**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SpringJPAMPTSolved\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\web.xml\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">

<display-name>JPASpringDataMVC</display-name>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:/jpaContext.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<servlet>

<servlet-name>EMSServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/config/servlet-config.xml</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>EMSServlet</servlet-name>

<url-pattern>\*.html</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>home.html</welcome-file>

</welcome-file-list>

</web-app>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\WEB-INF\config\servlet-config.xml\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd">

<mvc:annotation-driven />

<context:component-scan base-package="com.cg.controller" />

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"

p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />

<bean id="myBeansValidator"

class="org.springframework.validation.beanvalidation.LocalValidatorFactoryBean" />

<mvc:annotation-driven validator="myBeansValidator" />

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*project\jpaContext.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:jpa="http://www.springframework.org/schema/data/jpa"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd

http://www.springframework.org/schema/data/jpa http://www.springframework.org/schema/data/jpa/spring-jpa.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd">

<context:annotation-config />

<context:component-scan base-package="com.cg"/>

<!-- This bean is required to load ORM persistence and related annotations -->

<bean class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor" />

<!-- This bean will hold references to our database -->

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />

<property name="url" value="jdbc:oracle:thin:@localhost:1521:XE" />

<property name="username" value="hr" />

<property name="password" value="hr" />

</bean>

<!-- Spring way of entity manager factory configuration -->

<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">

<property name="dataSource" ref="dataSource"/>

<property name="packagesToScan" value="com.cg.entities"/>

<property name="persistenceProviderClass" value="org.hibernate.jpa.HibernatePersistenceProvider"/>

<property name="jpaPropertyMap">

<map>

<entry key="hibernate.dialect" value="org.hibernate.dialect.Oracle10gDialect"/>

<entry key="hibernate.hbm2ddl.auto" value="update"/>

</map>

</property>

</bean>

<!-- This bean will handle all transaction within an application -->

<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">

<property name="entityManagerFactory" ref="entityManagerFactory"/>

</bean>

<!-- This is required to handle transactions via Annotations like @Transactional -->

<tx:annotation-driven transaction-manager="transactionManager"/>

<jpa:repositories base-package="com.cg.dao"

entity-manager-factory-ref="entityManagerFactory"

transaction-manager-ref="transactionManager"/>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\WEB-INF\jsp\Home.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

<html>

<head>

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

<title>Home Page</title>

</head>

<body>

<h1>List Of Scheduled Workshop</h1>

<c:if test="${workshopList **ne** null}">

<table border="1">

<tr>

<th>Topic Name</th>

<th>Date</th>

<th>Venue</th>

<th>Do U Want To Register?</th>

</tr>

<c:forEach items="${workshopList}" var="workshop">

<tr>

<th>${workshop.workshopName}</th>

<th>${workshop.schDate}</th>

<th>${workshop.venue}</th>

<td><a href="findAvailableWS.html?wid=${workshop.workshopId}">Register</a></td>

</tr>

</c:forEach>

</table>

</c:if>

</div>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WEB-INF\jsp\WSRegPage.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

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

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

<html>

<head>

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

<title>Insert title here</title>

</head>

<body bgcolor="pink">

<h2>Workshop Registration Form: ${WID}</h2>

<c:url var="myAction" value="/RegistrationSuccess.html"/>

<form:form action="${myAction}" method="get" modelAttribute="participantModel">

<table border="1">

<tr>

<td> Name:</td>

<td><form:input path="name"/></td>

<td><form:errors path="name"/></td>

</tr>

<tr>

<td> Email Id:</td>

<td><form:input path="emailId"/></td>

<td><form:errors path="emailId"/></td>

</tr>

<tr>

<td> Mobile :</td>

<td><form:input path="mobile"/></td>

<td><form:errors path="mobile"/></td>

</tr>

<tr>

<td> Address :</td>

<td><form:input path="address"/></td>

<td><form:errors path="address"/></td>

</tr>

<tr>

<td><input type="submit" value="REGISTER"/></td><td><input type="reset" value="CLEAR"/></td>

