8ff" border="0" cellspacing="0" cellpadding="5">
    <tr>
      <td align="right" width="20%">Seat Number:</td>
        <td width="20%">
          <form:input path="seatNumber"/>
        </td>
        <td width="60%">
          <form:errors path="seatNumber" 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>
```

Spring form tag library

Reference to the form backing object

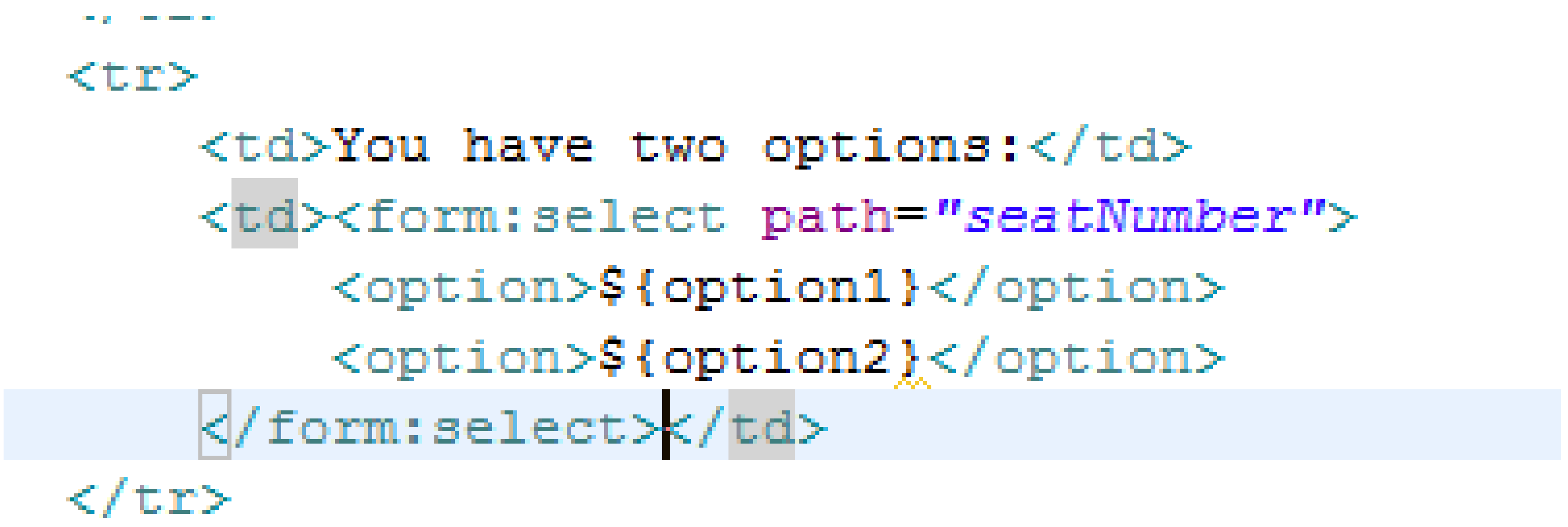
referenceData()

- Add some extra data that might be used in a form.



```java
protected Map referenceData(HttpServletRequest request)
    throws Exception{
    HashMap<String, String> dataMap = new HashMap<String,String>();
    dataMap.put("option1", "value1");
    dataMap.put("option2", "value2");

