



**CourseCube®**  
(Formerly Java Learning Center)

# **Spring 5.2 +Hibernate5.4**

**Mini Project**

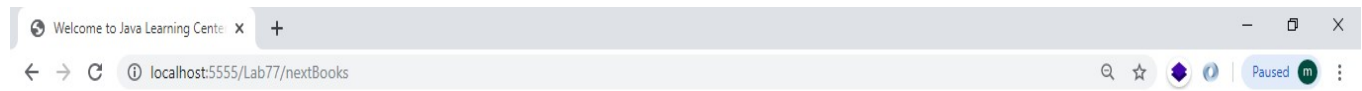
**Author  
Srinivas Dande**



## Lab 82: Mini Project with Spring and Hibernate.

### Lab82: Files required

1. index.jsp	2. booksList.jsp
3. viewBook.jsp	4. addEditBook.jsp
5. IndexController.java	6. BookController.java
7. BookService.java	8. BookServiceImpl.java
9. MailService.java	10. MailServiceImpl.java
11. BookDAO.java	12. BookDAOImpl.java
13. Book.java	14. JLCWebConfig.java
15. JLCWebAppInitializer.java	



### Books at Java Learning Center

11 to 15 of 21

Book ID	Book Name	Author	Price	Category	Publications	Add New Book	
<a href="#">116</a>	Learn Spring5	Srinivas	500	Java	JLC	Edit	Delete
<a href="#">117</a>	Learn Hibernate5	Srinivas	400	Java	JLC	Edit	Delete
<a href="#">118</a>	Master Angular	Vas	300	Web	JLC	Edit	Delete
<a href="#">120</a>	Master Pink	Sri	100	Java	SD	Edit	Delete
<a href="#">122</a>	Master Python 3	Srini	333	Python	SD	Edit	Delete

[Previous](#) [Next](#)

### 1. index.jsp

```
<html>
<body>
<br><h1> Java Learning Center </h1>
<h2> <a href="showAllBooks">Display All Books</a> </h2>
</body> </html>
```



## 2.booksList.jsp

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

<html>
<head>
<title>Welcome to CC!</title>
<style>
table,td,th{
border:0px rgb(180, 151, 151) solid;
border-collapse: collapse;
font-size: 25px;
}
.mytable{
width: 90%;
}
h2{
width: 90%;
background-color: goldenrod;
height:30px;
}
a{
color:blue;
}
tr:nth-child(odd){
background-color: azure;
}
tr:nth-child(even){
background-color:lavender;;
}

input[type=submit]{
color: blue;
font-family: verdana;
font-size: 20px;
font-weight:bold;
padding: 5px 5px;
margin: 5px 5px;
}
</style>
</head>
<body>
<div align="center">
<h1>Books at Java Learning Center</h1>
```



```
<h2 align="right"> ${FROM} to ${TO} of ${TotalBooks} </h2>
<table class="mytable">
<tr>
<th>Book ID</th>
<th>Book Name</th>
<th>Author</th>
<th>Price</th>
<th>Category</th>
<th>Publications </th>
<th colspan="2" align="center">
<form:form action="addEditBookForm">
<input type="hidden" name="bookId" value="0"/>
<input type="submit" value="Add New Book" />
</form:form> </th>
</tr>
<c:forEach var="mybook" items="${MyBooksList}">
<tr>
<td> <a href="viewBook?bookId=${mybook.bid }"> ${mybook.bid } </a> </td>
<td>${mybook.bname }</td>
<td>${mybook.author }</td>
<td>${mybook.price }</td>
<td>${mybook.category }</td>
<td>${mybook.pub }</td>

<td>
<form:form action="addEditBookForm">
<input type="hidden" name="bookId" value="${mybook.bid }"/>
<input type="submit" value=" Edit " />
</form:form>
</td>

<td>
<form:form action="deleteBook">
<input type="hidden" name="bookId" value="${mybook.bid }"/>
<input type="submit" value="Delete" />
</form:form>
</td>
</tr>
</c:forEach>
</table>
<h2 >
```