</tr>

</table>

</form:form>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Web-INF\jsp\RegistrationSuccess.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h3>Thanks For Completing The registration For The Workshop</h3>

<h2> Your Work Shop ID Is ${message}</h2>

<a href= "home.html">View Scheduled Workshop</a>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\jsp\FailurePage.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1>Sorry Seats are not Available To attend the currently Available Workshop</h1>

<a href= "home.html">View Scheduled Workshop</a>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WorkshopController.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import javax.validation.Valid;

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.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.SessionAttributes;

import com.cg.entities.Participants;

import com.cg.entities.Workshop;

import com.cg.service.IParticipantService;

import com.cg.service.IWorkshopService;

@Controller

@SessionAttributes("WID")

public class WorkshopController

{

@Autowired

private IParticipantService participantsService;

@Autowired

private IWorkshopService workshopService;

@RequestMapping("/home")

public String getHomePage(Model model)

{

model.addAttribute("workshopList",workshopService.getAllScheduledWS());

model.addAttribute("workshop",new Workshop());

return "Home";

}

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

public String registerParticipant(

@ModelAttribute("participantModel") @Valid Participants part,

BindingResult result,

Model model,HttpServletRequest req)

{

if(result.hasErrors())

{

return "WSRegPage";

}

HttpSession ses=req.getSession();

long wwid=(long)ses.getAttribute("WID");

System.out.println(" Session Data Is..................."+wwid);

part.setWid(wwid);

Participants participant= participantsService.addParticipants(part);

System.out.println("Data Added ............................."+

participant);

//model.addAttribute("participantModel", new Participants());

model.addAttribute("message",

"Your Registration Id Is:"+participant.getRegistrationId());

return "RegistrationSuccess";

}

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

public String showRegPage(@RequestParam("wid") long workshopId,Model model)

{

long count = participantsService.getNoOfParticipantCount(workshopId);

if(count>=5)

{

return "FailurePage";

}

else

{

System.out.println(" Id is \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+workshopId);

model.addAttribute("participantModel", new Participants());

model.addAttribute("WID", workshopId);

return "WSRegPage";

}

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IParticipantService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.cg.service;

**import** com.cg.entities.Participants;

**public** **interface** IParticipantService

{

**public** Participants addParticipants(Participants part) ;

**public** **long** getNoOfParticipantCount(**long** workshopId);

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ParticipantServiceImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.service;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.cg.dao.IParticipantDAO;

import com.cg.entities.Participants;

@Service("participantsService")

@Transactional //This annotation will make automatic transaction management

public class ParticipantsServiceImpl implements IParticipantService

{

@Autowired

private IParticipantDAO participantDao;

@Override

public Participants addParticipants(Participants part) {

// TODO Auto-generated method stub

return participantDao.addParticipants(part);

}

@Override

public long getNoOfParticipantCount(long workshopId)

{

// TODO Auto-generated method stub

return participantDao.getNoOfParticipantCount(workshopId);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IWorkshopService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.service;

import java.util.List;

import com.cg.entities.Workshop;

public interface IWorkshopService {

public List<Workshop>getAllScheduledWS();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WorkshopServiceImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.service;

import java.util.List;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.cg.dao.IParticipantDAO;

import com.cg.dao.IWorkshopDao;

import com.cg.entities.Workshop;

@Service("workshopService")

@Transactional //This annotation will make automatic transaction management

public class WorkshopServiceImpl implements IWorkshopService

{

@Autowired

private IWorkshopDao workshopDao;

@Override

public List<Workshop> getAllScheduledWS() {

// TODO Auto-generated method stub

return workshopDao.getAllScheduledWS();

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Participant.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.entities;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.Date;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

import javax.validation.constraints.Pattern;

import org.hibernate.validator.constraints.Email;

import org.hibernate.validator.constraints.NotEmpty;

@Entity

@Table(name="participants")

@SequenceGenerator(name="registrationid", initialValue=1,

allocationSize=100)

public class Participants implements Serializable

{

@Id

@GeneratedValue(strategy=GenerationType.SEQUENCE,

generator="registrationid")

@Column(name="registrationid ")

private Long registrationId ;

@NotEmpty(message=" Mandatory Name")

@Pattern(regexp="[A-Za-z]+",

message="Only Characters Allowed.")

