**16 Spring Assignment 2**

**MyWebAppInitializer**

**package** com.cognicant.fsd.spring.config;

**import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

**public** **class** MyWebAppInitializer **extends**

AbstractAnnotationConfigDispatcherServletInitializer{

@Override

**protected** Class<?>[] getRootConfigClasses() {

**return** **new** Class[]{RootConfig.**class**};

}

@Override

**protected** Class<?>[] getServletConfigClasses() {

**return** **new** Class[]{WebConfig.**class**};

}

@Override

**protected** String[] getServletMappings() {

**return** **new** String[]{"/"};

}

}

**RootConfig**

package com.cognicant.fsd.spring.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import com.cognicant.fsd.spring.datastream.BinaryBookStreamCRUD;

import com.cognicant.fsd.spring.datastream.BinarySubjectStreamCRUD;

@Configuration

public class RootConfig {

@Bean

BinarySubjectStreamCRUD binarySubjectStreamCRUD() {

return new BinarySubjectStreamCRUD();

}

@Bean

BinaryBookStreamCRUD binaryBookStreamCRUD() {

return new BinaryBookStreamCRUD();

}

}

package com.cognicant.fsd.spring.config;

import java.util.ArrayList;

import java.util.List;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

import org.springframework.web.servlet.view.InternalResourceViewResolver;

import org.springframework.web.servlet.view.JstlView;

@Configuration

@EnableWebMvc

@ComponentScan(basePackages = { "com.cognizant.fsd.spring.controller","com.cognicant.fsd.spring.service","com.cognicant.fsd.spring.repository" })

public class WebConfig extends WebMvcConfigurerAdapter {

@Bean

public InternalResourceViewResolver resolver() {

InternalResourceViewResolver resolver = new InternalResourceViewResolver();

resolver.setViewClass(JstlView.class);

resolver.setPrefix("/WEB-INF/views/");

resolver.setSuffix(".jsp");

return resolver;

}

}

**Book**

/\*\*

\*

\*/

package com.cognicant.fsd.spring.model;

import java.io.Serializable;

import java.time.LocalDate;

import org.springframework.format.annotation.DateTimeFormat;

/\*\*

\* @author Saugata Ray

\*

\*/

public class Book implements Serializable {

private static final long serialVersionUID = 3667779253735136971L;

private Long bookId;

private String title;

private Double price;

private Integer volume;

@DateTimeFormat(pattern = "yyyy-MM-dd")

private LocalDate publishDate;

public Long getBookId() {

return bookId;

}

public void setBookId(Long bookId) {

this.bookId = bookId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public Integer getVolume() {

return volume;

}

public void setVolume(Integer volume) {

this.volume = volume;

}

public LocalDate getPublishDate() {

return publishDate;

}

public void setPublishDate(LocalDate publishDate) {

this.publishDate = publishDate;

}

@Override

public String toString() {

return "Book [bookId=" + bookId + ", title=" + title + ", price=" + price + ", volume=" + volume

+ ", publishDate=" + publishDate + "]";

}

}

**Subject**

package com.cognicant.fsd.spring.model;

import java.io.Serializable;

import java.util.Set;

/\*\*

\* @author Saugata Ray

\*

\*/

public class Subject implements Serializable {

private static final long serialVersionId = 1L;

private Long subjectId;

private String subTitle;

private Integer durationInHours;

private Set<Book> references;

public Long getSubjectId() {

return subjectId;

}

public void setSubjectId(Long subjectId) {

this.subjectId = subjectId;

}

public String getSubTitle() {

return subTitle;

}

public void setSubTitle(String subTitle) {

this.subTitle = subTitle;

}

public Integer getDurationInHours() {

return durationInHours;

}

public void setDurationInHours(Integer durationInHours) {

this.durationInHours = durationInHours;

}

public Set<Book> getReferences() {

return references;

}

public void setReferences(Set<Book> references) {

this.references = references;

}

@Override

public String toString() {

return "Subject [subjectId=" + subjectId + ", subTitle=" + subTitle + ", durationInHours=" + durationInHours

+ ", references=" + references + "]";