```
<c:if test="\${ShowPrevious eq 'TRUE'}">
<a href="previousBooks">Previous </a> </c:if>
    ;
<c:if test="\${ShowNext eq 'TRUE'}">
<a href="nextBooks">Next </a> </c:if>
</h2>
</div>
</body>
</html>
```

### 3. viewBook.jsp

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

```
<html>
<head>
<title>Welcome to CC!</title>
<style>
table,td,th{
border:0px rgb(180, 151, 151) solid;
border-collapse: collapse;
font-size: 25px;
width: 50%;
}

.mytable{
width: 90%;
}

tr:nth-child(odd){
background-color: azure;
}
tr:nth-child(even){
background-color:lavender;;
}
</style>
</head>
<body>
<div align="center">
<table>
<tr>
<td align="center" colspan="3"> Book Details</td>
</tr>
<tr>
```



```
<td> Book Id</td>
<td> ${MyBook.bid} </td>
</tr>
<tr>
<td> Book Name</td>
<td> ${MyBook.bname} </td>
</tr>
<tr>
<td> Author</td>
<td> ${MyBook.author} </td>
</tr>
<tr>
<td> Price</td>
<td> ${MyBook.price} </td>
</tr>
<tr>
<td> Category</td>
<td> ${MyBook.category} </td>
</tr>
<tr>
<td> Publications</td>
<td> ${MyBook.pub} </td>
</tr>
</table>
<h2> <a href="showAllBooks">Goto Book Home</a> </h2>
</div>
</body>
</html>
```

#### **4.addEditBook.jsp**

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

<html> <head>
<title>Welcome to CC!</title>
<style>
table,td,th{
border:0px rgb(180, 151, 151) solid;
border-collapse: collapse;
font-size: 25px;
}
.mytable{
width: 90%;
}
```



```
tr:nth-child(odd){
background-color: azure;
}
tr:nth-child(even){
background-color:lavender;;
}
input[type=submit]{
color: blue;
font-family: verdana;
font-size: 20px;
font-weight:bold;
padding: 5px 5px;
margin: 5px 5px;
}
input{
color: blue;
font-family: verdana;
font-size: 20px;
font-weight:bold;
padding: 5px 5px;
margin: 5px 5px;
}
</style>  </head>
<body>
<div align="center">
<form:form action="saveUpdateBook" method="post"  modelAttribute="mybook">
<table>
<tr>
<c:if test="${OpType eq 'ADD' }">
<td align="center" colspan="3">Add New Book</td>
</c:if>
<c:if test="${OpType eq 'UPDATE' }">
<td align="center" colspan="3">Update Book</td>
</c:if>
</tr>
<tr>
<td> Book Name</td>
<td><form:input path="bname" /></td>
<td><font color=red size=4><form:errors path="bname" /></font></td>
</tr>
<tr>
<td>Author</td>
<td><form:input path="author" /></td>
<td><font color=red size=4><form:errors path="author" /></font></td>
```



```
</tr>
<tr>
<td>Price</td>
<td><form:input path="price" /></td>
<td><font color=red size=4><form:errors path="price" /></font></td>
</tr>
<tr>
<td>Category</td>
<td><form:input path="category" /></td>
<td><font color=red size=4><form:errors path="category" /></font></td>
</tr>
<tr>
<td>Publications</td>
<td><form:input path="pub" /></td>
<td><font color=red size=4><form:errors path="pub" /></font></td>
</tr>
<tr>
<td align="center" colspan="3">
<input type="hidden" name="OpType" value="${OpType}" />
<c:if test="${OpType eq 'ADD' }">
<input type="submit" value="Add New Book" />
</c:if>
<c:if test="${OpType eq 'UPDATE' }">
<input type="submit" value="Update Book" />
<form:hidden path="bid" />
</c:if>
</td>
</tr> </table>
</form:form>
</div> </body> </html>
```

### **5.IndexController.java**

```
package com.coursecube.spring.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
@Controller
public class IndexController {
    @GetMapping("/")
    public String showIndexPage() {
        System.out.println("-----showIndexPage-----");
        return "index";
    }
}
```