@Column(name="name")

private String name;

@NotEmpty(message=" Mandatory Email Address")

@Email(message=" Plaese enter valid email address")

@Column(name="emailId")

private String emailId;

@NotEmpty(message=" Mandatory Mobile")

@Pattern(regexp="[0-9]{10}",

message="10 Digit Mobile No.")

@Column(name="mobile")

private String mobile;

@NotEmpty(message=" Mandatory Address")

@Pattern(regexp="[A-Za-z]+",

message="Only Characters Allowed.")

@Column(name="address")

private String address;

@Temporal(TemporalType.DATE)

@Column(name="registrationDate")

private Date registrationDate;

@Column(name="workshopid")

private long wid;

public long getWid() { return wid; }

public void setWid(long wid) { this.wid = wid; }

public Long getRegistrationId() { return registrationId; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getEmailId() { return emailId; }

public void setEmailId(String emailId) { this.emailId = emailId; }

public String getMobile() { return mobile; }

public void setMobile(String mobile) { this.mobile = mobile; }

public String getAddress() { return address; }

public void setAddress(String address) { this.address = address; }

public Date getRegistrationDate() { return registrationDate; }

public void setRegistrationDate(Date registrationDate) { this.registrationDate = registrationDate; }

public void setRegistrationId(Long registrationId) { this.registrationId = registrationId; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Workshop.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.entities;

import java.io.Serializable;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

@Entity

@Table(name="workshop")

public class Workshop implements Serializable {

public Long getWorkshopId() { return workshopId; }

public void setWorkshopId(Long workshopId) { this.workshopId = workshopId; }

public String getWorkshopName() { return workshopName; }

public void setWorkshopName(String workshopName) { this.workshopName = workshopName; }

public String getVenue() { return venue; }

public void setVenue(String venue) { this.venue = venue; }

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name="workshopId")

private Long workshopId;

@Column(name="name")

private String workshopName;

@Temporal(TemporalType.DATE)

@Column(name="Scheduledate")

private Date schDate;

public Date getSchDate() { return schDate; }

public void setSchDate(Date schDate) { this.schDate = schDate; }

@Column(name="venue")

private String venue;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IWorkshopDao.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.dao;

import java.util.List;

import com.cg.entities.Workshop;

public interface IWorkshopDao

{ public List<Workshop>getAllScheduledWS();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WorkshopDaoImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.dao;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.TypedQuery;

import org.springframework.stereotype.Repository;

import com.cg.entities.Workshop;

@Repository("workshopDao")

public class WorkshopDaoImpl implements IWorkshopDao

{

@PersistenceContext

private EntityManager entityManager;

@Override

public List<Workshop> getAllScheduledWS()

{

TypedQuery<Workshop> query = entityManager.createQuery("SELECT w FROM Workshop w", Workshop.class);

List<Workshop> wsL=query.getResultList();

return wsL;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IParticipantDao.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.cg.dao;

**import** com.cg.entities.Participants;

**public** **interface** IParticipantDAO

{

**public** **abstract** Participants addParticipants(Participants part);

**public** **long** getNoOfParticipantCount(**long** workshopId);

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ParticipantDaoImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.dao;

import java.util.Date;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.TypedQuery;

import org.springframework.stereotype.Repository;

import com.cg.entities.Participants;

@Repository("participantDao")

public class ParticipantDaoImpl implements IParticipantDAO

{

@PersistenceContext

private EntityManager entityManager;

@Override

public Participants addParticipants(Participants part)