}

}

**HomeController**

**package** com.cognizant.fsd.spring.controller;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**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.bind.annotation.RequestParam;

**import** com.cognicant.fsd.spring.model.Book;

**import** com.cognicant.fsd.spring.model.Subject;

**import** com.cognicant.fsd.spring.service.BookService;

**import** com.cognicant.fsd.spring.service.SubjectService;

@Controller

**public** **class** HomeController {

@Autowired

**private** SubjectService subjectService;

@Autowired

**private** BookService bookService;

@RequestMapping(path = { "/" }, method = RequestMethod.***GET***)

**public** String index(Model model) {

**return** "index";

}

@RequestMapping(value = "/addSubject", method = RequestMethod.***GET***)

**public** String addSubject(Model model) {

System.***out***.println("add Subject 2");

Subject subject = **new** Subject();

model.addAttribute("subject", subject);

**return** "add-subject";

}

@RequestMapping(value = "/saveSubject", method = RequestMethod.***POST***)

**public** String saveSubject(@ModelAttribute Subject subject, Model model) {

System.***out***.println("save Subject " + subject.getSubTitle());

model.addAttribute("ENTITY\_NAME", "Subject");

subject = subjectService.addSubject(subject);

model.addAttribute("ENTITY\_ID", subject.getSubjectId());

**return** "success";

}

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

**public** String addBook(Model model) {

System.***out***.println("add Book");

model.addAttribute("book", **new** Book());

**return** "add-book";

}

@RequestMapping(value = "/saveBook", method = RequestMethod.***POST***)

**public** String saveBook(@ModelAttribute Book book, Model model) {

System.***out***.println("save book " + book.getPublishDate());

model.addAttribute("ENTITY\_NAME", "Book");

book = bookService.addBook(book);

model.addAttribute("ENTITY\_ID", book.getBookId());

**return** "success";

}

@RequestMapping(value = "/deleteBook", method = RequestMethod.***GET***)

**public** String deleteBook(Model model) {

System.***out***.println("delete Book");

**return** "delete-book";

}

@RequestMapping(value = "/performDeleteBook", method = RequestMethod.***GET***)

**public** String performDeleteBook(@RequestParam("bookId") String bookId, Model model) {

System.***out***.println("delete Book " + bookId);

**boolean** isSuccess = bookService.deleteBook(Long.*parseLong*(bookId));

model.addAttribute("ENTITY\_NAME", "Book");

model.addAttribute("ENTITY\_ID", bookId);

model.addAttribute("isSuccess", isSuccess);

**return** "delete-success";

}

@RequestMapping(value = "/deleteSubject", method = RequestMethod.***GET***)

**public** String deleteSubject(Model model) {

System.***out***.println("delete Book");

**return** "delete-subject";

}

@RequestMapping(value = "/performDeleteSubject", method = RequestMethod.***GET***)

**public** String performDeleteSubject(@RequestParam("subjectId") String subjectId, Model model) {

System.***out***.println("delete subjectId " + subjectId);

**boolean** isSuccess = subjectService.deleteSubject(Long.*parseLong*(subjectId));

model.addAttribute("ENTITY\_NAME", "Subject");

model.addAttribute("isSuccess", isSuccess);

model.addAttribute("ENTITY\_ID", subjectId);

**return** "delete-success";

}

@RequestMapping(value = "/search", method = RequestMethod.***GET***)

**public** String search(Model model) {

System.***out***.println("search Subject");

**return** "search";

}

@RequestMapping(value = "/performSearch", method = RequestMethod.***GET***)

**public** String performSearch(@RequestParam("id") String id, @RequestParam("entityName") String entityName,

Model model) {

System.***out***.println(entityName + "Perform Search" + id);

Subject subject = **null**;

Book book = **null**;

String name = "";

**if** (entityName.equals("subject")) {

subject = subjectService.searchSubject(Long.*parseLong*(id));

name = "subject";

}

**if** (entityName.equals("book")) {

book = bookService.searchBook(Long.*parseLong*(id));

name = "book";

}

model.addAttribute("ENTITY\_NAME", name);

model.addAttribute("book", book);

model.addAttribute("subject", subject);

