# MyWebAppInitializer.java

* **package** com.cognizant.fsd.spring.config;
* **import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
* /\*\*
* \* **@author** Sudhir Kumar Thakur
* \*/
* **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.java

* package com.cognizant.fsd.spring.config;
* import org.springframework.context.annotation.Bean;
* import org.springframework.context.annotation.Configuration;
* import com.cognizant.fsd.spring.datastream.BinaryBookStreamCRUD;
* import com.cognizant.fsd.spring.datastream.BinarySubjectStreamCRUD;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*/
* @Configuration
* public class RootConfig {
* @Bean
* BinarySubjectStreamCRUD binarySubjectStreamCRUD() {
* return new BinarySubjectStreamCRUD();
* }
* @Bean
* BinaryBookStreamCRUD binaryBookStreamCRUD() {
* return new BinaryBookStreamCRUD();
* }
* }

# WebConfig.java

* package com.cognizant.fsd.spring.config;
* 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;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*/
* @Configuration
* @EnableWebMvc
* @ComponentScan(basePackages = { "com.cognizant.fsd.spring.controller","com.cognizant.fsd.spring.service","com.cognizant.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;
* }
* }

# HomeController.java

* 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.cognizant.fsd.spring.model.Book;
* import com.cognizant.fsd.spring.model.Subject;
* import com.cognizant.fsd.spring.service.BookService;
* import com.cognizant.fsd.spring.service.SubjectService;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*/
* @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("Inside addSubject()");
* 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("Saving Subject with title : " + 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("Inside addBook()");
* 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("Saving book with title : " + book.getTitle());
* 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("Inside deleteBook()");
* return "delete-book";
* }
* @RequestMapping(value = "/performDeleteBook", method = RequestMethod.GET)
* public String performDeleteBook(@RequestParam("bookId") String bookId, Model model) {
* System.out.println("Deleting Book with ID : " + 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("Inside deleteSubject()");
* return "delete-subject";
* }
* @RequestMapping(value = "/performDeleteSubject", method = RequestMethod.GET)
* public String performDeleteSubject(@RequestParam("subjectId") String subjectId, Model model) {
* System.out.println("Deleting subject with ID : " + 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("Inside search()");
* 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 + " Performing 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";
* }
* }

# BinaryBookStreamCRUD.java

* package com.cognizant.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.cognizant.fsd.spring.model.Book;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*/
* public class BinaryBookStreamCRUD implements InitializingBean{
* private String fileName = "C:\\Spring\\File\\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.java

* package com.cognizant.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.cognizant.fsd.spring.model.Subject;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*/
* public class BinarySubjectStreamCRUD implements InitializingBean {
* private DataOutputStream outStream =null;
* private String fileName = "C:\\Spring\\File\\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");
* }
* }

# Book.java

* package com.cognizant.fsd.spring.model;
* import java.io.Serializable;
* import java.time.LocalDate;
* import org.springframework.format.annotation.DateTimeFormat;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*
* \*/
* 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.java

* package com.cognizant.fsd.spring.model;
* import java.io.Serializable;
* import java.util.Set;
* /\*\*
* \* @author Sudhir Kumar Thakur
* \*
* \*/
* 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 + "]";
* }
* }

# BookRepository.java

* package com.cognizant.fsd.spring.repository;
* import java.util.List;
* import com.cognizant.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.java

* package com.cognizant.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.cognizant.fsd.spring.datastream.BinaryBookStreamCRUD;
* import com.cognizant.fsd.spring.model.Book;
* import com.cognizant.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.java

* package com.cognizant.fsd.spring.repository;
* import java.util.List;
* import com.cognizant.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.java

* **package** com.cognizant.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.cognizant.fsd.spring.datastream.BinarySubjectStreamCRUD;
* **import** com.cognizant.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;
* }
* }

# BookService.java

* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import com.cognizant.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.java

* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.stereotype.Service;
* import com.cognizant.fsd.spring.model.Book;
* import com.cognizant.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.java

* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import com.cognizant.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.java

* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.stereotype.Service;
* import com.cognizant.fsd.spring.model.Subject;
* import com.cognizant.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();
* }
* }

# add-book.jsp

* <%@ 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.jsp

* <%@ 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.jsp

* <%@ 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.jsp

* <%@ 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.jsp

* <%@ 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>

# 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 Assignment 2</title>
* </head>
* <body>
* <h1>Please select from Menu Items Below</h1>
* <ul id=*"menu"*>
* <li><a href=*"*<spring:url value=*"/addSubject"*/>*"*>Add Subject</a></li>
* <li><a href=*"*<spring:url value=*"/addBook"*/>*"*>Add Book</a></li>
* <li><a href=*"*<spring:url value=*"/deleteSubject"*/>*"*>Delete Subject</a></li>
* <li><a href=*"*<spring:url value=*"/deleteBook"*/>*"*>Delete Book</a></li>
* <li><a href=*"*<spring:url value=*"/search"*/>*"*>Search</a></li>
* </ul>
* </body>
* </html>
* <script type=*"text/javascript"*>
* #menu li { padding: 0px; }
* #menu li a { margin: 0px; display: block; width: 100%; height: 100%; }
* </script>

# result.jsp

* <%@ 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.jsp

* <%@ 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&nbsp;
* <input type=*"radio"* id=*"entityName2"* name=*"entityName"* value=*"book"*> Book<br>
* <input type=*"text"* id=*"id"* style="margin-top:*10px*;" name=*"id"*/>
* <input type=*"submit"* value=*"Search"*/>
* </form>
* </body>
* </html>

# success.jsp

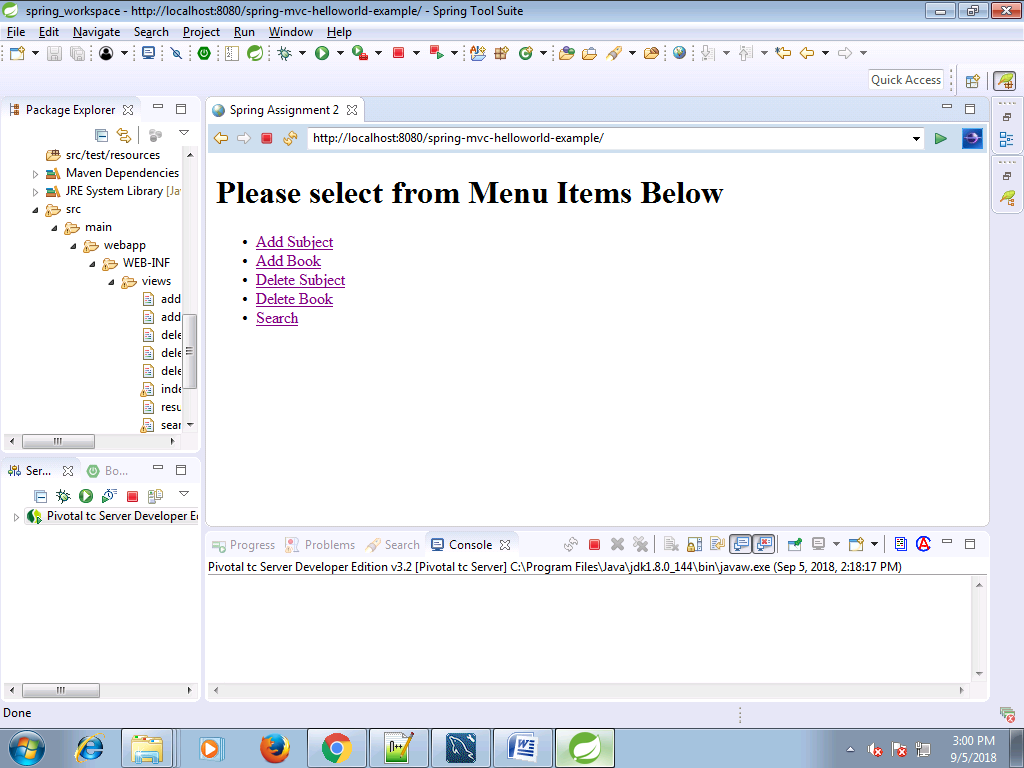
* <%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*
* pageEncoding=*"ISO-8859-1"*%>
* <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
* <%@ taglib prefix=*"spring"* uri=*"http://www.springframework.org/tags"*%>
* <html>
* <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>