{

part.setRegistrationDate(new Date());

entityManager.persist(part);

entityManager.flush();

return part;

}

@Override

public long getNoOfParticipantCount(long workshopId)

{

/\*String qry="SELECT COUNT(part) FROM Participants part GROUP BY "+

" part.workshop.workshopId HAVING "+

" part.workshop.workshopId IN (:wid ) ";\*/

String qry="SELECT COUNT(part) FROM Participants part GROUP BY "+

" part.wid HAVING "+

" part.wid IN (:woid ) ";

Long woId=(long)workshopId;

TypedQuery<Long> tq=entityManager.createQuery(qry,Long.class);

tq.setParameter("woid", woId);

Long count=0L;

try

{

count=tq.getSingleResult();

System.out.println(" Count Is : "+count.longValue());

}

catch(Exception e)

{

return 0L;

}

return count;

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SpringJPADataMPTSolved\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\web.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">

<display-name>JPASpringDataMVC</display-name>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:/jpaContext.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<servlet>

<servlet-name>EMSServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/config/servlet-config.xml</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>EMSServlet</servlet-name>

<url-pattern>\*.html</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>home.html</welcome-file>

</welcome-file-list>

</web-app>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\config\servlet-config.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context=<http://www.springframework.org/schema/context>

xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd">

<mvc:annotation-driven />

<context:component-scan base-package="com.cg" />

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"

p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />

<bean id="myBeansValidator"

class="org.springframework.validation.beanvalidation.LocalValidatorFactoryBean" />

<mvc:annotation-driven validator="myBeansValidator" />

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*project\resources\jpaContext.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:jpa="http://www.springframework.org/schema/data/jpa"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd

http://www.springframework.org/schema/data/jpa http://www.springframework.org/schema/data/jpa/spring-jpa.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd">

<context:annotation-config />

<context:component-scan base-package="com.cg"/>

<!-- This bean is required to load ORM persistence and related annotations -->

<bean class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor" />

<!-- This bean will hold references to our database -->

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />

<property name="url" value="jdbc:oracle:thin:@localhost:1521:XE" />

<property name="username" value="hr" />

<property name="password" value="hr" />

</bean>

<!-- Spring way of entity manager factory configuration -->

<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">

<property name="dataSource" ref="dataSource"/>

<property name="packagesToScan" value="com.cg.entities"/>

<property name="persistenceProviderClass" value="org.hibernate.jpa.HibernatePersistenceProvider"/>

<property name="jpaPropertyMap">

<map>

<entry key="hibernate.dialect" value="org.hibernate.dialect.Oracle10gDialect"/>

<entry key="hibernate.hbm2ddl.auto" value="update"/>

</map>

</property>

</bean>

<!-- This bean will handle all transaction within an application -->

<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">

<property name="entityManagerFactory" ref="entityManagerFactory"/>

</bean>

<!-- This is required to handle transactions via Annotations like @Transactional -->

<tx:annotation-driven transaction-manager="transactionManager"/>

<jpa:repositories base-package="com.cg.dao"

entity-manager-factory-ref="entityManagerFactory"

transaction-manager-ref="transactionManager"/>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\WEB-INF\jsp\Home.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

<html>

<head>

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

<title>Home Page</title>

</head>

<body>

<h1>List Of Scheduled Workshop</h1>

<c:if test="${workshopList **ne** null}">

<table border="1">

<tr>

<th>Topic Name</th>

<th>Date</th>

<th>Venue</th>

<th>Do U Want To Register?</th>

</tr>

<c:forEach items="${workshopList}" var="workshop">

<tr>

<th>${workshop.workshopName}</th>

<th>${workshop.schDate}</th>

<th>${workshop.venue}</th>

<td><a href="findAvailableWS.html?wid=${workshop.workshopId}">Register</a></td>

</tr>

</c:forEach>

</table>

</c:if>

</div>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\jsp\WSRegPage.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