**return** "result";

}

}

**BookService**

package com.cognicant.fsd.spring.service;

import java.util.List;

import com.cognicant.fsd.spring.model.Book;

public interface BookService {

public Book addBook(Book book);

public boolean deleteBook(long bookId);

public Book searchBook(long bookId);

public List<Book> fetchAllBook();

}

**BookServiceImpl**

package com.cognicant.fsd.spring.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.cognicant.fsd.spring.model.Book;

import com.cognicant.fsd.spring.repository.BookRepository;

@Service("bookService")

public class BookServiceImpl implements BookService {

@Autowired

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

@Override

public Book addBook(Book book) {

return bookRepository.addBook(book);

}

@Override

public boolean deleteBook(long bookId) {

return bookRepository.deleteBook(bookId);

}

@Override

public Book searchBook(long bookId) {

return bookRepository.searchBook(bookId);

}

@Override

public List<Book> fetchAllBook() {

return bookRepository.fetchAllBook();

}

}

**SubjectService**

package com.cognicant.fsd.spring.service;

import java.util.List;

import com.cognicant.fsd.spring.model.Subject;

public interface SubjectService {

public Subject addSubject(Subject subject);

public boolean deleteSubject(long subjectId);

public Subject searchSubject(long subjectId);

public List<Subject> fetchAllSubject();

}

**SubjectServiceImpl**

package com.cognicant.fsd.spring.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.cognicant.fsd.spring.model.Subject;

import com.cognicant.fsd.spring.repository.SubjectRepository;

@Service("subjectService")

public class SubjectServiceImpl implements SubjectService{

@Autowired

private SubjectRepository subjectRepository;

public void setSubjectRepository(SubjectRepository subjectRepository) {

this.subjectRepository = subjectRepository;

}

@Override

public Subject addSubject(Subject subject) {

return subjectRepository.addSubject(subject);

}

@Override

public boolean deleteSubject(long subjectId) {

return subjectRepository.deleteSubject(subjectId);

}

@Override

public Subject searchSubject(long subjectId) {

return subjectRepository.searchSubject(subjectId);

}

@Override

public List<Subject> fetchAllSubject() {

return subjectRepository.fetchAllSubject();

}

}

**BookRepository**

package com.cognicant.fsd.spring.repository;

import java.util.List;

import com.cognicant.fsd.spring.model.Book;

public interface BookRepository {

public Book addBook(Book book);

public boolean deleteBook(long bookId);

public Book searchBook(long bookId);

public List<Book> fetchAllBook();

}

**BookRepositoryImpl**

package com.cognicant.fsd.spring.repository;

import java.util.ArrayList;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import com.cognicant.fsd.spring.datastream.BinaryBookStreamCRUD;

import com.cognicant.fsd.spring.model.Book;

import com.cognicant.fsd.spring.model.Subject;

@Repository("bookRepository")

public class BookRepositoryImpl implements BookRepository{

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

@Autowired

private BinaryBookStreamCRUD binaryBookStreamCRUD;

@Override

public Book addBook(Book book) {

bookList.add(book);

saveToBinaryFile(bookList);

return book;

}

@Override

public boolean deleteBook(long bookId) {

boolean flag=bookList.removeIf(book->book.getBookId()==bookId);

saveToBinaryFile(bookList);

return flag;

}

@Override

public Book searchBook(long bookId) {

Book book=null;

try {

bookList=binaryBookStreamCRUD.readRecords();

book= bookList.stream()

.filter(b -> b.getBookId() == bookId).reduce((a, b) -> {

throw new IllegalStateException("Multiple elements: " + a + ", " + b);

}).get();

}catch(Exception exception) {}

return book;

}

public void saveToBinaryFile(List<Book> bookList) {

try {

binaryBookStreamCRUD.open();

bookList.forEach(b->{

binaryBookStreamCRUD.writeBook(b);

});

binaryBookStreamCRUD.close();

} catch (Exception e) {

e.printStackTrace();

}

}

@Override

public List<Book> fetchAllBook() {

return bookList;