    return dataMap;
}
```

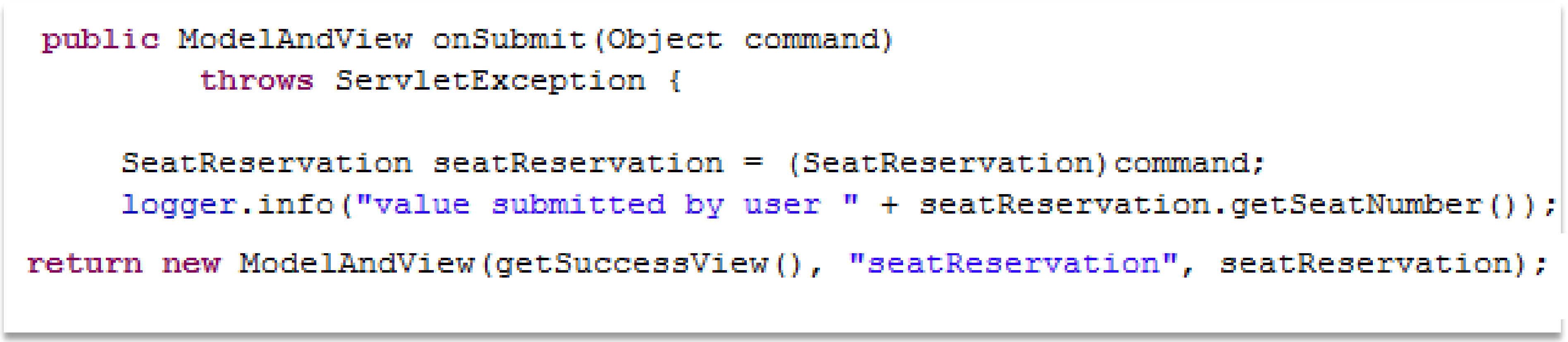
The Reference Data in the JSP



```
<tr>
    <td>You have two options:</td>
    <td><form:select path="seatNumber">
        <option>${option1}</option>
        <option>${option2}</option>
    </form:select></td>
</tr>
```

onSubmit()

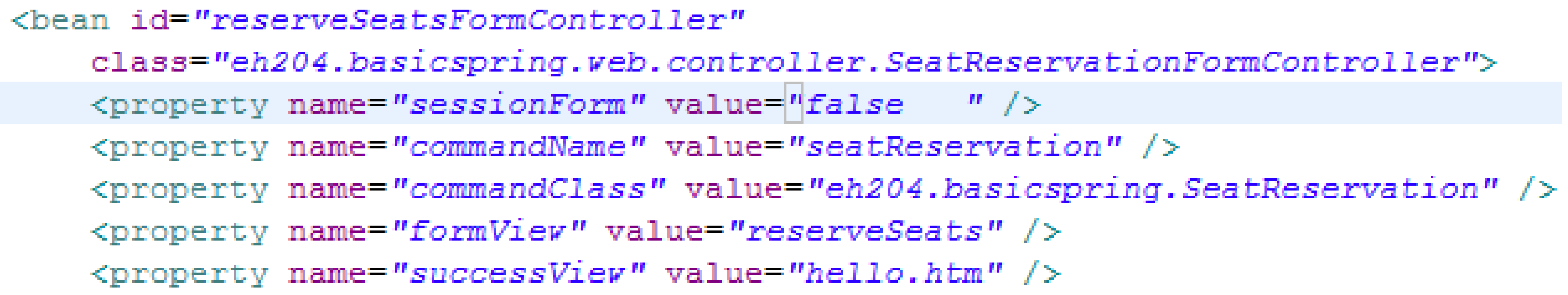
- Process the data submitted by the user



```
 public ModelAndView onSubmit(Object command)
         throws ServletException {

     SeatReservation seatReservation = (SeatReservation)command;
     logger.info("value submitted by user " + seatReservation.getSeatNumber());
return new ModelAndView(getSuccessView(), "seatReservation", seatReservation);
```

Controller Configuration

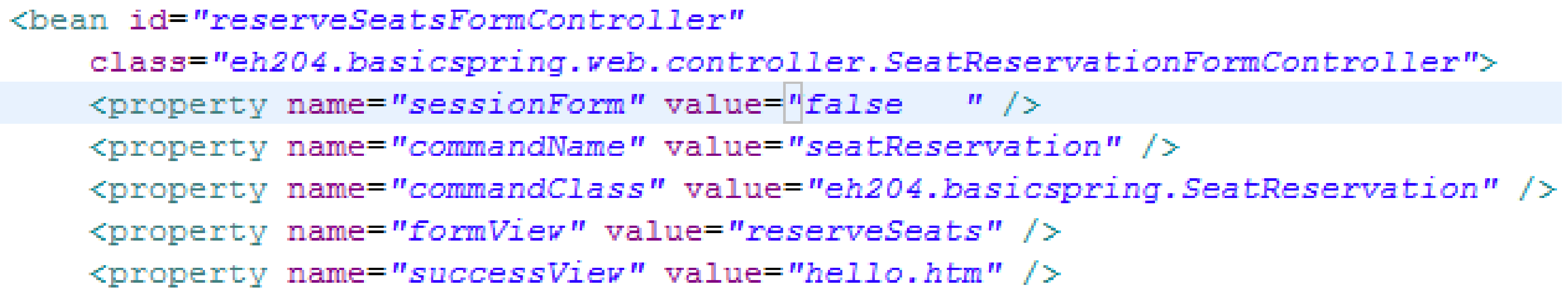
- sessionForm determines whether the container uses the same instance of the command object or not. A setting to false forces the instantiation of a new instance per request