# pom.xml

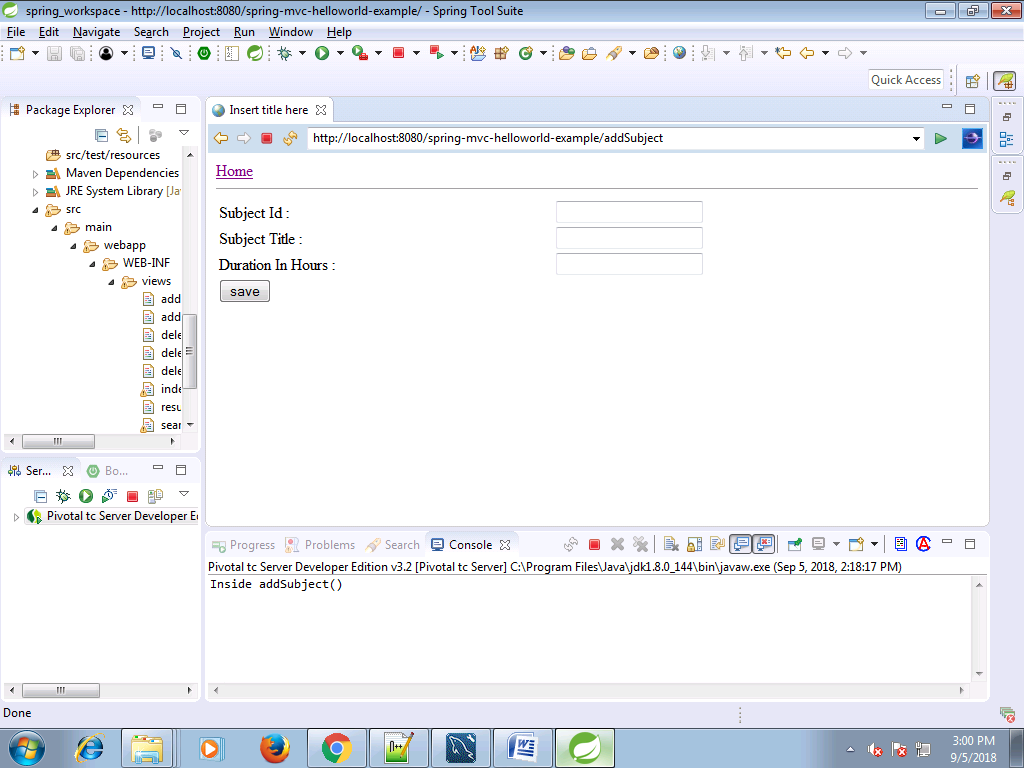
* <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*
* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>
* <modelVersion>4.0.0</modelVersion>
* <groupId>com.boraji.tutorial.spring</groupId>
* <artifactId>spring-mvc-helloworld-example</artifactId>
* <version>0.0.1-SNAPSHOT</version>
* <packaging>war</packaging>
* <properties>
* <failOnMissingWebXml>false</failOnMissingWebXml>
* </properties>
* <dependencies>
* <!-- Spring MVC Dependency -->
* <dependency>
* <groupId>org.springframework</groupId>
* <artifactId>spring-webmvc</artifactId>
* <version>4.3.7.RELEASE</version>
* </dependency>
* <!-- JSTL Dependency-->
* <dependency>
* <groupId>javax.servlet.jsp.jstl</groupId>
* <artifactId>javax.servlet.jsp.jstl-api</artifactId>
* <version>1.2.1</version>
* </dependency>
* <dependency>
* <groupId>taglibs</groupId>
* <artifactId>standard</artifactId>
* <version>1.1.2</version>
* </dependency>
* <!-- Servlet Dependency -->
* <dependency>
* <groupId>javax.servlet</groupId>
* <artifactId>javax.servlet-api</artifactId>
* <version>3.1.0</version>
* <scope>provided</scope>
* </dependency>
* <!-- JSP Dependency -->
* <dependency>
* <groupId>javax.servlet.jsp</groupId>
* <artifactId>javax.servlet.jsp-api</artifactId>
* <version>2.3.1</version>
* <scope>provided</scope>
* </dependency>
* </dependencies>
* <build>
* <sourceDirectory>src/main/java</sourceDirectory>
* <plugins>
* <plugin>
* <artifactId>maven-compiler-plugin</artifactId>
* <version>3.5.1</version>
* <configuration>
* <source>1.8</source>
* <target>1.8</target>
* </configuration>
* </plugin>
* <!-- Embedded Apache Tomcat required for testing war -->
* <plugin>
* <groupId>org.apache.tomcat.maven</groupId>
* <artifactId>tomcat7-maven-plugin</artifactId>
* <version>2.2</version>
* <configuration>
* <path>/</path>
* </configuration>
* </plugin>
* </plugins>
* </build>
* </project>

