springmvc+spring+jdbc+mysql开发【员工管理系统】

开发步骤：

步骤一：导入jar包

c3p0-0.9.1.2.jar

commons-dbutils-1.4.jar

commons-logging.jar

jackson-core-asl-1.9.11.jar

jackson-mapper-asl-1.9.11.jar

mysql-connector-java-5.1.7-bin.jar

org.springframework.asm-3.0.5.RELEASE.jar

org.springframework.beans-3.0.5.RELEASE.jar

org.springframework.context-3.0.5.RELEASE.jar

org.springframework.core-3.0.5.RELEASE.jar

org.springframework.expression-3.0.5.RELEASE.jar

org.springframework.web-3.0.5.RELEASE.jar

org.springframework.web.servlet-3.0.5.RELEASE.jar

步骤二：编写web.xml文档

①注册springmvc核心控制器

②注册争对POST请求的编码器

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

<web-app version=*"2.5"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

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

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*

*http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*>

<!-- 注册springmvc核心控制器 -->

<servlet>

<servlet-name>DispatcherServlet</servlet-name>

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

<init-param>

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

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

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>DispatcherServlet</servlet-name>

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

</servlet-mapping>

<!-- 注册争对POST请求的编码器 -->

<filter>

<filter-name>CharacterEncodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CharacterEncodingFilter</filter-name>

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

</filter-mapping>

<welcome-file-list>

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

</welcome-file-list>

</web-app>

步骤三：引入c3p0-config.xml文件

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

<c3p0-config>

<default-config>

<property name=*"driverClass"*>com.mysql.jdbc.Driver</property>

<property name=*"jdbcUrl"*>jdbc:mysql://localhost:3306/emp</property>

<property name=*"user"*>root</property>

<property name=*"password"*>123456</property>

<property name=*"acquireIncrement"*>2</property>

<property name=*"initialPoolSize"*>5</property>

<property name=*"minPoolSize"*>1</property>

<property name=*"maxPoolSize"*>5</property>

</default-config>

</c3p0-config>

步骤四：在src中加入springmvc的配置文件

<?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:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd*

*"*>

<import resource=*"com/yjy/emp/config/spring-emp.xml"*/>

</beans>

步骤五：编写emp的spring配置文件

<?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: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-3.0.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*"*>

<!-- 注册EmpAction -->

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

<!-- 注册适配器 -->

<bean class=*"org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter"*>

<property name=*"messageConverters"*>

<list>

<bean class=*"org.springframework.http.converter.json.MappingJacksonHttpMessageConverter"*/>

</list>

</property>

</bean>

</beans>

步骤六：编写加载c3p0配置文件类

**public** **class** JdbcUtil {

/\*\*

\* 去src目录下加载c3p0-config.xml配置文件

\*/

**private** **static** ComboPooledDataSource *dataSource* = **new** ComboPooledDataSource();

/\*\*

\* 获取数据源

\*/

**public** **static** ComboPooledDataSource getDataSource() {

**return** *dataSource*;

}

}

步骤七：编写数据库及实体类

databse.sql:

**create** database emp;

use emp;

**DROP** **TABLE** IF **EXISTS** `emp`;

**CREATE** **TABLE** `emp` (

`empno` **varchar**(40) **NOT** **NULL** **DEFAULT** '',

`username` **varchar**(25) **DEFAULT** **NULL**,

`salary` **int**(11) **DEFAULT** **NULL**,

`hiredate` datetime **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`empno`)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

**INSERT** **INTO** `emp` **VALUES** ('1','aa',1222,'3015-05-20');

**INSERT** **INTO** `emp` **VALUES** ('2','BB',12344,'2016-05-20');

**INSERT** **INTO** `emp` **VALUES** ('3','CC',342112,'2016-06-10');

实体对象：

**public** **class** Emp {

**private** Integer empno;

**private** String username;

**private** **double** salary;

**private** Date hiredate;

}

步骤八：编写dao类

@Repository("empDao")

**public** **class** EmpDao {

/\*\*

\* 查询所有员工

\*/

**public** List<Emp> findAll() **throws** Exception{

QueryRunner queryRunner = **new** QueryRunner(JdbcUtil.*getDataSource*());

String sql = "SELECT empno,username,salary,hiredate FROM emp";

**return** queryRunner.query(sql,**new** BeanListHandler<Emp>(Emp.**class**));

}

**public** **static** **void** main(String[] args) **throws** Exception {

EmpDao dao = **new** EmpDao();

**for**(Emp emp : dao.findAll()){

System.***out***.println(emp.toString());

}

}

}

步骤九：编写service层

@Service("empService")

**public** **class** EmpService {

@Resource

**private** EmpDao empDao;

//查询所有员工

**public** List<Emp> findAllEmp() **throws** Exception{

**return** empDao.findAll();

}

}

步骤十：编写action层

@Controller

@RequestMapping("/emp")

**public** **class** EmpAction {

@Resource

**private** EmpService empService;

//查询所有的员工

@RequestMapping("/findAllEmpMethod")

**public** @ResponseBody List<Emp> findAllEmpMethod() **throws** Exception{

//调用业务层，返回List对象

List<Emp> empList = empService.findAllEmp();

// for(Emp e : empList){

// System.out.println(e);

// }

**return** empList;

}

}

步骤十一：编写页面

<%@ page language=*"java"* contentType=*"text/html; charset=utf-8"*

pageEncoding=*"utf-8"*%>

<!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=utf-8"*>

<title>显示所有员工</title>

<script type=*"text/javascript"* src=*"js/jquery-1.10.1.min.js"*></script>

<script type=*"text/javascript"*>

$(**function**(){

**var** url="${pageContext.request.contextPath}/emp/findAllEmpMethod.action";

**var** args={"time":**new** Date()};

$.getJSON(url,args,**function**(data){

**if**(data.length==0){

alert("当前没有信息！！");

}**else**{

**for**(**var** i = 0; i < data.length; i++){

**var** empno = data[i].empno;

**var** username = data[i].username;

**var** salary = data[i].salary;

**var** hiredate = data[i].hiredate;

$("#showEmp").append(

"<tr>"

+"<td>"+empno+"</td>"

+"<td>"+username+"</td>"

+"<td>"+salary+"</td>"

+"<td>"+hiredate+"</td>"

+"</tr>"

);

}

}

});

})

</script>

</head>

<body>

<table id=*"showEmp"* border=*"1"* cellpadding=*"5"*>

<tr>

<td>学号</td>

<td>姓名</td>

<td>薪水</td>

<td>入职时间</td>

</tr>

</table>

</body>

</html>