```
<bean id="reserveSeatsFormController"
    class="eh204.basicspring.web.controller.SeatReservationFormController">
    <property name="sessionForm" value="false    " />
    <property name="commandName" value="seatReservation" />
    <property name="commandClass" value="eh204.basicspring.SeatReservation" />
    <property name="formView" value="reserveSeats" />
    <property name="successView" value="hello.htm" />
```

ControllerConfiguration

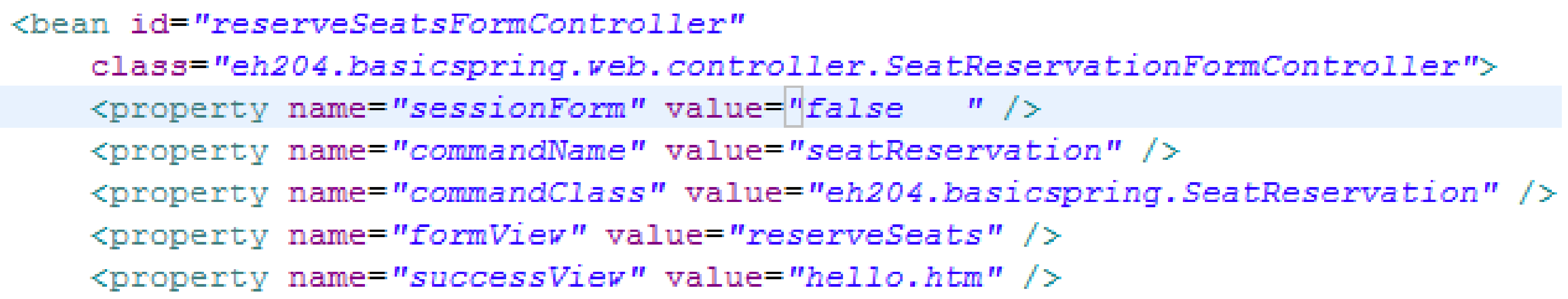
- commandName is the name of the command object parameter used in the FormController



```
<bean id="reserveSeatsFormController"
    class="eh204.basicspring.web.controller.SeatReservationFormController">
    <property name="sessionForm" value="false    " />
    <property name="commandName" value="seatReservation" />
    <property name="commandClass" value="eh204.basicspring.SeatReservation" />
    <property name="formView" value="reserveSeats" />
    <property name="successView" value="hello.htm" />
```

ControllerConfiguration

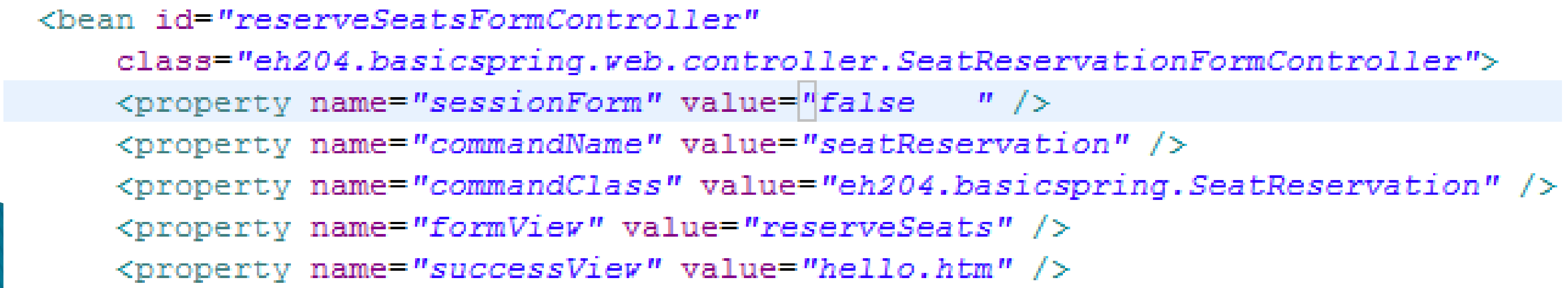
- commandClass is the complete class name for the command object