## 6.BookController.java

```
package com.coursecube.spring.controller;

import java.util.List;
import javax.servlet.http.HttpServletRequest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import com.coursecube.spring.entity.Book;
import com.coursecube.spring.service.BookService;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
@Controller
@SessionAttributes("MyBooksList")
public class BookController {

    @Autowired
    BookService bookService;

    int start=0;
    int end=5;
    int totalDisplay=5;
    int totalBooks;

    private void showFirstPage(Model model) {
        start=0;
        end=5;
        totalDisplay=5;
        totalBooks=(int)bookService.getBookCount();

        model.addAttribute("FROM", start+1);
        model.addAttribute("TO",end);
        model.addAttribute("TotalBooks",totalBooks);

        model.addAttribute("ShowPrevious", "FALSE");
        model.addAttribute("ShowNext", "TRUE");

    }
}
```



```
@GetMapping("/showAllBooks")
public String showBooksList(Model model) {
    System.out.println("-----BookController--showBooksList()-----");
    List<Book> blist=bookService.getAllBooks(start,totalDisplay);
    model.addAttribute("MyBooksList", blist);
    showFirstPage(model);
    return "booksList";
}

@GetMapping("/nextBooks")
public String showNextBooks(Model model) {
    System.out.println("-----BookController--showNextBooks()-----");

    start=start+totalDisplay;
    if(start<=0) {
        start=0;
        model.addAttribute("ShowPrevious", "FALSE");
    }
    else {
        model.addAttribute("ShowPrevious", "TRUE");
    }
    model.addAttribute("FROM", start+1);

    end=end+totalDisplay;
    if(end>=totalBooks) {
        model.addAttribute("TO",totalBooks);
        model.addAttribute("ShowNext", "FALSE");
    } else {
        model.addAttribute("TO",end);
        model.addAttribute("ShowNext", "TRUE");
    }
}

List<Book> blist=bookService.getAllBooks(start,totalDisplay);
model.addAttribute("MyBooksList", blist);
model.addAttribute("TotalBooks",totalBooks);
return "booksList";
}

@GetMapping("/previousBooks")
public String showPreviousBooks(Model model) {
    System.out.println("-----BookController--showPreviousBooks()-----");

    start=start-totalDisplay;
```



```
if(start<=0) {
start=0;
model.addAttribute("ShowPrevious", "FALSE");
}
else {
model.addAttribute("ShowPrevious", "TRUE");
}
model.addAttribute("FROM", start+1);

end=end-totalDisplay;
if(end>=totalBooks) {
model.addAttribute("TO",totalBooks);
model.addAttribute("ShowNext", "FALSE");
} else {
model.addAttribute("TO",end);
model.addAttribute("ShowNext", "TRUE");
}

List<Book> blist=bookService.getAllBooks(start,totalDisplay);
model.addAttribute("MyBooksList", blist);

model.addAttribute("TotalBooks",totalBooks);
return "booksList";
}

@PostMapping("/addEditBookForm")
public String addBookForm(@RequestParam("bookId") Integer bookId,Model model) {
System.out.println("-----BookController--addEditBookForm()-----");
System.out.println(bookId);
Book book=new Book();
String opType="ADD";
if(bookId!=0) {
book=bookService.getBookById(bookId);
opType="UPDATE";
}
model.addAttribute("mybook", book);
model.addAttribute("OpType", opType);
showFirstPage(model);
return "addEditBook";
}
```



```
@PostMapping("/saveUpdateBook")
public String saveUpdateBook( @ModelAttribute("mybook") Book book,BindingResult
result,Model model,HttpServletRequest req) {
System.out.println("-----BookController--saveUpdateBook()-----");
String opType=req.getParameter("OpType");
System.out.println(opType);

if(opType.equals("ADD")) {
bookService.addBook(book);
}
if(opType.equals("UPDATE")) {
bookService.updateBook(book);
}

List<Book> blist=bookService.getAllBooks(start,totalDisplay);
model.addAttribute("MyBooksList", blist);
showFirstPage(model);
return "booksList";
}

@PostMapping("/deleteBook")
public String deleteBook(@RequestParam("bookId") Integer bookId,Model model) {
System.out.println("-----BookController--deleteBook()-----");
bookService.deleteBook(bookId);
List<Book> blist=bookService.getAllBooks(start,totalDisplay);
model.addAttribute("MyBooksList", blist);
showFirstPage(model);
return "booksList";
}

@GetMapping("/viewBook")
public String viewBook(@RequestParam("bookId") String bookId,Model model) {
System.out.println("-----BookController--viewBook()-----");
System.out.println(bookId);
Book book=bookService.getBookById(Integer.parseInt(bookId));
model.addAttribute("MyBook", book);
showFirstPage(model);
return "viewBook";
}
}
```