}

}

**SubjectRepository**

package com.cognicant.fsd.spring.repository;

import java.util.List;

import com.cognicant.fsd.spring.model.Subject;

public interface SubjectRepository {

public Subject addSubject(Subject subject);

public boolean deleteSubject(long subjectId);

public Subject searchSubject(long subjectId);

public List<Subject> fetchAllSubject();

}

**SubjectRepositoryImpl**

package com.cognicant.fsd.spring.repository;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import com.cognicant.fsd.spring.datastream.BinarySubjectStreamCRUD;

import com.cognicant.fsd.spring.model.Book;

import com.cognicant.fsd.spring.model.Subject;

@Repository("subjectRepository")

public class SubjectRepositoryImpl implements SubjectRepository{

private List<Subject> subjectList=new ArrayList<Subject>();

@Autowired

private BinarySubjectStreamCRUD binarySubjectStreamCRUD;

@Override

public Subject addSubject(Subject subject) {

subjectList.add(subject);

saveToBinaryFile(subjectList);

return subject;

}

@Override

public boolean deleteSubject(long subjectId) {

boolean flag=subjectList.removeIf(sub->sub.getSubjectId()==subjectId);

saveToBinaryFile(subjectList);

return flag;

}

@Override

public Subject searchSubject(long subjectId) {

Subject subject=null;

try {

subjectList=binarySubjectStreamCRUD.readRecords();

System.out.println("subjectList "+subjectList.size());

subject = subjectList.stream()

.filter(sub -> sub.getSubjectId() == subjectId).reduce((a, b) -> {

throw new IllegalStateException("Multiple elements: " + a + ", " + b);

}).get();

}catch(Exception exception) {}

return subject;

}

public void saveToBinaryFile(List<Subject> subjectList) {

try {

binarySubjectStreamCRUD.open();

subjectList.forEach(sub->{

binarySubjectStreamCRUD.writeSubject(sub);

});

binarySubjectStreamCRUD.close();

} catch (Exception e) {

e.printStackTrace();

}

}

@Override

public List<Subject> fetchAllSubject() {

return subjectList;

}

}

**BinaryBookStreamCRUD**

**package** com.cognicant.fsd.spring.datastream;

**import** java.io.BufferedOutputStream;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeFormatter;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.beans.factory.InitializingBean;

**import** com.cognicant.fsd.spring.model.Book;

**public** **class** BinaryBookStreamCRUD **implements** InitializingBean{

**private** String fileName = "C:\\Saugata\\binaryFile\\book.dat";

**private** DataOutputStream outStream = **null**;

**public** DataOutputStream openOutputStream(String name) **throws** Exception {

DataOutputStream out = **null**;

File file = **new** File(name);

out = **new** DataOutputStream(**new** BufferedOutputStream(**new** FileOutputStream(file)));

**return** out;

}

**public** **void** writeBook(Book book){

writeRecords(book,fileName);

}

**private** **void** writeRecords(Book book, String fileName ) {

**try** {

outStream.writeLong(book.getBookId());

outStream.writeUTF(book.getTitle());

outStream.writeInt(book.getVolume());

outStream.writeDouble(book.getPrice());

outStream.writeUTF(book.getPublishDate().toString());

} **catch** (IOException e) {

System.***out***.println("IOERROR: " + e.getMessage() + "\n");

}

} // writeRecords()

**public** List<Book> readRecords(){

List<Book> bookList=**new** ArrayList<Book>();

**try** {

DataInputStream inStream = **new** DataInputStream(**new** FileInputStream(fileName));

**while** (**true**) { // Infinite loop

Long bookId = inStream.readLong(); // Read a record

String title = inStream.readUTF();

Integer volume = inStream.readInt();

Double price = inStream.readDouble();

String publishDate = inStream.readUTF();

Book book=**new** Book();

book.setBookId(bookId);

book.setTitle(title);

book.setVolume(volume);

book.setPrice(price);

book.setPublishDate(stringToDate(publishDate));

bookList.add(book);

} // while

} **catch** (Exception e) {

System.***out***.println(bookList.size());

System.***err***.println("eeeeeeee "+e);