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

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

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

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

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

<html>

<head>

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

<title>Insert title here</title>

</head>

<body bgcolor="pink">

<h2>Workshop Registration Form: ${WID}</h2>

<c:url var="myAction" value="/RegistrationSuccess.html"/>

<form:form action="${myAction}" method="get" modelAttribute="participantModel">

<table border="1">

<tr>

<td> Name:</td>

<td><form:input path="name"/></td>

<td><form:errors path="name"/></td>

</tr>

<tr>

<td> Email Id:</td>

<td><form:input path="emailId"/></td>

<td><form:errors path="emailId"/></td>

</tr>

<tr>

<td> Mobile :</td>

<td><form:input path="mobile"/></td>

<td><form:errors path="mobile"/></td>

</tr>

<tr>

<td> Address :</td>

<td><form:input path="address"/></td>

<td><form:errors path="address"/></td>

</tr>

<tr>

<td><input type="submit" value="REGISTER"/></td><td><input type="reset" value="CLEAR"/></td>

</tr>

</table>

</form:form>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\jsp\RegistrationSuccess.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h3>Thanks For Completing The registration For The Workshop</h3>

<h2> Your Work Shop ID Is ${message}</h2>

<a href= "home.html">View Scheduled Workshop</a>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***WEB-INF\jsps\FailurePage.jsp\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\***

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

pageEncoding="ISO-8859-1"%>

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

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

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

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1>Sorry Seats are not Available To attend the currently Available Workshop</h1>

<a href= "home.html">View Scheduled Workshop</a>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Participant.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.entities;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.Date;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

import javax.validation.constraints.Pattern;

import org.hibernate.validator.constraints.Email;

import org.hibernate.validator.constraints.NotEmpty;

@Entity

@Table(name="participants")

@SequenceGenerator(name="registrationid", initialValue=1, allocationSize=100)

public class Participants implements Serializable

{

@Id

@GeneratedValue(strategy=GenerationType.SEQUENCE,generator="registrationid")

@Column(name="registrationid ")

private Long registrationId ;

@NotEmpty(message=" Mandatory Name")

@Pattern(regexp="[A-Za-z]+",

message="Only Characters Allowed.")

@Column(name="name")

private String name;

@NotEmpty(message=" Mandatory Email Address")

@Email(message=" Plaese enter valid email address")

@Column(name="emailId")

private String emailId;

@NotEmpty(message=" Mandatory Mobile")

@Pattern(regexp="[0-9]{10}",

message="10 Digit Mobile No.")

@Column(name="mobile")

private String mobile;

@NotEmpty(message=" Mandatory Address")

@Pattern(regexp="[A-Za-z]+",

message="Only Characters Allowed.")

@Column(name="address")

private String address;

@Temporal(TemporalType.DATE)

@Column(name="registrationDate")

private Date registrationDate;

@Column(name="workshopid")

private long wid;

public long getWid() {return wid;}

public void setWid(long wid) { this.wid = wid; }

public Long getRegistrationId() { return registrationId; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getEmailId() { return emailId; }

public void setEmailId(String emailId) { this.emailId = emailId; }

public String getMobile() { return mobile; }

public void setMobile(String mobile) { this.mobile = mobile; }

public String getAddress() { return address; }

public void setAddress(String address) { this.address = address; }

public Date getRegistrationDate() { return registrationDate; }

public void setRegistrationDate(Date registrationDate) { this.registrationDate = registrationDate; }

public void setRegistrationId(Long registrationId) { this.registrationId = registrationId; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Workshop.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.entities;

import java.io.Serializable;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

@Entity

@Table(name="workshop")

public class Workshop implements Serializable {

public Long getWorkshopId() { return workshopId; }

public void setWorkshopId(Long workshopId) { this.workshopId = workshopId; }

public String getWorkshopName() { return workshopName; }

public void setWorkshopName(String workshopName) { this.workshopName = workshopName; }

public String getVenue() { return venue; }

public void setVenue(String venue) { this.venue = venue; }

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name="workshopId")

private Long workshopId;