### 7.BookService.java

```
package com.coursecube.spring.service;

import java.util.List;
import com.coursecube.spring.entity.Book;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
public interface BookService {
    public List<Book> getAllBooks();
    public List<Book> getAllBooks(int start,int total);
    public Book getBookById(Integer bid);
    public void addBook(Book book);
    public void updateBook(Book book);
    public void deleteBook(Integer bid);
    public int getBookCount();
}
```

### 8.BookServiceImpl.java

```
package com.coursecube.spring.serviceimpl;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.coursecube.spring.dao.BookDAO;
import com.coursecube.spring.entity.Book;
import com.coursecube.spring.service.BookService;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
@Service
public class BookServiceImpl implements BookService{

    @Autowired
    BookDAO bookDAO;

    @Autowired
    MailService mailService;
```



```
@Override
public List<Book> getAllBooks() {
    System.out.println("-----BookServiceImpl--getAllBooks()-----");
    return bookDAO.getAllBooks();
}

@Override
public List<Book> getAllBooks(int start, int total) {
    System.out.println("-----BookServiceImpl--getAllBooks(start,total)-----");
    return bookDAO.getAllBooks(start,total);
}

@Override
public Book getBookById(Integer bid) {
    System.out.println("-----BookServiceImpl--getBookById()-----");
    return bookDAO.getBookById(bid);
}

@Override
public void addBook(Book book) {
    System.out.println("-----BookServiceImpl--addBook()-----");
    bookDAO.addBook(book);
    String from="<username>@gmail.com";
    String to="<username>@gmail.com";
    String subject="New Book @ JLC Book Store";
    String body="<font color=red size=5> New Book : "+ book.getBname() +" <br/> By  
"+book.getAuthor()+" <br/> at Rs." +book.getPrice();
    mailService.sendMail(from, to, subject, body);
}

@Override
public void updateBook(Book book) {
    System.out.println("-----BookServiceImpl--updateBook()-----");
    bookDAO.updateBook(book);
}

@Override
public void deleteBook(Integer bid) {
    System.out.println("-----BookServiceImpl--deleteBook()-----");
    bookDAO.deleteBook(bid);
}

public int getBookCount() {
    return bookDAO.getBookCount();
}
}
```



### 9.MailService.java

```
package com.coursecube.spring.service;

import java.io.File;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
public interface MailService {
    public void sendMail(String from,String to,String subject,String body) ;
    public void sendMail(String from,String to,String subject,String body,File file) ;
}
```

### 10.MailServiceImpl.java

```
package com.coursecube.spring.serviceimpl;

import java.io.File;
import javax.mail.internet.MimeMessage;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.FileSystemResource;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.stereotype.Service;
import com.coursecube.spring.service.MailService;

@Service
public class MailServiceImpl implements MailService {
    @Autowired
    JavaMailSender mailSender;

    public void sendMail(String from, String to, String subject, String body) {
        try {
            MimeMessage mimeMessage = mailSender.createMimeMessage();
            MimeMessageHelper mailMsg = new MimeMessageHelper(mimeMessage);
            mailMsg.setFrom(from);
            mailMsg.setTo(to);
            mailMsg.setSubject(subject);
            mailMsg.setText(body, true);
            mailSender.send(mimeMessage);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```