```xml
<bean id="reserveSeatsFormController"
    class="eh204.basicspring.web.controller.SeatReservationFormController">
    <property name="sessionForm" value="false    " />
    <property name="commandName" value="seatReservation" />
    <property name="commandClass" value="eh204.basicspring.SeatReservation" />
    <property name="formView" value="reserveSeats" />
    <property name="successView" value="hello.htm" />
```

Controller Configuration

- formView is the logical name for the form view
- successView is the logical name for the view to show after a successful submission

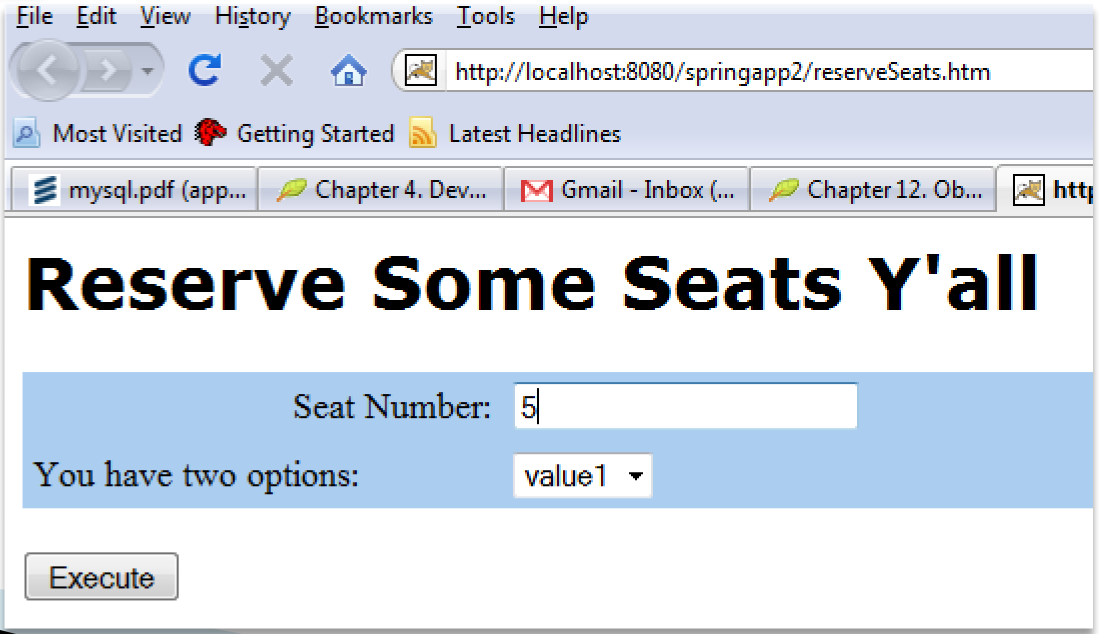


```
<bean id="reserveSeatsFormController"
    class="eh204.basicspring.web.controller.SeatReservationFormController">
    <property name="sessionForm" value="false    " />
    <property name="commandName" value="seatReservation" />
    <property name="commandClass" value="eh204.basicspring.SeatReservation" />
    <property name="formView" value="reserveSeats" />
    <property name="successView" value="hello.htm" />
```