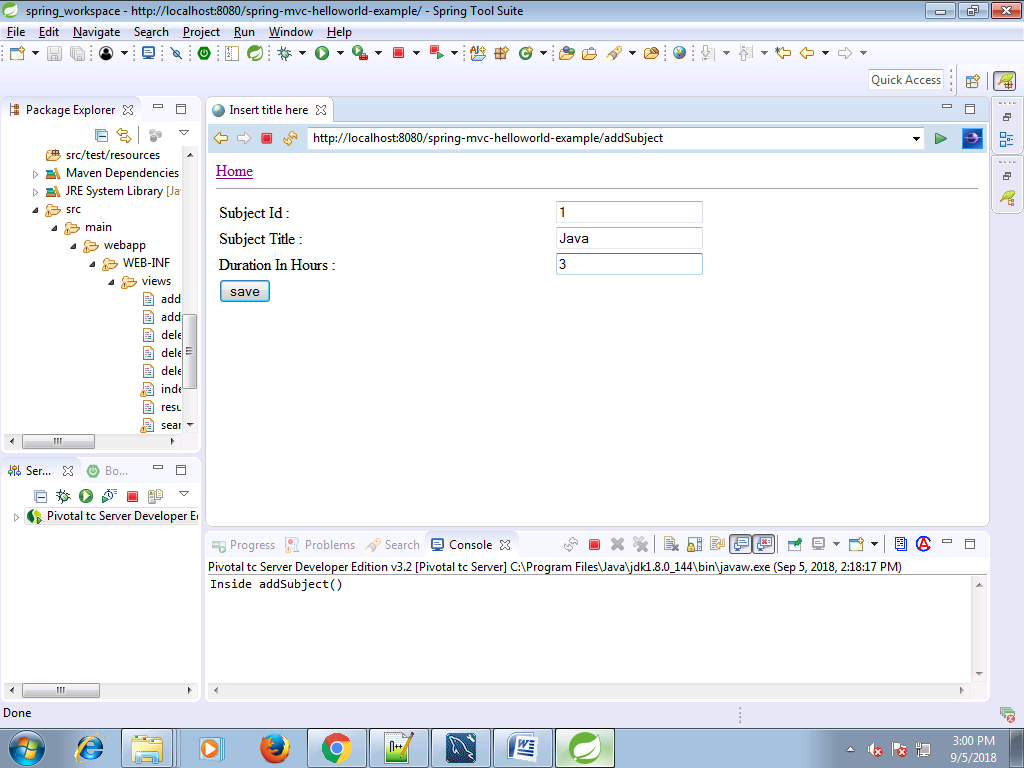
**Screenshots**

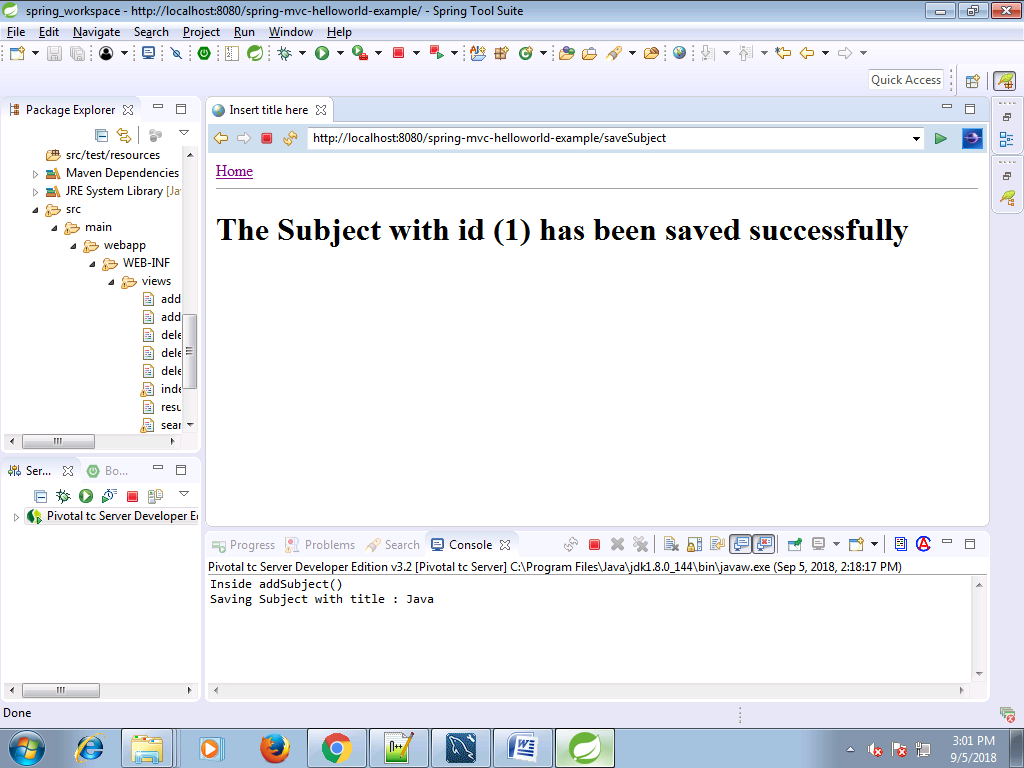
**Menus**

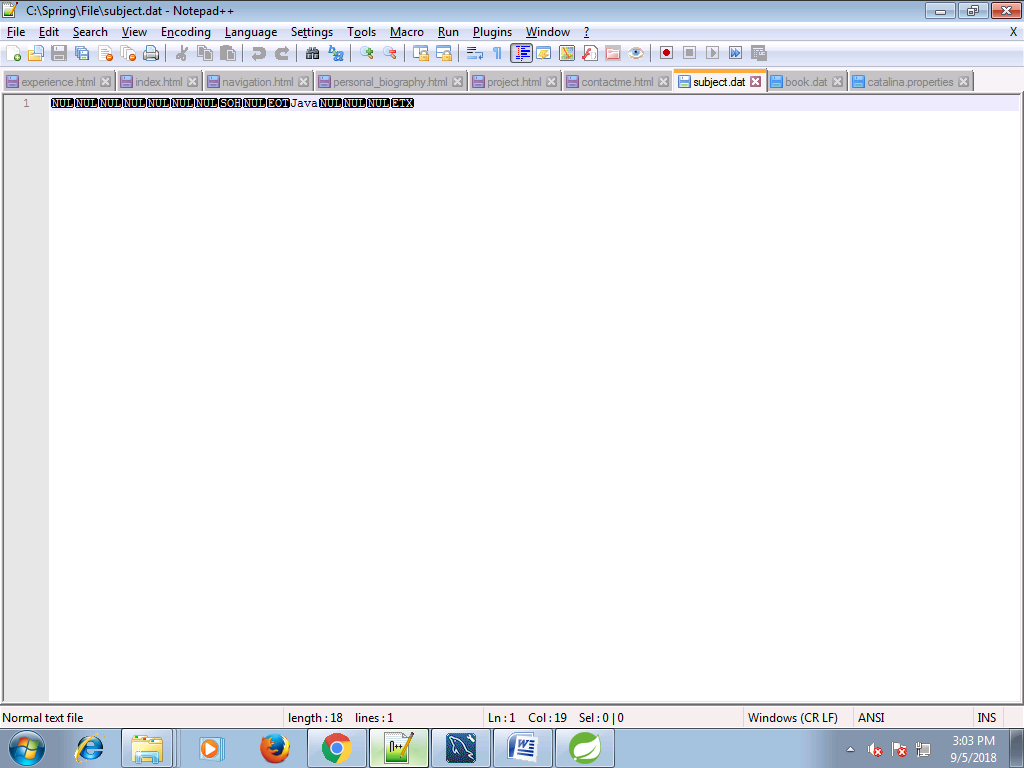