}

**return** bookList;

}

**public** **void** open() **throws** Exception {

**this**.outStream= **new** DataOutputStream(**new** FileOutputStream(fileName));

}

**public** **void** close() **throws** Exception {

**this**.outStream.close();

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

//this.out = openOutputStream("data.dat");

}

**public** LocalDate stringToDate(String date) {

DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("yyyy-MM-dd");

//convert String to LocalDate

**return** LocalDate.*parse*(date, formatter);

}

}

**BinarySubjectStreamCRUD**

**package** com.cognicant.fsd.spring.datastream;

**import** java.io.BufferedOutputStream;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.beans.factory.InitializingBean;

**import** com.cognicant.fsd.spring.model.Subject;

**public** **class** BinarySubjectStreamCRUD **implements** InitializingBean {

**private** DataOutputStream outStream =**null**;

**private** String fileName = "C:\\Saugata\\binaryFile\\subject.dat";

**public** DataOutputStream openOutputStream(String name) **throws** Exception {

DataOutputStream out = **null**;

File file = **new** File(name);

out = **new** DataOutputStream(**new** BufferedOutputStream(**new** FileOutputStream(file)));

**return** out;

}

**public** **void** writeSubject(Subject subject){

writeRecords(subject, fileName);

}

**private** **void** writeRecords(Subject subject, String fileName ) {

**try** {

// Open stream

outStream.writeLong(subject.getSubjectId());

outStream.writeUTF(subject.getSubTitle());

outStream.writeInt(subject.getDurationInHours());

//outStream.close(); // Close the stream

} **catch** (IOException e) {

System.***out***.println("IOERROR: " + e.getMessage() + "\n");

}

} // writeRecords()

**public** List<Subject> readRecords(){

List<Subject> subjectList=**new** ArrayList<Subject>();

**try** {

DataInputStream inStream = **new** DataInputStream(**new** FileInputStream(fileName));

**while** (**true**) { // Infinite loop

Long subjectId = inStream.readLong(); // Read a record

String title = inStream.readUTF();

Integer durationInHours = inStream.readInt();

Subject subject=**new** Subject();

subject.setSubjectId(subjectId);;

subject.setSubTitle(title);

subject.setDurationInHours(durationInHours);

subjectList.add(subject);

} // while

} **catch** (Exception e) {

System.***out***.println(subjectList.size());

System.***err***.println("eeeeeeee "+e);

}

**return** subjectList;

}

**public** **void** open() **throws** Exception {

**this**.outStream= **new** DataOutputStream(**new** FileOutputStream(fileName));

}

**public** **void** close() **throws** Exception {

**this**.outStream.close();

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

//this.out = openOutputStream("data.dat");

}

}

**index.jsp**

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

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

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

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Spring 4 MVC - Hello World Example | BORAJI.COM</title>

</head>

<body>

<h1>Welcome to Spring MVC</h1>

<a href=*"*<spring:url value=*"/addSubject"*/>*"*>Add Subject</a><br/>

<a href=*"*<spring:url value=*"/addBook"*/>*"*>Add Book</a><br/>

<a href=*"*<spring:url value=*"/deleteSubject"*/>*"*>Delete Subject</a><br/>

<a href=*"*<spring:url value=*"/deleteBook"*/>*"*>Delete Book</a><br/>

<a href=*"*<spring:url value=*"/search"*/>*"*>Search</a><br/>

</body>

</html>

**addBook**

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<table style="width: *100%*">

<form:form method=*"POST"* commandName=*"book"* action=*"*${pageContext.request.contextPath}*/saveBook"*>

<tr>

<td><form:label path=*"bookId"*>Book Id :</form:label></td>

<td><form:input path=*"bookId"* /></td>

</tr>

<tr>

<td><form:label path=*"title"*>Book Title :</form:label></td>

<td><form:input path=*"title"* /></td>

</tr>

<tr>

<td><form:label path=*"price"*>Price :</form:label></td>

<td><form:input path=*"price"* /></td>

</tr>

<tr>

<td><form:label path=*"volume"*>Volume :</form:label></td>

<td><form:input path=*"volume"* /></td>

</tr>

<tr>

<td><form:label path=*"price"*>Book Price :</form:label></td>

<td><form:input path=*"price"* /></td>

</tr>

<tr>

<td><form:label path=*"publishDate"*>Publish Date (yyyy-MM-dd):</form:label></td>