```
protected Object formBackingObject(HttpServletRequest request)
                throws ServletException {
        SeatReservation seatReservation = new SeatReservation();
        seatReservation.setSeatNumber(5);

        return seatReservation;
    }
```

Populates the commandObject

5

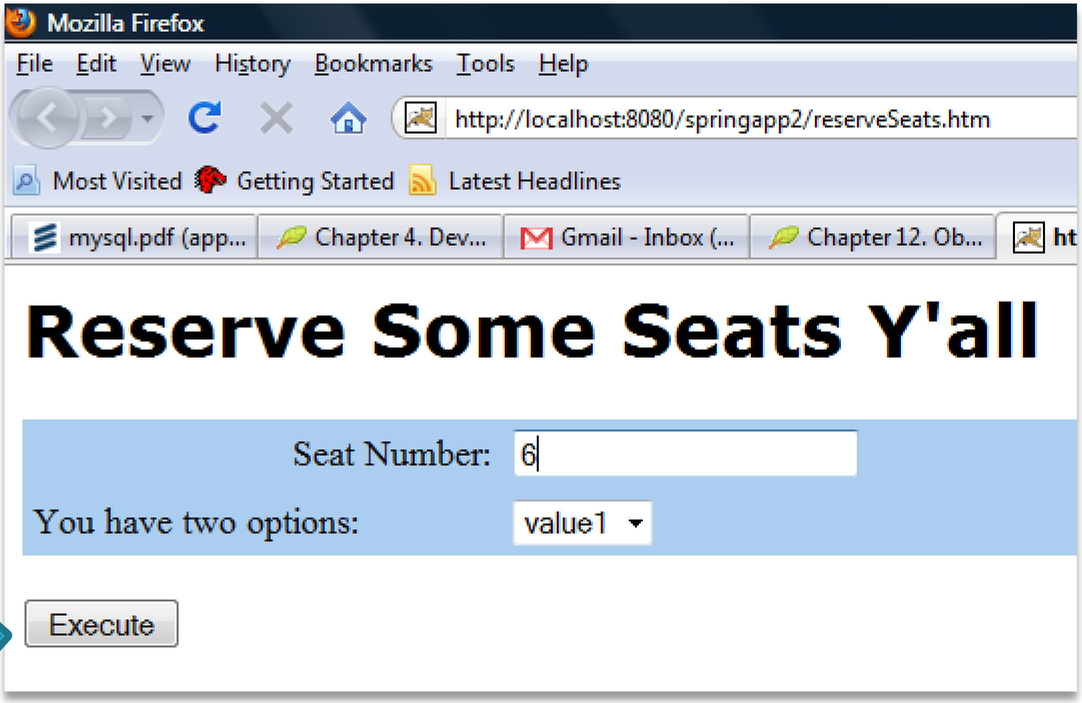


Reserve Some Seats Y'all

Seat Number: 5

You have two options: value1

Execute


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/springapp2/reserveSeats.htm
Most Visited Getting Started Latest Headlines
mysql.pdf (app... Chapter 4. Dev... Gmail - Inbox (... Chapter 12. Ob... ht

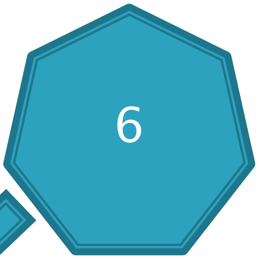
Reserve Some Seats Y'all

Seat Number: 6

You have two options: value1

Execute

User submits the form



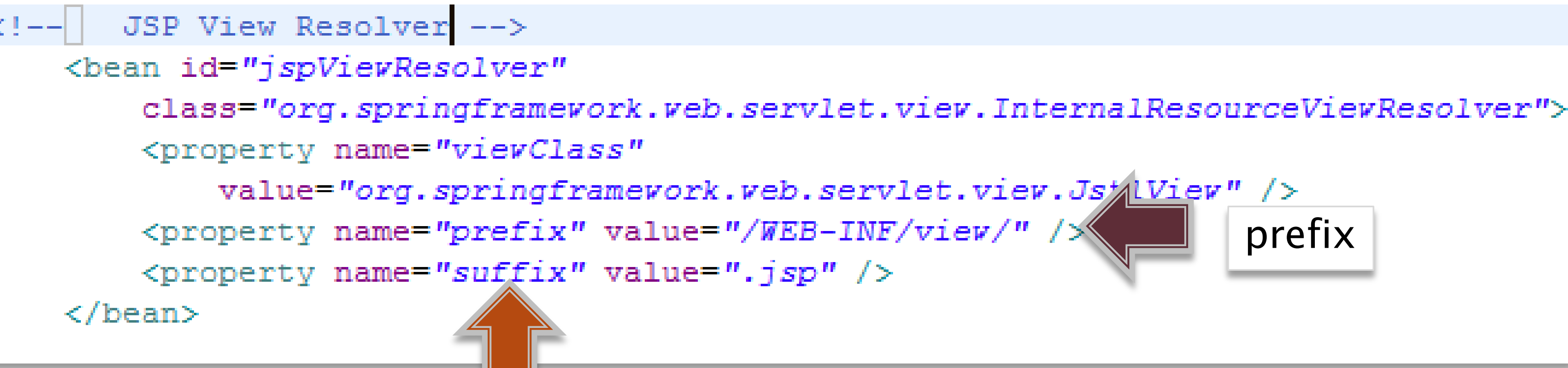
Populates the commandObject