****

**Add a Subject**

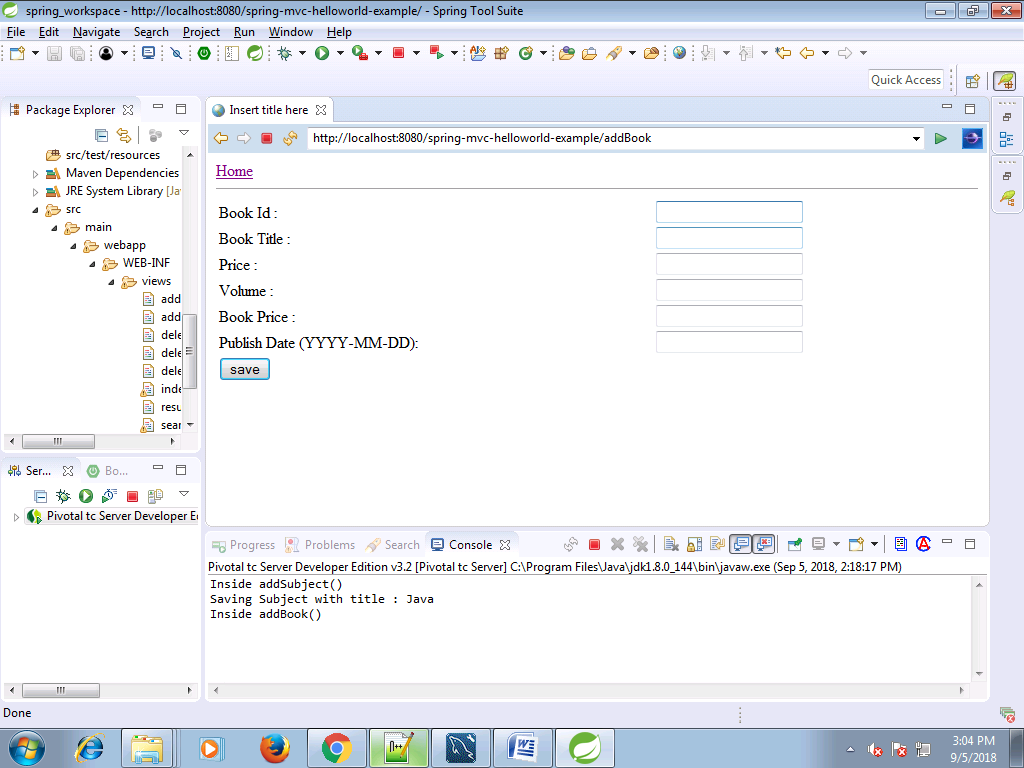
****

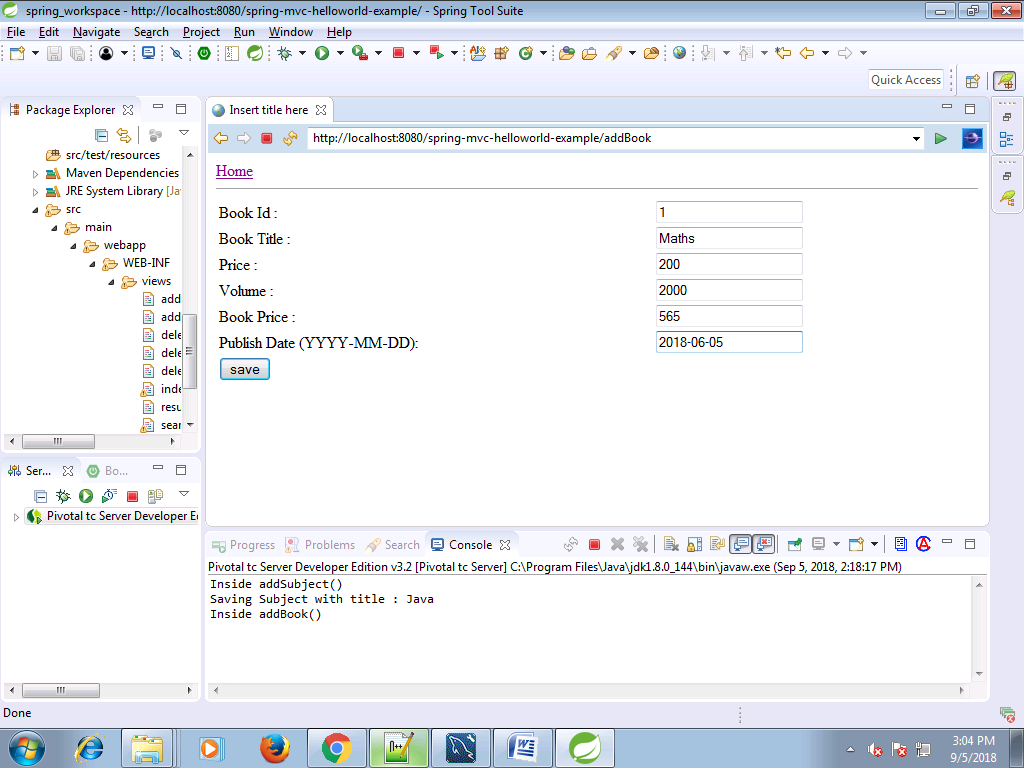
****

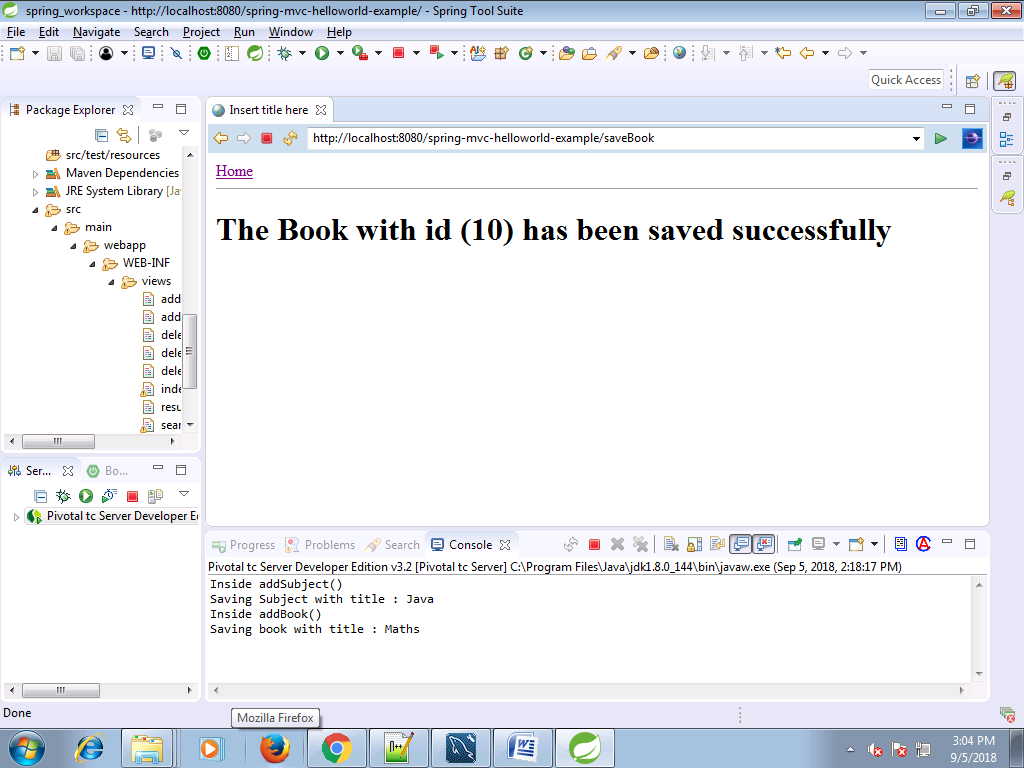
****

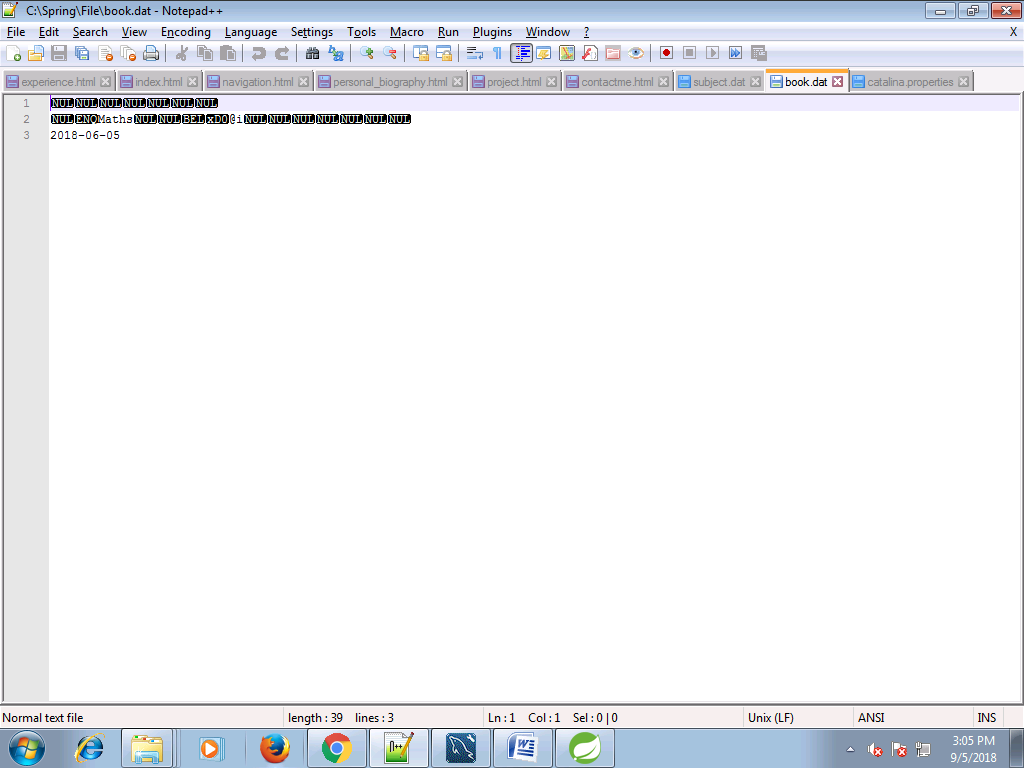
