



Create database and table in Mysql:

Mysql> Create database spring;

Mysql> use spring;

Mysql> CREATE TABLE `EMP\_TBL` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(45) NOT NULL,

`email` varchar(45) NOT NULL,

`address` varchar(45) NOT NULL,

`telephone` varchar(45) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8;

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/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.sailotech</groupId>

<artifactId>MVCSPRINGHBNET</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>MVCSPRINGHBNET Maven Webapp</name>

<url>http://maven.apache.org</url>

<!-- Specifying the Versions of Spring, Hiberante, MySQL etc -->

<properties>

<spring.version>4.1.5.RELEASE</spring.version>

<hibernate.version>4.3.8.Final</hibernate.version>

<mysql.version>5.1.10</mysql.version>

<junit-version>4.11</junit-version>

<servlet-api-version>3.1.0</servlet-api-version>

<jsp-version>2.1</jsp-version>

<jstl-version>1.2</jstl-version>

<java.version>1.7</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

<scope>test</scope>

</dependency>

<!-- Hibernate 4 dependencies -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>${hibernate.version}</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-c3p0</artifactId>

<version>${hibernate.version}</version>

</dependency>

<!--MYSQL Connector -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>${mysql.version}</version>

</dependency>

<!-- Servlet and JSP -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>${servlet-api-version}</version>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>${jsp-version}</version>

<scope>provided</scope>

</dependency>

<!-- JSTL dependency -->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>${jstl-version}</version>

</dependency>

<!-- JUnit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>${junit-version}</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>MVCSPRINGHBNET</finalName>

</build>

</project>

WEB-INF/Web.xml

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<context-param>

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

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

</context-param>

<listener>

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

</listener>

<servlet>

<servlet-name>spring</servlet-name>

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

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

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

Src/main/resources :

File : application.properties

#Database related properties

database.driver=com.mysql.jdbc.Driver

database.url=jdbc:mysql://localhost:3306/spring?characterEncoding=latin1&useConfigs=maxPerformance

database.user=root

database.password=root

#Hibernate related properties

hibernate.dialect=org.hibernate.dialect.MySQLDialect

hibernate.show\_sql=true

hibernate.format\_sql=true

hibernate.hbm2ddl.auto=update

WEB-INF/ spring-servlet.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:mvc=*"http://www.springframework.org/schema/mvc"* xmlns:context=*"http://www.springframework.org/schema/context"*

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

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd*

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

<!-- Specifying base package of the Components like Controller, Service,

DAO -->

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

<!-- Getting Database properties -->

<context:property-placeholder location=*"classpath:application.properties"* />

<mvc:annotation-driven />

<!-- Specifying the Resource location to load JS, CSS, Images etc -->

<mvc:resources mapping=*"/resources/\*\*"* location=*"/resources/"* />

<!-- View Resolver -->

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/pages/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<!-- DataSource -->

<bean class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*

id=*"dataSource"*>

<property name=*"driverClassName"* value=*"${database.driver}"*></property>

<property name=*"url"* value=*"${database.url}"*></property>

<property name=*"username"* value=*"${database.user}"*></property>

<property name=*"password"* value=*"${database.password}"*></property>

</bean>

<!-- Hibernate SessionFactory -->

<bean id=*"sessionFactory"*

class=*"org.springframework.orm.hibernate4.LocalSessionFactoryBean"*>

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

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>${hibernate.dialect}</prop>

<prop key=*"hibernate.hbm2ddl.auto"*>${hibernate.hbm2ddl.auto}</prop>

<prop key=*"hibernate.format\_sql"*>${hibernate.format\_sql}</prop>

<prop key=*"hibernate.show\_sql"*>${hibernate.show\_sql}</prop>

</props>

</property>

<property name=*"packagesToScan"* value=*"com.stech.model"*></property>

</bean>

<!-- Transaction -->

<bean id=*"transactionManager"*

class=*"org.springframework.orm.hibernate4.HibernateTransactionManager"*>

<property name=*"sessionFactory"* ref=*"sessionFactory"* />

</bean>

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

</beans>

Src/main/java

Package : com.stech.controller

**package** com.stech.controller;

**import** java.io.IOException;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**import** org.jboss.logging.Logger;

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

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

**import** org.springframework.web.bind.annotation.ModelAttribute;

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

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

**import** org.springframework.web.servlet.ModelAndView;

**import** com.stech.model.Employee;

**import** com.stech.service.EmployeeService;

@Controller

**public** **class** EmployeeController {

@SuppressWarnings("unused")

**private** **static** **final** Logger ***logger*** = Logger.*getLogger*(EmployeeController.**class**);

**public** EmployeeController() {

System.***out***.println("EmployeeController()");

}

@Autowired

**private** EmployeeService employeeService;

@RequestMapping(value = "/")

**public** ModelAndView listEmployee(ModelAndView model) **throws** IOException {

List<Employee> listEmployee = employeeService.getAllEmployees();