<td><form:input path=*"publishDate"* /></td>

</tr>

<tr>

<td colspan=*"2"*><input type=*"submit"* value=*"save"*/></td>

</tr>

</form:form>

</table>

</body>

</html>

**add-subject**

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<table style="width:*100%*">

<form:form method=*"POST"* commandName=*"subject"* action=*"*${pageContext.request.contextPath}*/saveSubject"* >

<tr>

<td><form:label path=*"subjectId"*>Subject Id :</form:label></td>

<td><form:input path=*"subjectId"*/></td>

</tr>

<tr>

<td><form:label path=*"subTitle"*>Subject Title :</form:label></td>

<td><form:input path=*"subTitle"*/></td>

</tr>

<tr>

<td><form:label path=*"durationInHours"*>Duration In Hours :</form:label></td>

<td><form:input path=*"durationInHours"*/></td>

</tr>

<tr>

<td colspan=*"2"*><input type=*"submit"* value=*"save"*/></td>

</tr>

</form:form>

</table>

</body>

</html>

**delete-book**

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

pageEncoding=*"ISO-8859-1"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>enter the book id to delete :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performDeleteBook"*>

<input id=*"bookId"* name=*"bookId"*/>

<input type=*"submit"* value=*"delete"*>

</form>

</body>

</html>

**delete-subject**

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

pageEncoding=*"ISO-8859-1"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>enter the subject id to delete :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performDeleteSubject"*>

<input id=*"subjectId"* name=*"subjectId"*/>

<input type=*"submit"* value=*"delete"*>

</form>

</body>

</html>

**delete-subject**

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

pageEncoding=*"ISO-8859-1"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>enter the subject id to delete :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performDeleteSubject"*>

<input id=*"subjectId"* name=*"subjectId"*/>

<input type=*"submit"* value=*"delete"*>

</form>

</body>

</html>

**delete-success**

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

pageEncoding=*"ISO-8859-1"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>The ${ENTITY\_NAME} with id (${ENTITY\_ID}) has been deleted successfully</h1>

</body>

</html>

**result**

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

pageEncoding=*"ISO-8859-1"*%>

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

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<c:choose>

<c:when test=*"*${ENTITY\_NAME **eq** 'subject'}*"*>

<table>

<tr>

<td>Subject Id :</td>

<td>${subject.subjectId}</td>

</tr>

<tr>

<td>Subject Title :</td>

<td>${subject.subTitle}</td>

</tr>

<tr>

<td>Duration In Hours :</td>

<td>${subject.durationInHours}</td>

</tr>

</table>

</c:when>

<c:when test=*"*${ENTITY\_NAME **eq** 'book'}*"*>

<table style="width:*100%*">

<tr>

<td>Book Id :</td>

<td>${book.bookId}</td>

</tr>

<tr>

<td>Book Title :</td>

<td>${book.title}</td>

</tr>

<tr>

<td>Price :</td>

<td>${book.price}</td>

</tr>

<tr>

<td>Volume :</td>

<td>${book.volume}</td>

</tr>

<tr>

<td>Publish Date (yyyy-MM-dd) :</td>

<td>${book.publishDate}</td>

</tr>

</table>

</c:when>

<c:otherwise>

</c:otherwise>

</c:choose>

</body>

</html>

search

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

pageEncoding=*"ISO-8859-1"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>please choose entity & enter the id :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performSearch"*>

<input type=*"radio"* id=*"entityName1"* name=*"entityName"* value=*"subject"* checked> Subject<br>

<input type=*"radio"* id=*"entityName2"* name=*"entityName"* value=*"book"*> Book<br>

<input type=*"text"* id=*"id"* name=*"id"*/>

<input type=*"submit"* value=*"Search"*/>

</form>

</body>

</html>

**success**

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

pageEncoding=*"ISO-8859-1"*%>

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>The ${ENTITY\_NAME} with id (${ENTITY\_ID}) has been saved successfully</h1>

</body>

</html>





















