@Column(name="name")

private String workshopName;

@Temporal(TemporalType.DATE)

@Column(name="Scheduledate")

private Date schDate;

public Date getSchDate() { return schDate; }

public void setSchDate(Date schDate) { this.schDate = schDate; }

@Column(name="venue")

private String venue;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IParticipantService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.cg.service;

**import** com.cg.entities.Participants;

**public** **interface** IParticipantService

{

**public** Participants addParticipants(Participants part) ;

**public** **long** getNoOfParticipantCount(**long** workshopId);

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ParticipantServiceImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.service;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.cg.dao.ParticipantRepository;

import com.cg.entities.Participants;

@Service("participantsService")

@Transactional //This annotation will make automatic transaction management

public class ParticipantsServiceImpl implements IParticipantService

{

@Autowired

private ParticipantRepository participantDao;

@Override

public Participants addParticipants(Participants part) {

// TODO Auto-generated method stub

return participantDao.save(part);

}

@Override

public long getNoOfParticipantCount(long workshopId)

{

// TODO Auto-generated method stub

return participantDao.getNoOfParticipantCount(workshopId);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IWorkshop.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.service;

import java.util.List;

import com.cg.entities.Workshop;

public interface IWorkshopService {

public List<Workshop>getAllScheduledWS();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WorkshopServiceImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.service;

import java.util.List;

import com.cg.entities.Workshop;

public interface IWorkshopService {

public List<Workshop>getAllScheduledWS();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ParticipantRepository.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.dao;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import com.cg.entities.Participants;

@Repository("participantDao")

public interface ParticipantRepository extends JpaRepository<Participants, Long> {

@Query("SELECT COUNT(part) FROM Participants part GROUP BY part.wid HAVING part.wid IN (:woid )")

public long getNoOfParticipantCount(@Param("woid") long workshopId) ;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WorkshopRepository.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.dao;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import com.cg.entities.Workshop;

@Repository("workshopDao")

public interface WorkshopRepository extends JpaRepository<Workshop, Long> {

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WorkshopController.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import javax.validation.Valid;

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.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.SessionAttributes;

import com.cg.entities.Participants;

import com.cg.entities.Workshop;

import com.cg.service.IParticipantService;

import com.cg.service.IWorkshopService;

@Controller

@SessionAttributes("WID")

public class WorkshopController

{

@Autowired

private IParticipantService participantsService;

@Autowired

private IWorkshopService workshopService;

@RequestMapping("/home")

public String getHomePage(Model model)

{

model.addAttribute("workshopList",workshopService.getAllScheduledWS());

model.addAttribute("workshop",new Workshop());

return "Home";

}

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

public String registerParticipant(

@ModelAttribute("participantModel") @Valid Participants part,

BindingResult result,

Model model,HttpServletRequest req)

{

if(result.hasErrors())

{

return "WSRegPage";

}

HttpSession ses=req.getSession();

long wwid=(long)ses.getAttribute("WID");

System.out.println(" Session Data Is..................."+wwid);

part.setWid(wwid);

Participants participant= participantsService.addParticipants(part);

System.out.println("Data Added ............................."+

participant);

//model.addAttribute("participantModel", new Participants());

model.addAttribute("message",

"Your Registration Id Is:"+participant.getRegistrationId());

return "RegistrationSuccess";

}

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

public String showRegPage(@RequestParam("wid") long workshopId,Model model)

{

long count = participantsService.getNoOfParticipantCount(workshopId);

if(count>=5)

{

return "FailurePage";

}

else

{

System.out.println(" Id is \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+workshopId);

model.addAttribute("participantModel", new Participants());

model.addAttribute("WID", workshopId);

return "WSRegPage";

}

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Spring With Angular\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SpringWithAngular\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\web.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">

<display-name>SpringWithAngular</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/rest/\*</url-pattern>

</servlet-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

</web-app>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WEB-INF\dispatcher-servlet.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd">