****

**Add a book**

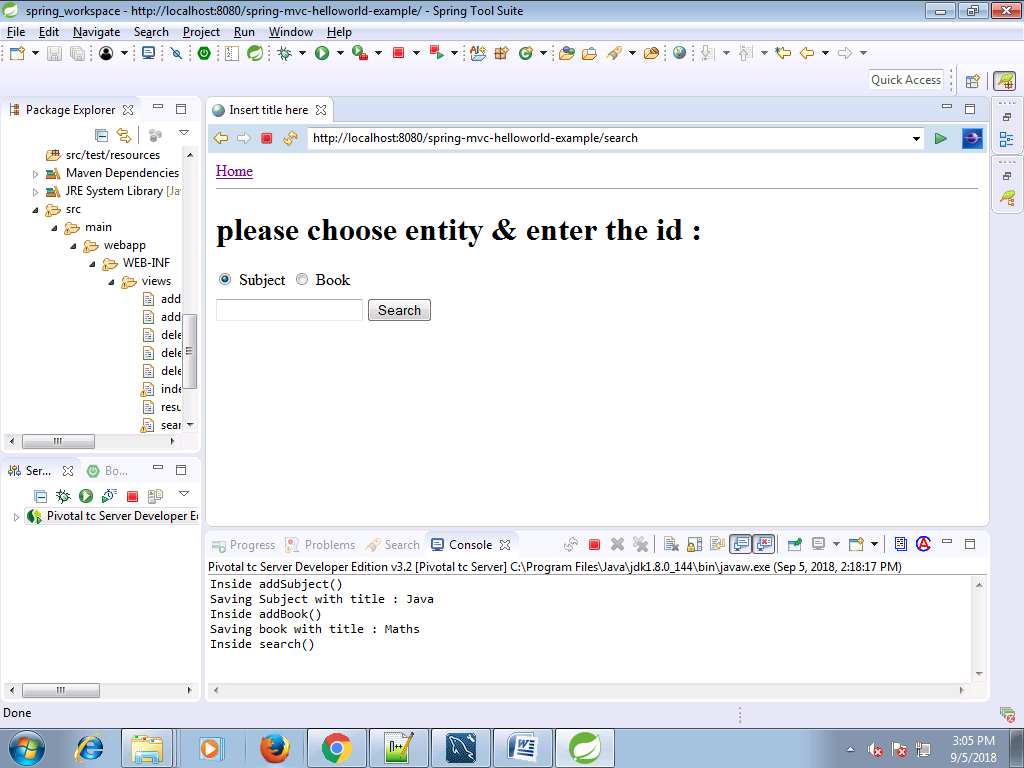
****

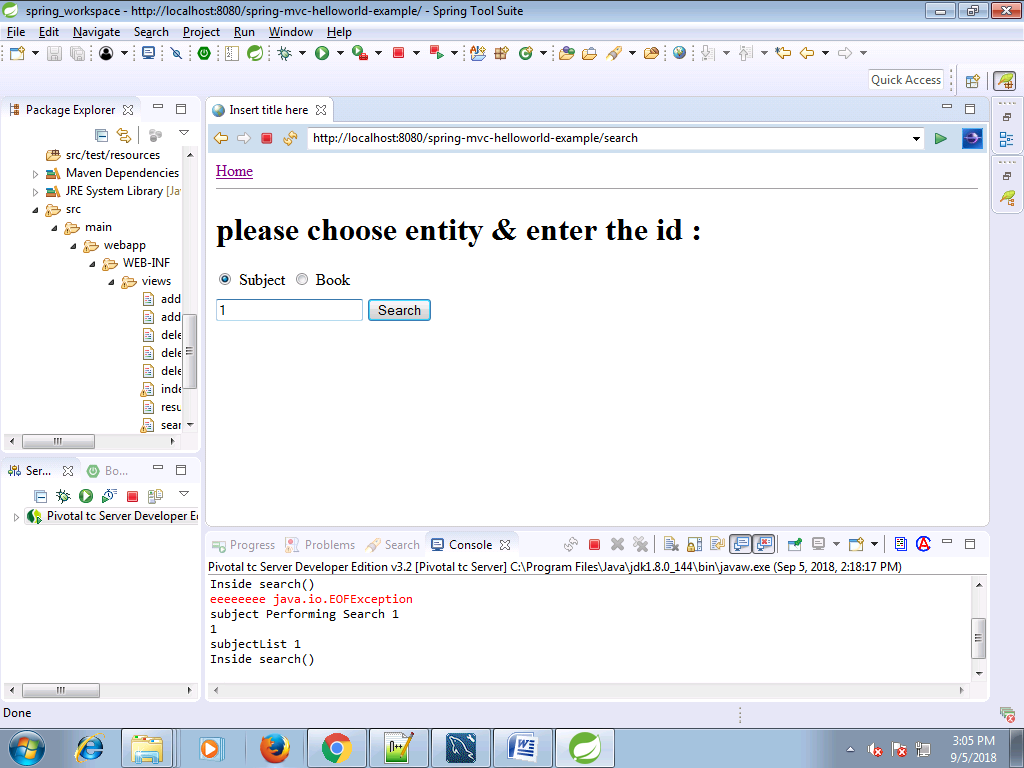
****

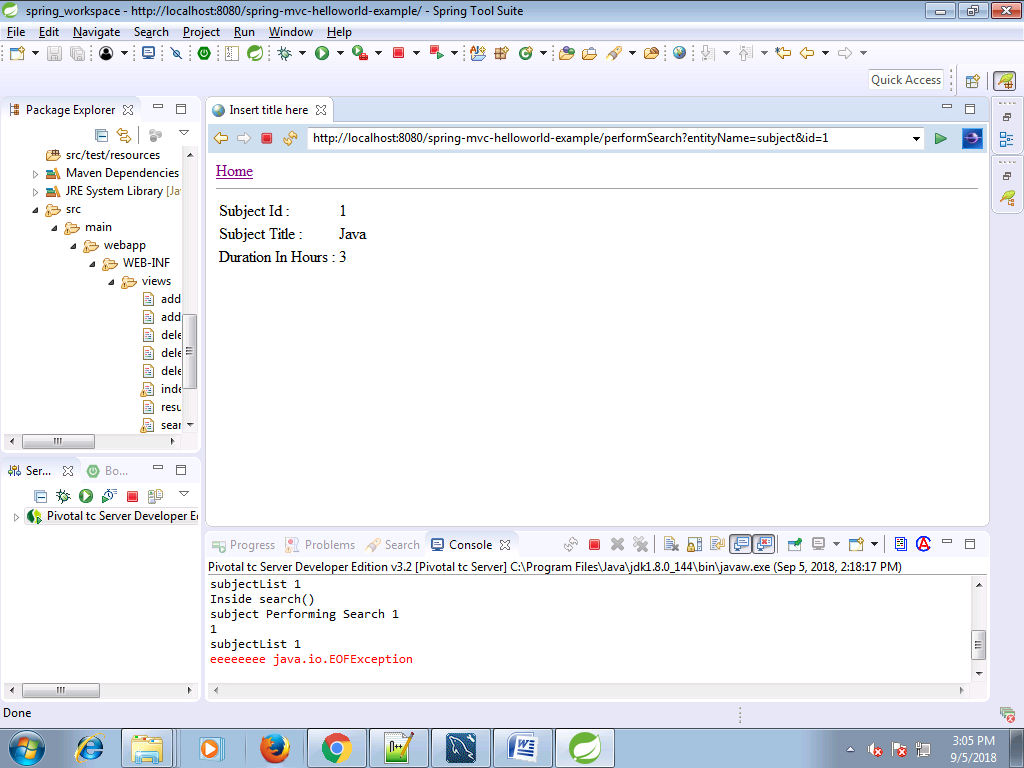