```
public ModelAndView onSubmit(Object Command){
        SeatReservation seatReservation = (SeatReservation)command;
        logger.info("value submitted by user " +
                seatReservation.getSeatNumber());
}
```

Quick URLMapping Review

```
<bean id="urlMapping"
    class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
        <props>
            <prop key="/hello.htm">helloCtrl</prop>
            <prop key="/reserveSeats.htm">reserveSeatsFormController
            </prop>
        </props>
    </property>
</bean>


<bean id="helloCtrl" class="eh204.basicspring.web.controller.HelloController" />

<bean id="reserveSeatsFormController"
    class="eh204.basicspring.web.controller.SeatReservationFormController">
    <property name="sessionForm" value="false    " />
    <property name="commandName" value="seatReservation" />
    <property name="commandClass" value="eh204.basicspring.SeatReservation" />
    <property name="formView" value="reserveSeats" />
    <property name="successView" value="hello.htm" />
```

Quick View Resolver Review



```
!-- JSP View Resolver -->
    <bean id="jspViewResolver"
        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="viewClass"
            value="org.springframework.web.servlet.view.JstlView" />
        <property name="prefix" value="/WEB-INF/view/" />
        <property name="suffix" value=".jsp" />
    </bean>
```

prefix

suffix

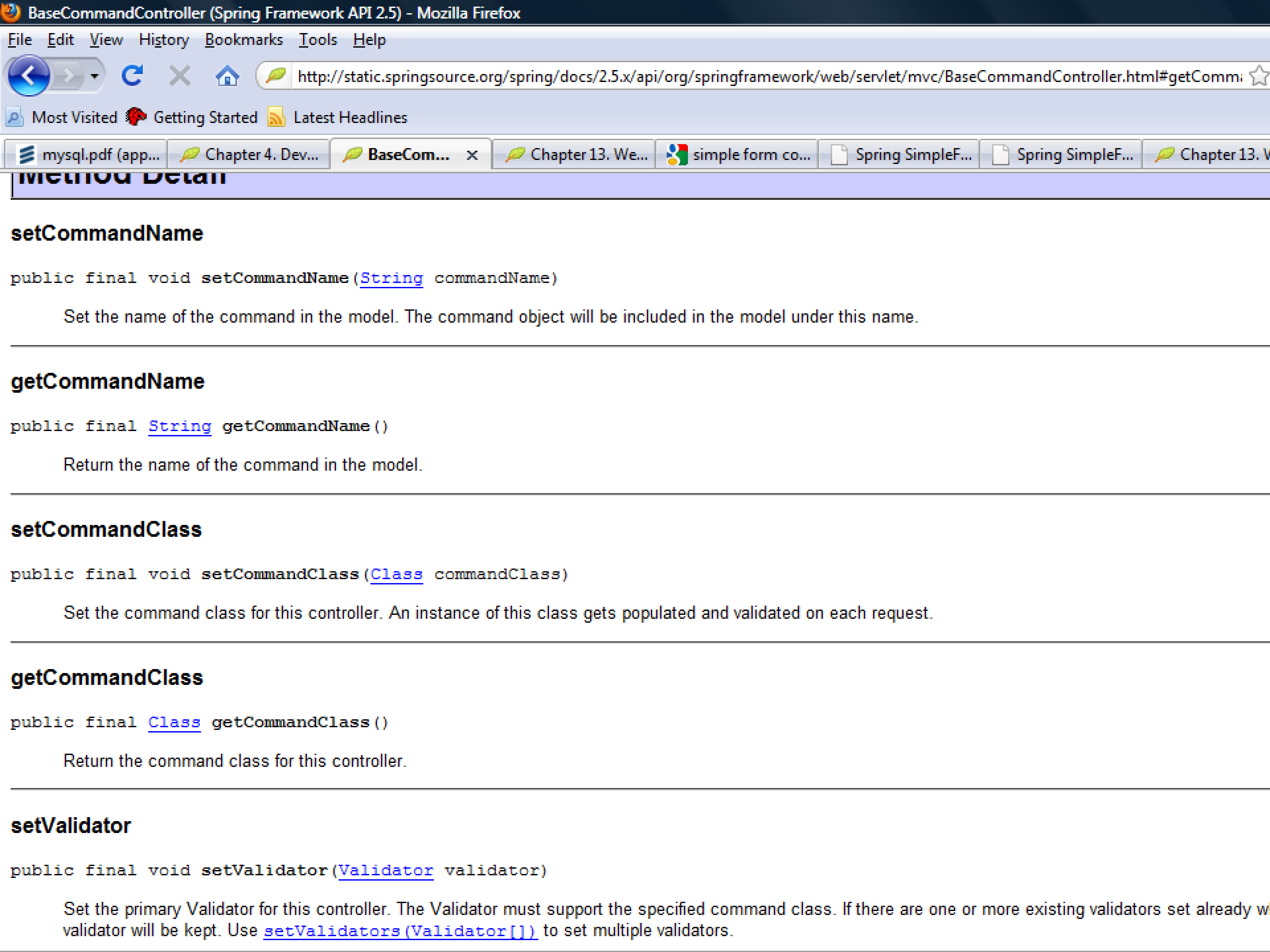
If the logical view name is : `reservationForm`

ViewResolver translates it to:

/WEB-INF/view/reservationForm.jsp

prefix

suffix


BaseCommandController (Spring Framework API 2.5) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://static.springsource.org/spring/docs/2.5.x/api/org/springframework/web/servlet/mvc/BaseCommandController.html#getComm
Most Visited Getting Started Latest Headlines
mysql.pdf (app...
Chapter 4. Dev...
BaseCom...
Chapter 13. We...
simple form co...
Spring SimpleF...
Spring SimpleF...
Chapter 13. W

Method Detail

setCommandName

```
public final void setCommandName(String commandName)
```

Set the name of the command in the model. The command object will be included in the model under this name.

getCommandName

```
public final String getCommandName()
```

Return the name of the command in the model.

setCommandClass

```
public final void setCommandClass(Class commandClass)
```

Set the command class for this controller. An instance of this class gets populated and validated on each request.

getCommandClass

```
public final Class getCommandClass()
```

Return the command class for this controller.

setValidator

```
public final void setValidator(Validator validator)
```

Set the primary Validator for this controller. The Validator must support the specified command class. If there are one or more existing validators set already w
validator will be kept. Use `setValidators(Validator[])` to set multiple validators.