<!-- To enable Spring Components -->

<context:component-scan base-package="com.cg.springwithangular" />

<!-- Mapping for Spring ViewResolver -->

<mvc:annotation-driven/>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Keep\*\*\*\*\*WebContent\angular.js\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*Webcontent\myjson.json\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*\*

\* \_refreshPageData() will call automatically when page loaded

\* Calling \_refreshPageData() after every add & Remove of data

\*/

**var** myWebservice=angular.module("myApp",[]);

myWebservice.controller("ServiceController",

**function**($scope, $http) {

\_refreshPageData();

//Check url according to your Application name & PORT

**function** \_refreshPageData(){

$http({

method: 'GET',

url: 'http://localhost:8083/SpringWithAngular/rest/countries.json'

}).success(**function**(data)

{

$scope.posts = data; // response data

});

}

$scope.form = {

countryId : 1,

countryName : "",

population : ""

};

$scope.add=**function**(){

//Check url according to your Application name & PORT

$http({

method:'POST',

url:'http://localhost:8083/SpringWithAngular/rest/countries/create/'+

$scope.form.countryId+'/'+$scope.form.countryName+'/'+$scope.form.population

}).success(**function**(data)

{

alert("DATA ADDED");

\_refreshPageData();

});

}

$scope.remove=**function**(data){

//Check url according to your Application name & PORT

$http({

method:'DELETE',

url:'http://localhost:8083/SpringWithAngular/rest//countries/delete/'+data

}).success(**function**(data)

{

alert('Data Deleted')

\_refreshPageData();

});

}

});

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Webcontent\DemoService.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<!DOCTYPE html>

<html ng-app="myApp">

<head>

<meta charset="ISO-8859-1">

<title>Index Page</title>

<script type="text/javascript" src="angular.js"></script>

<script type="text/javascript" src="myjson.js"></script>

</head>

<body ng-controller="ServiceController">

<h1>COUNTRY LIST</h1>

<!-- list of items with a button to remove that item -->

<table border="1">

<thead>

<tr>

<th>Country Id</th>

<th>Country Name</th>

<th>Country Population</th>

<th>Action</th>

</tr>

</thead>

<tr ng-repeat="c in posts">

<td>{{c.countryId}}</td>

<td>{{c.countryName}}</td>

<td>{{c.population}}</td>

<td><button ng-click="remove($index)">Remove</button></td>

</tr>

</table>

<!-- input and button it add a new item -->

<h1>ADD Country</h1>

<table>

<tr>

<td> Country Id </td>

<td>

<input type="text" ng-model="form.countryId">

</td>

</tr>

<tr>

<td> Country Name

</td>

<td> <input type="text" ng-model="form.countryName"></td>

</tr>

<tr>

<td> Country Population</td>

<td> <input type="text" ng-model="form.population"></td>

</tr>

<tr><td><button type="submit" ng-click="add()">Add</button>

</td>

</table>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Country.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.cg.springwithangular.beans;

**public** **class** Country {

**private** String countryId;

**private** String countryName;

**private** String population;

**public** String getCountryId() {

**return** countryId;

}

**public** **void** setCountryId(String countryId) { **this**.countryId = countryId; }

**public** String getCountryName() { **return** countryName; }

**public** **void** setCountryName(String countryName) { **this**.countryName = countryName; }

**public** String getPopulation() { **return** population; }

**public** **void** setPopulation(String population) { **this**.population = population; }

@Override

**public** String toString() {

**return** "Country [countryId=" + countryId + ", countryName="

+ countryName + ", population=" + population + "]";

}

**public** Country(){ }

**public** Country(String countryId, String countryName, String population) {

**super**();

**this**.countryId = countryId;

**this**.countryName = countryName;

**this**.population = population;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CountryDB.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.springwithangular.staticdb;

import java.util.ArrayList;

import java.util.List;

import com.cg.springwithangular.beans.Country;

public class CountryDb

{

private static List<Country> countryList=new ArrayList<Country>();

static{

countryList.add(new Country("1001","India","3234567"));

countryList.add(new Country("1002","Pakistan","64567"));

countryList.add(new Country("1003","SriLanka","34567"));

countryList.add(new Country("1004","China","454567"));

countryList.add(new Country("1005","USA","134567"));