```
public void sendMail(String from, String to, String subject, String body, File file) {  
    try {  
        MimeMessage mimeTypeMessage = mailSender.createMimeMessage();  
        MimeMessageHelper mailMsg = new MimeMessageHelper(mimeMessage, true);  
        mailMsg.setFrom(from);  
        mailMsg.setTo(to);  
        mailMsg.setSubject(subject);  
        mailMsg.setText(body, true);  
        FileSystemResource fileRes = new FileSystemResource(file);  
        mailMsg.addAttachment(file.getName(), fileRes);  
        mailSender.send(mimeMessage);  
    } catch (Exception ex) { ex.printStackTrace(); }  
}
```

### **11.BookDAO.java**

```
package com.coursecube.spring.dao;  
  
import java.util.List;  
import com.coursecube.spring.entity.Book;  
/*  
 * @Author : Srinivas Dande  
 * @company : Java Learning Center  
 */  
public interface BookDAO {  
    public List<Book> getAllBooks();  
    public List<Book> getAllBooks(int start,int total);  
    public Book getBookById(Integer bid);  
    public void addBook(Book book);  
    public void updateBook(Book book);  
    public void deleteBook(Integer bid);  
    public int getBookCount();  
}
```

### **12.BookDAOImpl.java**

```
package com.coursecube.spring.daoimpl;  
  
import java.math.BigInteger;  
import java.util.List;  
import org.hibernate.*;  
import org.hibernate.criterion.DetachedCriteria;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.orm.hibernate5.HibernateTemplate;  
import org.springframework.stereotype.Repository;  
import org.springframework.transaction.annotation.Transactional;
```





```
import com.coursecube.spring.dao.BookDAO;
import com.coursecube.spring.entity.Book;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
@Repository
@Transactional
public class BookDAOImpl implements BookDAO{

    @Autowired
    HibernateTemplate hibernateTemp;

    @Override
    public List<Book> getAllBooks() {
        System.out.println("-----BookDAOImpl--getAllBooks()-----");
        return hibernateTemp.loadAll(Book.class);
    }
    @Override
    public List<Book> getAllBooks(int start, int total) {
        System.out.println("-----BookDAOImpl--getAllBooks(start,total)-----");

        DetachedCriteria dc=DetachedCriteria.forClass(Book.class);
        return (List<Book>) hibernateTemp.findByCriteria(dc,start,total);
    }

    @Override
    public Book getBookById(Integer bid) {
        System.out.println("-----BookDAOImpl--getBookById()-----");
        return hibernateTemp.load(Book.class, bid, LockMode.READ);
    }
    @Override
    public void addBook(Book book) {
        System.out.println("-----BookDAOImpl--addBook()-----");
        hibernateTemp.save(book);
    }
    @Override
    public void updateBook(Book book) {
        System.out.println("-----BookDAOImpl--updateBook()-----");
        hibernateTemp.update(book);
    }
    @Override
    public void deleteBook(Integer bid) {
```



```
        System.out.println("-----BookDAOImpl--deleteBook()-----");
        Book book = hibernateTemp.get(Book.class, bid);
        if (book != null)
            hibernateTemp.delete(book);
    }
    public int getBookCount() {
        SessionFactory sessionFactory=hibernateTemp.getSessionFactory();
        Session session=sessionFactory.openSession();
        String sql="select count(*) from mybooks";

        BigInteger big=(BigInteger) session.createNativeQuery(sql).uniqueResult();
        return big.intValue();
    }
}
```

### **13.Book.java**

```
package com.coursecube.spring.entity;

import java.math.BigDecimal;
import javax.persistence.*;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
@Entity
@Table(name="mybooks")
public class Book {
    @Id
    @Column(name="bid")
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer bid;

    @Column(name="bname")
    private String bname;

    @Column(name="author")
    private String author;

    @Column(name="price")
    private BigDecimal price;

    @Column(name="category")
    private String category;
```



```
@Column(name="pub")
private String pub;

