

## 1.BookController.java

```
package com.controller;

import java.util.ArrayList;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.exception.InvalidBookException;
import com.model.Book;
import com.service.BookService;

//use appropriate annotation to configure BookController as Controller
@Controller
public class BookController {

    // use appropriate annotation
    @Autowired
    private BookService bookService;
```

```
@RequestMapping(value = "/index", method = RequestMethod.GET)

public String showIndex() {

    return "index";

}
```

```
@RequestMapping(value = "/showPage", method = RequestMethod.GET)

public String showPage(@ModelAttribute("book") Book book) {

    return "showPage";

}
```

```
// invoke the service class - addBook method.
```

```
@RequestMapping(value = "/addBook", method = RequestMethod.GET)

public String addBook(@ModelAttribute("book") Book book, BindingResult result, ModelMap
model) throws InvalidBookException {

    String res = "";

    if (result.hasErrors()) {

        res = "showpage";

    } else {

        boolean bool = bookService.addBook(book);

        model.addAttribute("message", "Book added successfully");

        res = "showpage";

    }

    return res;

}

// fill the code
```

```
@RequestMapping(value = "/viewList", method = RequestMethod.GET)
public String viewAllBooks(ModelMap model) {
```

```
    // fill the code

    List<Book> list = bookService.viewAllBooks();
    model.addAttribute("bookList",list);
    return "viewList";
```

```
}
```

```
// Use appropriate annotation
```

```
@ExceptionHandler(value = InvalidBookException.class)
public ModelAndView exceptionHandler(Exception e) {
    ModelAndView modelAndView = new ModelAndView("exceptionPage");
    modelAndView.addObject("err",e.getMessage());
    // fill the code
    return modelAndView;
}
```

```
@ModelAttribute("bookCategory")
```

```
public List<String> populateBookCategory() {
    List<String> list = new ArrayList<>();
    list.add("Science");
    list.add("Humor");
    list.add("Novel");
    list.add("Health");
    list.add("Travel");
    list.add("Motivational");
    list.add("Art");
}
```

```

        list.add("Encyclopedia");
        list.add("Biography");
        list.add("Cooking");
        return list;
    }
}

```

## 2. InternationalizationConfig.java

```

package com.controller;

import java.util.Locale;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.support.ResourceBundleMessageSource;
import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;
import org.springframework.web.servlet.LocaleResolver;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
import org.springframework.web.servlet.i18n.SessionLocaleResolver;

@Configuration
public class InternationalizationConfig implements WebMvcConfigurer {

```

```

// Create a SessionLocaleResolver object and set the default locale to
// English return the SessionLocaleResolver object

@Bean
public LocaleResolver localeResolver() {

    SessionLocaleResolver resolver = new SessionLocaleResolver();
    resolver.setDefaultLocale(new Locale("en"));
    return resolver;
}

// Create LocaleChangeInterceptor object and set the parameter name as
// language and return the localeChangeInterceptor
@Bean
public LocaleChangeInterceptor localeChangeInterceptor() {
    LocaleChangeInterceptor interceptor = new LocaleChangeInterceptor();
    interceptor.setParamName("language");
    return interceptor;
}

// register the LocaleChangeInterceptor
@Override
public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(localeChangeInterceptor());
}

@Bean
public ResourceBundleMessageSource messageSource() {

```

```

        ResourceBundleMessageSource bundleMessageSource = new
ResourceBundleMessageSource();

bundleMessageSource.setBasenames("messages","messages_fr","messages_de");

        return bundleMessageSource;
    }

    @Bean
    public LocalValidatorFactoryBean getValidator() {
        LocalValidatorFactoryBean bean = new LocalValidatorFactoryBean();
        bean.setValidationMessageSource(messageSource());
        return bean;
    }
}

```

### 3. ClassicBookHouseApplication.java

```

package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan({"com.controller","com.model","com.service"})
public class ClassicBookHouseApplication {

    public static void main(String[] args) {

```

```
        SpringApplication.run(ClassicBookHouseApplication.class, args);
    }

}
```

#### **4. InvalidBookException.java**

```
package com.exception;

public class InvalidBookException extends Exception{
    private static final long serialVersionUID = 1L;
    public InvalidBookException(String msg) {

        super(msg);
        //fill the code
    }

}
```

#### **5. Book.java**

```
package com.model;

import javax.validation.constraints.NotBlank;
import javax.validation.constraints.NotEmpty;
import javax.validation.constraints.PositiveOrZero;
```

```
import org.springframework.stereotype.Component;

//use appropriate annotation to make this class as a component class
@Component
public class Book {

    //fill the code
    @NotEmpty(message = "{error.isbn}")
    private String isbn;

    //fill the code
    @NotBlank(message = "{error.bookName}")
    private String bookName;

    //fill the code
    @PositiveOrZero(message = "{error.quantity}")
    private int quantity;

    private String category;

    //fill the code
    @PositiveOrZero(message = "{error.price}")
    private int price;


    public String getCategory() {

        return category;
    }

    public void setCategory(String category) {

        this.category = category;
    }

    public String getBookName() {

        return bookName;
    }
}
```



```
}  
  
public void setBookName(String bookName) {  
    this.bookName = bookName;  
}
```

```
  
public int getPrice() {  
    return price;  
}
```

```
  
public void setPrice(int price) {  
    this.price = price;  
}
```

```
  
public int getQuantity() {  
    return quantity;  
}
```

```
  
public void setQuantity(int quantity) {  
    this.quantity = quantity;  
}
```

```
  
public String getIsbn() {  
    return isbn;  
}
```

```
  
public void setIsbn(String isbn) {  
    this.isbn = isbn;  
}
```

```
  
public Book() {}  
}
```