}

public static List<Country> getCountryList() {

return countryList;

}

public static void setCountryList(List<Country> countryList) {

CountryDb.countryList = countryList;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ICountryService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.springwithangular.service;

import java.util.List;

import com.cg.springwithangular.beans.Country;

public interface ICountryService {

public List<Country> getAllCountries();

public void addCountry(Country country);

public Country deleteCountry(int id);

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CountryServiceImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.springwithangular.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.cg.springwithangular.beans.Country;

import com.cg.springwithangular.dao.ICountryDao;

@Service

public class CountryServiceImpl implements ICountryService{

@Autowired

ICountryDao countryDao;

@Override

public List<Country> getAllCountries() {

// TODO Auto-generated method stub

return countryDao.getAllCountries();

}

@Override

public void addCountry(Country country) {

// TODO Auto-generated method stub

countryDao.addCountry(country);

}

@Override

public Country deleteCountry(int id) {

// TODO Auto-generated method stub

return countryDao.deleteCountry(id);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ICountryDao.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.springwithangular.dao;

import java.util.List;

import com.cg.springwithangular.beans.Country;;

public interface ICountryDao {

public List<Country> getAllCountries();

public void addCountry(Country country);

public Country deleteCountry(int id);

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CountryDaoImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.springwithangular.dao;

import java.util.List;

import org.springframework.stereotype.Repository;

import com.cg.springwithangular.beans.Country;

import com.cg.springwithangular.staticdb.CountryDb;

@Repository

public class CountryDaoImpl implements ICountryDao {

@Override

public List<Country> getAllCountries() {

// TODO Auto-generated method stub

return CountryDb.getCountryList();

}

@Override

public void addCountry(Country country) {

// TODO Auto-generated method stub

CountryDb.getCountryList().add(country);

}

@Override

public Country deleteCountry(int id) {

// TODO Auto-generated method stub

return CountryDb.getCountryList().remove(id); }}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CountryController.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.springwithangular.controller;

import java.util.List;

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

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

import com.cg.springwithangular.beans.Country;

import com.cg.springwithangular.service.ICountryService;

/\*\*

\*

\* @author rvikash REST Controller with different HTTP methods as

\* GET,POST,DELETE and their respective URL mappings class level

\* request mapping as "countries"

\* return type is JSON

\*/

@RestController

public class CountryController {

@Autowired

ICountryService service;

/\*\*

\* Method fetches all country details supporting HTTP GET method

\*

\* @param MODEL

\* @return String - List Country JSON

\*/

@RequestMapping(value = "/countries",

method = RequestMethod.GET,headers="Accept=application/json")

public List<Country> getAllCounties(Model model) {

return service.getAllCountries();

}

/\*\*

\* Method creates a spring form to create a country

\*

\* @param id name population

\* @return List-format JSON

\*/

@RequestMapping(value = "/countries/create/{id}/{name}/{popu}",

headers="Accept=application/json", method = RequestMethod.POST)

public List<Country> createCountry

(@PathVariable String id,@PathVariable String name,@PathVariable String popu) {

Country country=new Country();

country.setCountryId(id);

country.setCountryName(name);

country.setPopulation(popu);

service.addCountry(country);

return service.getAllCountries();

}

/\*\*

\* Method creates a spring form to delete a country \* \* @param id

\* @return List format Json

\*/

@RequestMapping(value = "/countries/delete/{id}",

headers="Accept=application/json",method = RequestMethod.DELETE)

public List<Country> deleteCountry(@PathVariable int id) {

service.deleteCountry(id);

return service.getAllCountries();

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Thank You\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***