****

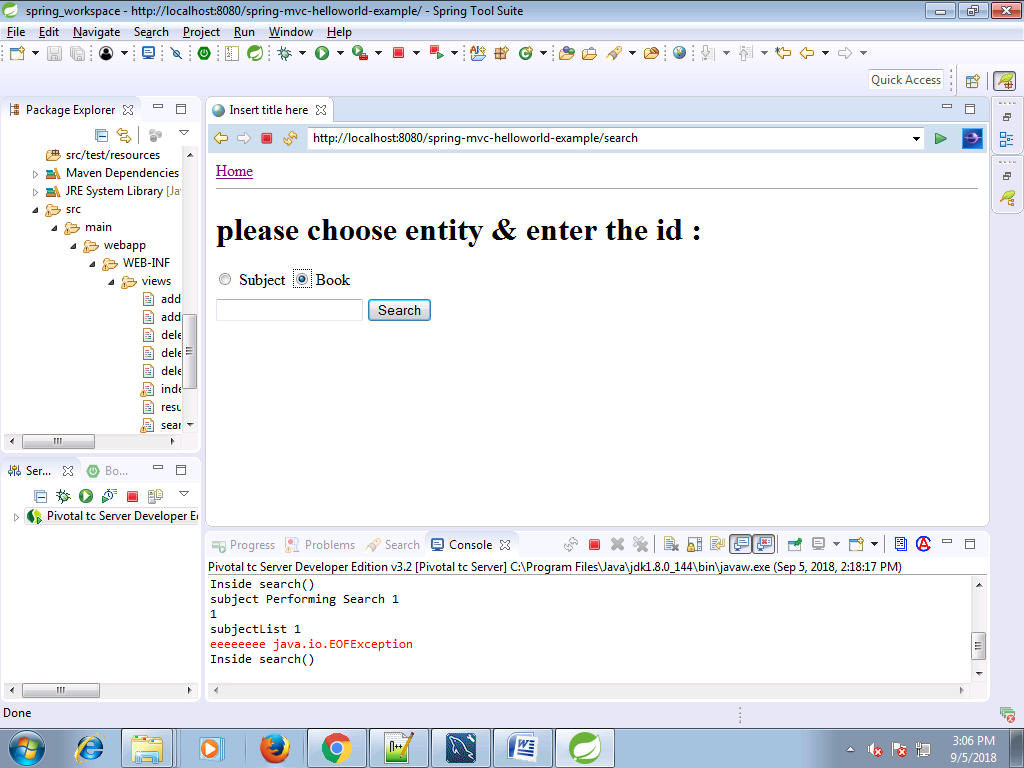
****

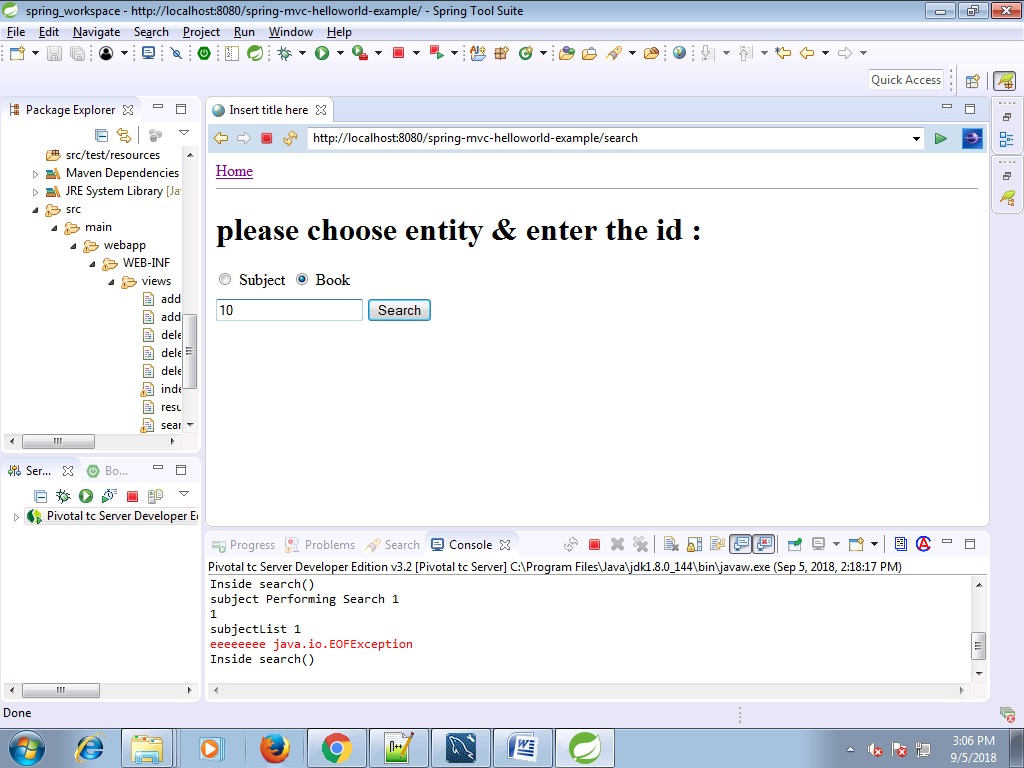
**Search book/subject**

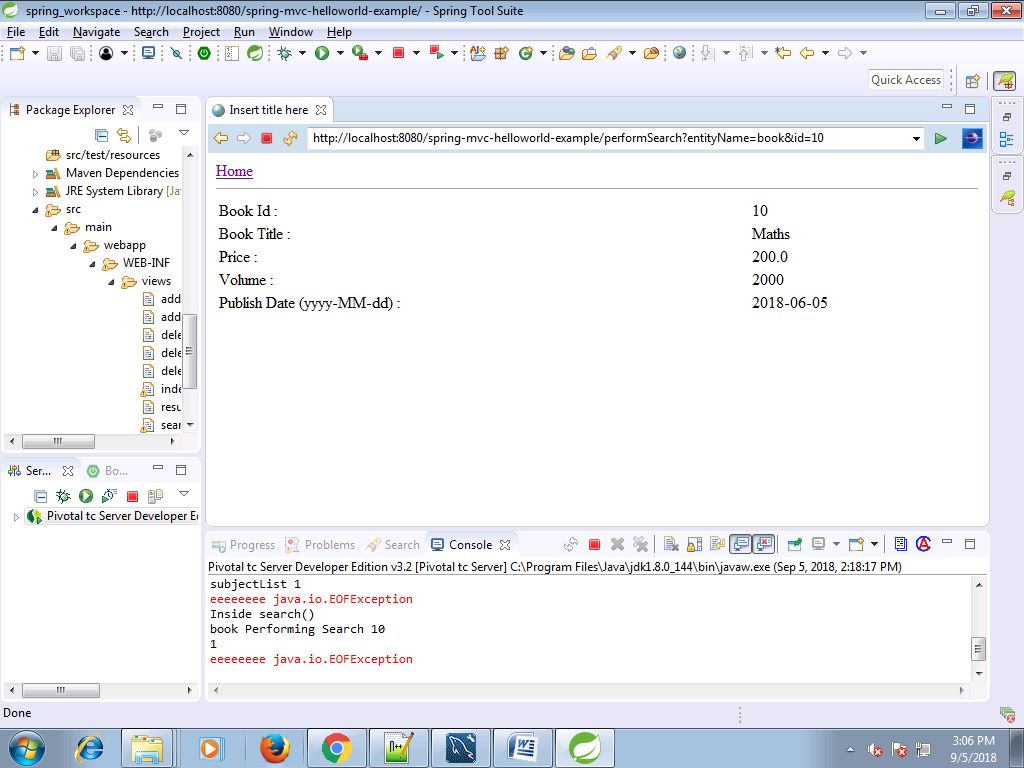