## 7. BookService.java

```
package com.service;

import java.util.ArrayList;

import org.springframework.stereotype.Service;

import com.exception.InvalidBookException;
import com.model.Book;

//use appropriate annotation to configure ElectricityService as a Service
@Service
public class BookService {

    private static ArrayList<Book> bookList=new ArrayList<Book>();

    //    Validate ISBN number and add book object into the book list
    public boolean addBook(Book book) throws InvalidBookException {
        for(Book checkBook:bookList) {
            if(checkBook.getIsbn().equalsIgnoreCase(book.getIsbn())) {
                throw new InvalidBookException("Book with "+book.getIsbn()+" already
exists!");
            }
        }
        bookList.add(book);
        return true;
    }
}
```

```

        public ArrayList<Book> viewAllBooks(){
            //fill the code
            return this.bookList;
        }

        public ArrayList<Book> getBookList() {
            return this.bookList;
        }

        public void setBookList(ArrayList<Book> list) {
            this.bookList = list;
        }
    }
}

```

## 8.exceptionPage.jsp

```

<%@page isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<html>
<body bgcolor="lavender">
<h1><center>Classic Book House</center></h1>

        <div>

        <center>

        <h1> ${err} </h1>

```

```
<a href="index"> Home</a>
```

```
</center>
```

```
</div>
```

```
</body>
```

```
</html>
```

## 9.index.jsp

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
```

```
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form" %>
```

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<body style="background-color:lavender">
```

```
<h1><center>Classic Book House</center></h1>
```

```
<center>
```

```
<a href="showPage">Add Book</a><br><br>
```

```
<a href="viewList" >View all Book</a>
```

```
</center>
```

```
</body>
```

```
</html>
```

## 10.showPage.jsp

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```
<head>
```

```
<style>
```

```
* {
    box-sizing: border-box;
}
```

```
.row {
    margin-left: -5px;
    margin-right: -5px;
}
```

```
.column {
    float: left;
    width: 60%;
    padding: 3px;
}
```

```
.column1 {
    float: left;
    width: 25%;
    padding: 3px;
}

/* Clearfix (clear floats) */
.row::after {
    content: "";
    clear: both;
    display: table;
}

table {
    border-collapse: collapse;
    border-spacing: 0;
    width: 100%;
}

th, td {
    text-align: left;
    padding: 5px;
}

#message {
    color: red;
}

</style>
```

</head>

<body style="background-color: lavender">

<!-- Do not change the below line -->

<a href="/showPage?language=en">English</a>|

<a href="/showPage?language=de">German</a>|

<a href="/showPage?language=fr">French</a>

</align>

<!-- Design the page as per the requirements-->

<h1>

<center>Classic Book House</center>

</h1>

<form:form method="GET" action="/addBook" modelAttribute="book">

<div class="row">

<div class="column1">

<table>

<tr>

<td><a href="index">Home</a><br></td>

<tr>

<td><a href="viewList">View All Books</a></td>

</tr>

</table>

</div>

<div class="column">

<table

style="margin: 0px auto; margin-left: auto; margin-right: auto">

```

        <tr>
            <td><spring:message code="label.isbn" /></td>
            <td><form:input path="isbn" id="isbn" /></td>
            <td><form:errors path="isbn" /></td>
        </tr>

        <tr>
            <td><spring:message code="label.bookName" /></td>
            <td><form:input path="bookName" id="bookName"
/></td>
            <td><form:errors path="bookName" /></td>
        </tr>

        <tr>
            <td><spring:message code="label.quantity" /></td>
            <td><form:input path="quantity" id="quantity" /></td>
            <td><form:errors path="quantity" /></td>
        </tr>

        <tr>
            <td><spring:message code="label.category" /></td>
            <td><form:select path="category" id="category"
items="\${bookCategory}"></form:select></td>
        </tr>

        <tr>
            <td><spring:message code="label.price" /></td>
            <td><form:input path="price" id="price" /></td>

```



```

        <td><form:errors path="price" /></td>
    </tr>

    <tr>
        <td><input type="submit" value="Submit"
id="submit"/></td>
    </tr>

</table>
</div>
</div>

<div id="message">${message}</div>
</center>
</form:form>

</body>
</html>

```

## 11.viewList.jsp

```

<%@page isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<html>

```

```
<body bgcolor="lavender">
```

```
<h1><center>Classic Book House</center></h1>
```

```
<center>
```

```
<table>
```

```
<tr>
```

```
<th>ISBN</th>
```

```
<th>Book Name</th>
```

```
<th>Book Category</th>
```

```
<th>Quantity</th>
```

```
<th>Price</th>
```

```
</tr>
```

```
<c:forEach var="book" items="${bookList }">
```

```
<tr>
```

```
<td>${book.getIsbn() }</td>
```

```
<td>${book.getBookName() }</td>
```

```
<td>${book.getCategory() }</td>
```

```
<td>${book.getQuantity() }</td>
```

```
<td>${book.getPrice() }</td>
```

```
</tr>
```

```
</c:forEach>
```

```
</table>
```

```
</center>
```

```
<a href="index">Back to Home</a>
```

</body>

</html>