//Constructors
//Setters and Getters

@Override
public String toString() {
    return "[" + bid + " , " + bname + " , " + author + " , " + price + " , " + category + " , " + pub + "]";
}
}
```

#### **14.JLCWebConfig.java**

```
package com.coursecube.spring.config;

import java.util.Properties;
import javax.sql.DataSource;
import org.hibernate.SessionFactory;
import org.springframework.context.*;
import org.springframework.context.annotation.*;
import org.springframework.context.support.ReloadableResourceBundleMessageSource;
import org.springframework.jdbc.datasource.DriverManagerDataSource;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.JavaMailSenderImpl;
import org.springframework.orm.hibernate5.*;
import org.springframework.transaction.PlatformTransactionManager;
import org.springframework.transaction.annotation.EnableTransactionManagement;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.view.*;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
@EnableWebMvc
@Configuration
@EnableTransactionManagement
@ComponentScan({ "com.coursecube.spring" })
public class JLCWebConfig {

    @Bean
    public InternalResourceViewResolver viewResolver() {
        InternalResourceViewResolver viewResolver = new InternalResourceViewResolver();
        viewResolver.setViewClass(JstlView.class);
        viewResolver.setPrefix("/WEB-INF/myjsps/");
    }
}
```



```
viewResolver.setSuffix(".jsp");
return viewResolver;
}

@Bean
public MessageSource messageSource() {
    ReloadableResourceBundleMessageSource messageSource = new
    ReloadableResourceBundleMessageSource();
    messageSource.setBasename("classpath:messages");
    messageSource.setDefaultEncoding("UTF-8");
    return messageSource;
}

@Bean()
public HibernateTemplate hibernateTemplate(SessionFactory sessionFactory) {
    return new HibernateTemplate(sessionFactory);
}

@Bean
public PlatformTransactionManager transactionManager(SessionFactory sessionFactory) {
    return new HibernateTransactionManager(sessionFactory);
}

@Bean
public LocalSessionFactoryBean sessionFactory(DataSource dataSource) {
    LocalSessionFactoryBean sessionFactoryBean = new LocalSessionFactoryBean();
    sessionFactoryBean.setDataSource(dataSource);

    Properties props = new Properties();
    props.put("hibernate.show_sql", "true");
    props.put("hibernate.hbm2ddl.auto", "update");
    props.put("hibernate.transaction.factory_class",
    "org.hibernate.transaction.JDBCTransactionFactory");

    sessionFactoryBean.setHibernateProperties(props);
    sessionFactoryBean.setPackagesToScan("com.coursecube.spring.entity");
    return sessionFactoryBean;
}

@Bean
public DataSource dataSource() {
    DriverManagerDataSource ds = new DriverManagerDataSource();
    ds.setDriverClassName("com.mysql.jdbc.Driver");
    ds.setUrl("jdbc:mysql://localhost:3306/bookstoredb");
    ds.setUsername("root");
    ds.setPassword("srinivas");
    return ds;
}
```



```
@Bean
public JavaMailSender javaMailSenderImpl() {
    JavaMailSenderImpl mailSender = new JavaMailSenderImpl();
    mailSender.setHost("smtp.gmail.com");
    mailSender.setPort(465);

    mailSender.setUsername("<username>@gmail.com");
    mailSender.setPassword("<password>");

    Properties prop = mailSender.getJavaMailProperties();
    prop.put("mail.smtp.host", "smtp.gmail.com");
    prop.put("mail.smtp.socketFactory.port", "465");
    prop.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    prop.put("mail.smtp.auth", "true");
    prop.put("mail.smtp.starttls.enable", "true");
    return mailSender;
}
}
```

### **15. JLCWebAppInitializer.java**

```
package com.coursecube.spring.config;

import org.springframework.web.servlet.support.*;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */
public class JLCWebAppInitializer extends
AbstractAnnotationConfigDispatcherServletInitializer {
    @Override
    protected Class<?>[] getRootConfigClasses() {
        return new Class[] { JLCWebConfig.class };
    }
    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] { JLCWebConfig.class };
    }
    @Override
    protected String[] getServletMappings() {
        return new String[] { "/" };
    }
}
```