****

****

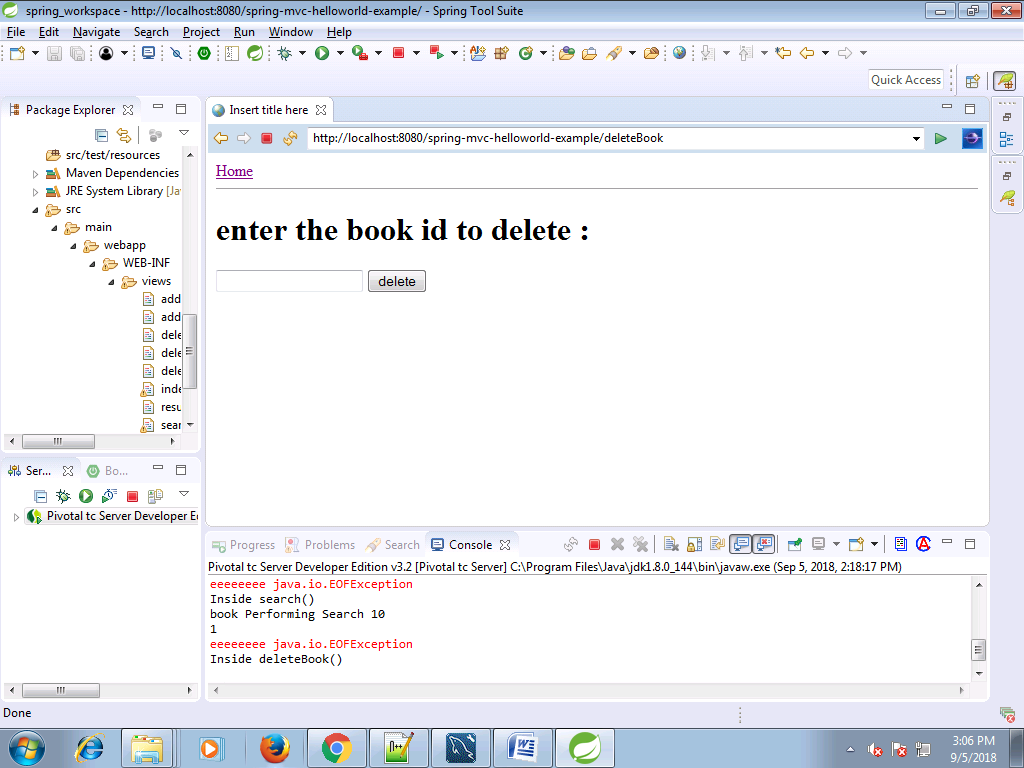
****

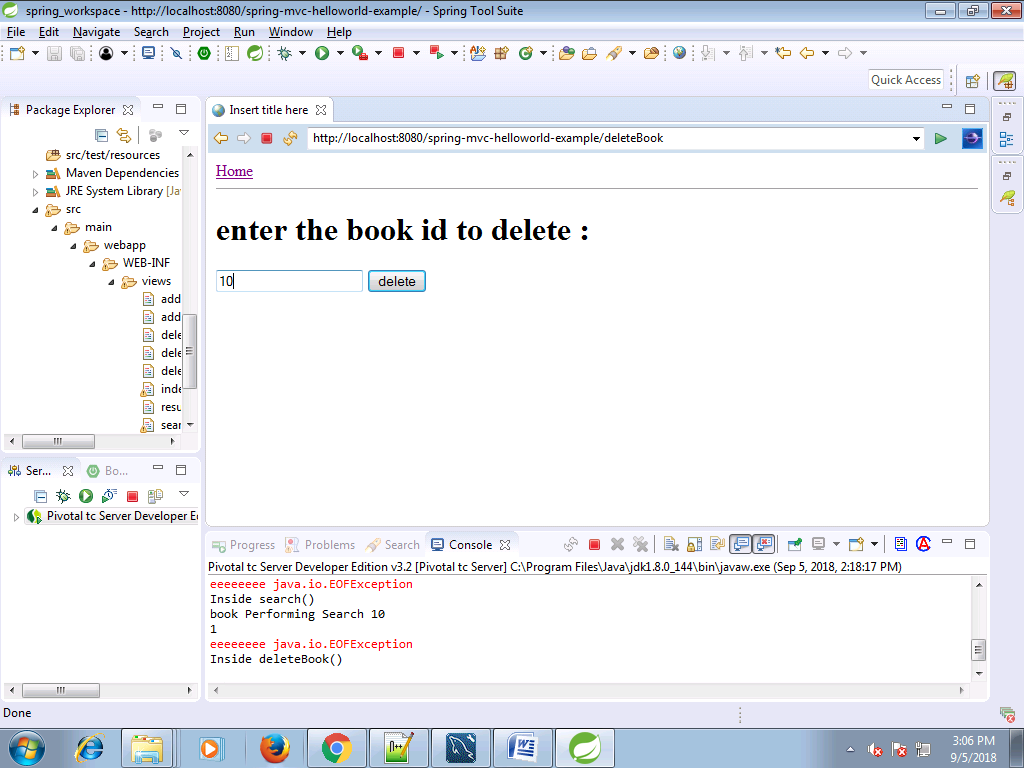
****

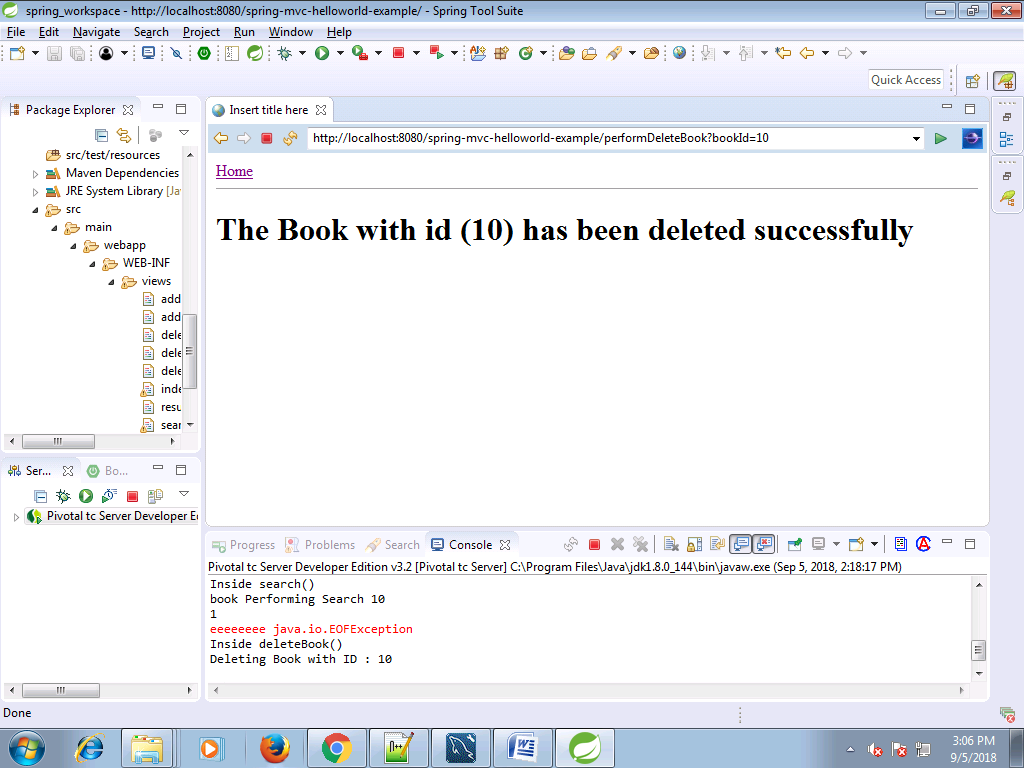
****

****

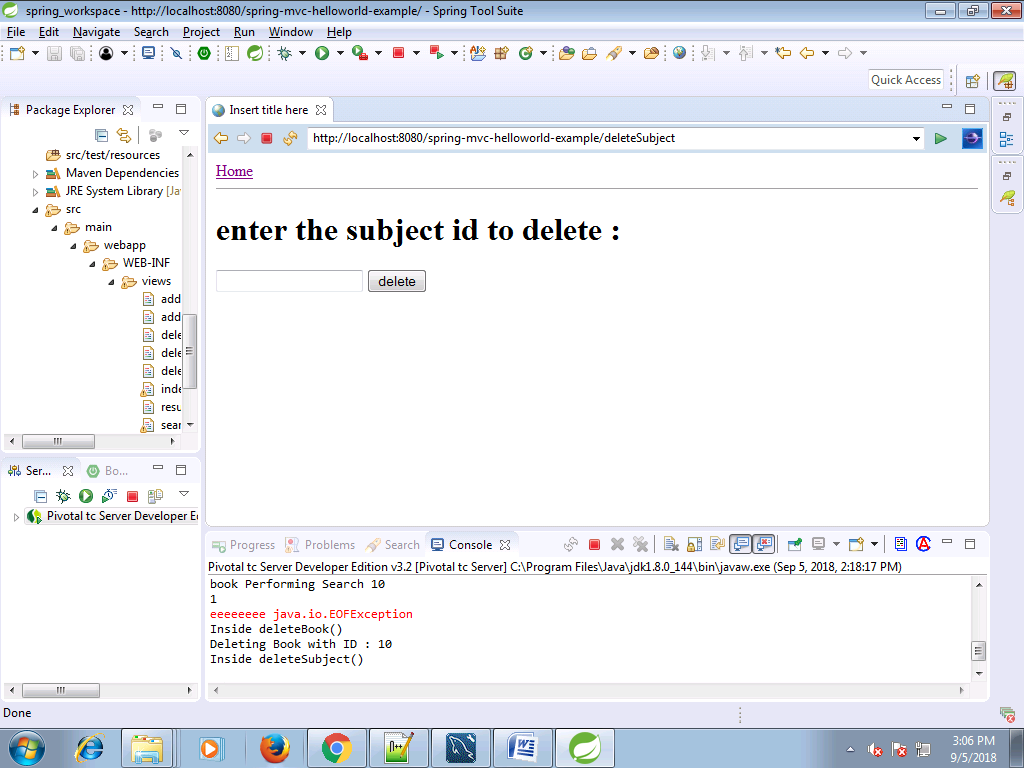