**for** (Employee employee : listEmployee) {

System.***out***.println(employee.getId());

System.***out***.println(employee.getEmail());

}

model.addObject("listEmployee", listEmployee);

model.setViewName("home");

**return** model;

}

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

**public** ModelAndView newContact(ModelAndView model) {

Employee employee = **new** Employee();

model.addObject("employee", employee);

model.setViewName("EmployeeForm");

**return** model;

}

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

**public** ModelAndView saveEmployee(@ModelAttribute Employee employee) {

**if** (employee.getId() == 0) { // if employee id is 0 then creating the

// employee other updating the employee

employeeService.addEmployee(employee);

} **else** {

employeeService.updateEmployee(employee);

}

**return** **new** ModelAndView("redirect:/");

}

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

**public** ModelAndView deleteEmployee(HttpServletRequest request) {

**int** employeeId = Integer.*parseInt*(request.getParameter("id"));

employeeService.deleteEmployee(employeeId);

**return** **new** ModelAndView("redirect:/");

}

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

**public** ModelAndView editContact(HttpServletRequest request) {

System.***out***.println("ID ="+request.getParameter("id"));

**int** employeeId = Integer.*parseInt*(request.getParameter("id"));

Employee employee = employeeService.getEmployee(employeeId);

ModelAndView model = **new** ModelAndView("EmployeeForm");

model.addObject("employee", employee);

**return** model;

}

}

Package :

Src/java/main :

Package : com.stech.service

**package** com.stech.service;

**import** java.util.List;

**import** com.stech.model.Employee;

**public** **interface** EmployeeService {

**public** **void** addEmployee(Employee employee);

**public** List<Employee> getAllEmployees();

**public** **void** deleteEmployee(Integer employeeId);

**public** Employee getEmployee(**int** employeeid);

**public** Employee updateEmployee(Employee employee);

}

package com.stech.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.stech.dao.EmployeeDAO;

import com.stech.model.Employee;

@Service

@Transactional

public class EmployeeServiceImpl implements EmployeeService {

@Autowired

private EmployeeDAO employeeDAO;

@Transactional

public void addEmployee(Employee employee) {

employeeDAO.addEmployee(employee);

}

@Transactional

public List<Employee> getAllEmployees() {

return employeeDAO.getAllEmployees();

}

@Transactional

public void deleteEmployee(Integer employeeId) {

employeeDAO.deleteEmployee(employeeId);

}

public Employee getEmployee(int empid) {

return employeeDAO.getEmployee(empid);

}

public Employee updateEmployee(Employee employee) {

// TODO Auto-generated method stub

return employeeDAO.updateEmployee(employee);

}

public void setEmployeeDAO(EmployeeDAO employeeDAO) {

this.employeeDAO = employeeDAO;

}

}

Src/main/java

Package : com.stech.model

**package** com.stech.model;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name = "EMP\_TBL")

**public** **class** Employee **implements** Serializable {

**private** **static** **final** **long** ***serialVersionUID*** = -3465813074586302847L;

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

@Column

**private** String name;

@Column

**private** String email;

@Column

**private** String address;

@Column

**private** String telephone;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getTelephone() {

**return** telephone;

}

**public** **void** setTelephone(String telephone) {

**this**.telephone = telephone;

}

}

Src/java/main

Package: com.stech.dao

**package** com.stech.dao;

**import** java.util.List;

**import** com.stech.model.Employee;

**public** **interface** EmployeeDAO {

**public** **void** addEmployee(Employee employee);

**public** List<Employee> getAllEmployees();

**public** **void** deleteEmployee(Integer employeeId);

**public** Employee updateEmployee(Employee employee);

**public** Employee getEmployee(**int** employeeid);

}

**package** com.stech.dao;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.hibernate.SessionFactory;

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

**import** org.springframework.stereotype.Repository;

**import** com.stech.model.Employee;

@Repository

**public** **class** EmployeeDAOImpl **implements** EmployeeDAO {

@Autowired

**private** SessionFactory sessionFactory;

**public** **void** addEmployee(Employee employee) {

sessionFactory.getCurrentSession().saveOrUpdate(employee);

}

@SuppressWarnings("unchecked")

**public** List<Employee> getAllEmployees() {

List <Employee >list = **new** ArrayList<Employee>();

list = sessionFactory.getCurrentSession().createQuery("from Employee")

.list();

**return** list;

}

**public** **void** deleteEmployee(Integer employeeId) {

Employee employee = (Employee) sessionFactory.getCurrentSession().load(

Employee.**class**, employeeId);

**if** (**null** != employee) {

**this**.sessionFactory.getCurrentSession().delete(employee);

}

}

**public** Employee getEmployee(**int** empid) {

**return** (Employee) sessionFactory.getCurrentSession().get(

Employee.**class**, empid);

}

**public** Employee updateEmployee(Employee employee) {

sessionFactory.getCurrentSession().update(employee);

**return** employee;

}

}