**Delete a book**

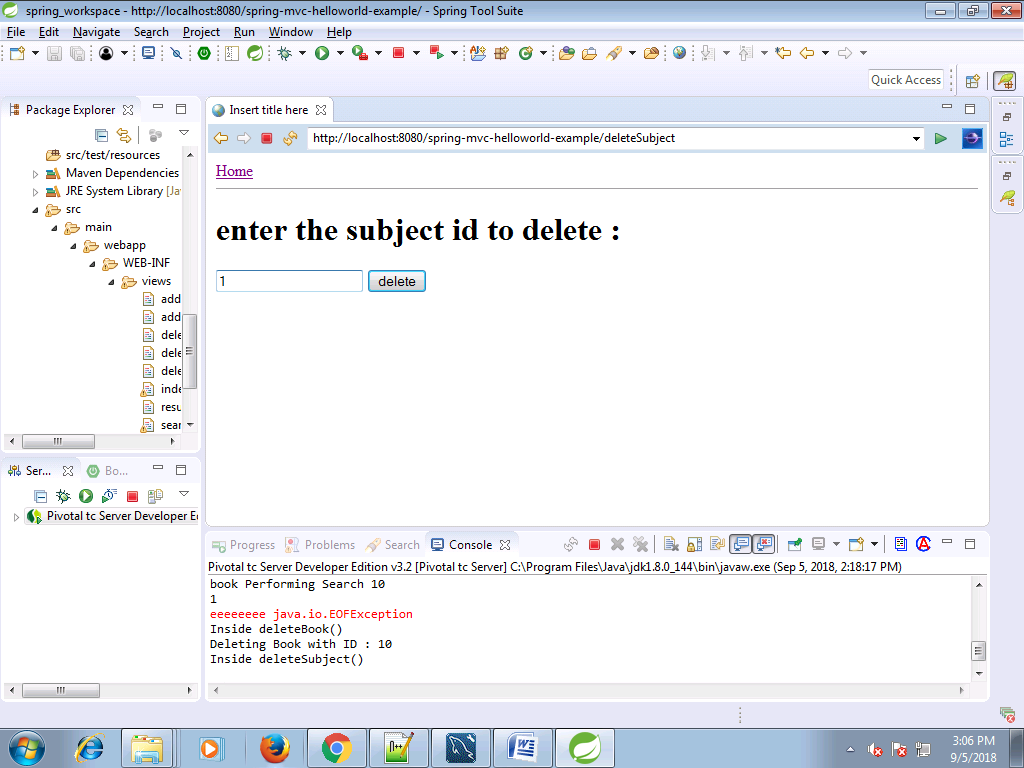
****

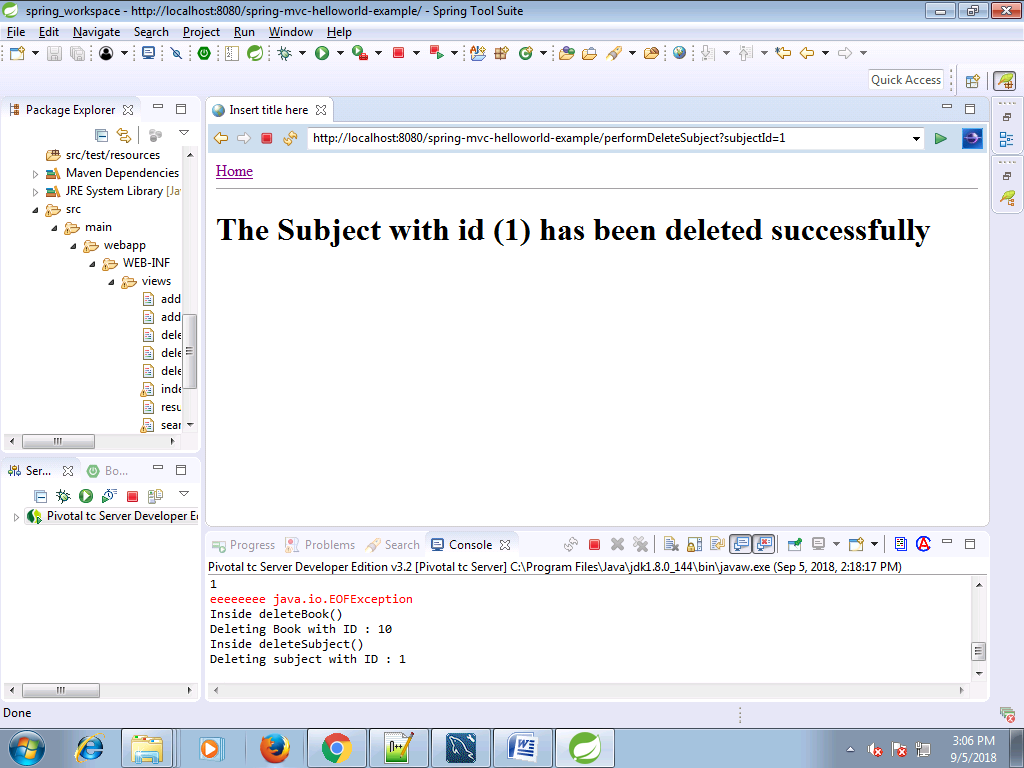
****

****

**Delete a subject**

****

****

****