# 基于JSP创建电脑知识库

2016160182 王 滔

1. **需求描述**

本项目实现一个简易的知识库：

* 1. 管理员可以对知识库记录做增加、修改和删除。
  2. 每个员工登录后可以修改自己的资料，并查看其他人的通讯信息；
  3. 使用Struts MVC + Spring IOC + Hibernate 或 Spring MVC + Spring IOC + Hibernate 的结构。

**附：页面功能设计参考**

**页面一： 登陆**

用户名：\_\_\_\_\_

密 码：\_\_\_\_\_\_

登陆按钮

**说明：**1、系统管理员的用户名和密码是预先存在数据库中；

1. 系统管理员登录后，进入维护电脑知识的界面（界面二）；
2. 普通用户登录后，进入查询界面（界面三）；

**页面二：系统管理员维护电脑知识**

**序号 知识类别 标题 知识内容 更新时间 操作**

1 硬件保养 做好防静电 静电有可能照成电脑损坏 2013-07-09 修改 删除

2 上网技巧 解除禁止网页内容 点选IE浏览器 2013-08-01 修改 删除

……..

-------------------------------------------------------------------------------------------------------------------------------------

**知识类别**：\_\_\_\_

**标 题**：\_\_\_\_

**知识内容**：\_\_\_\_\_

**更新时间**：\_\_\_\_\_

**提交按钮**

说明：1、本页是系统管理员登录后的首页，进行用户的列表、新加、删除、修改；

2、点提交按钮后，新加的知识录入系统后，回到本页，刷新本页面数据；

3、删除前必须弹出提示框，确认后再删除；

4、修改界面可以重做，也可以就用本页上部分的新加用户部分；

**页面三：普通用户查询电脑知识**

**序号 知识类别 标题 知识内容 更新时间 操作**

1 硬件保养 做好防静电 静电有可能照成电脑损坏 2013-07-09

2 上网技巧 解除禁止网页内容 点选IE浏览器 2013-08-01

……..

-------------------------------------------------------------------------------------------------------------------------------------

**说明：**1、进行用户的查询列表，无操作权限；

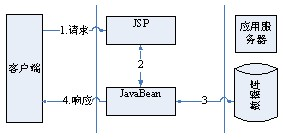
1. **方案设计**

采用Struts MVC + Spring IOC + Hibernate 的结构实现。

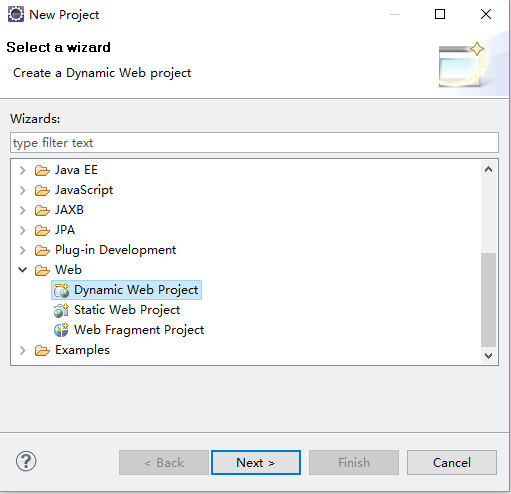
**1.**

1. 语言：JAVA
2. 工具：eclipse
3. 中间件：apache-tomcat-7.0.41
4. 数据库：Mysql

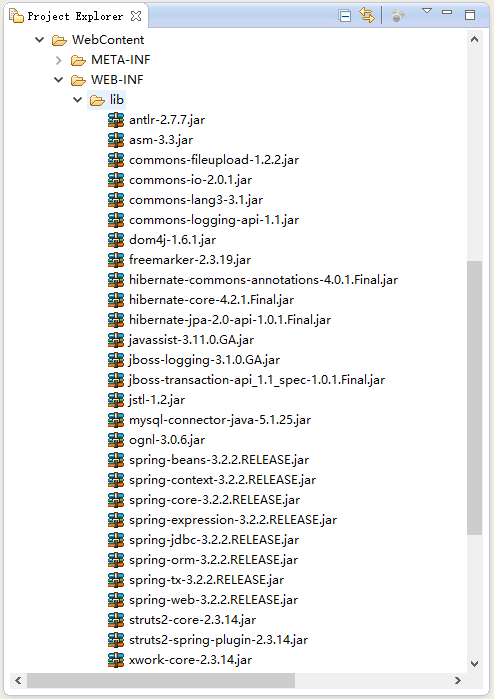
**2.服务示意图**

****

1. **实施步骤：**
2. **创建java Web工程**



1. **导入常用jar包**



1. **编写web配置文件**



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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>EXAM\_WT</display-name>

<welcome-file-list>

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

</welcome-file-list>

<filter>

<filter-name>struts</filter-name>

<filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>struts</filter-name>

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

</filter-mapping>

<listener>

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

</listener>

<context-param>

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

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

</context-param>

</web-app>

1. **创建strtus配置文件**

（1）struts.xml

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

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"

"http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

<include file=*"struts-\*.xml"* />

</struts>

（2）struts-admin.xml

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

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"

"http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

<package name=*"admin"* extends=*"struts-default"*>

<action name=*"admin\_\*"* class=*"adminAction"* method=*"{1}"*>

<result name=*"login\_success"* type=*"redirectAction"*>product\_query</result>

<result name=*"login\_input"*>/login.jsp</result>

</action>

</package>

</struts>

(3) struts-admin.xml

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

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"

"http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

<package name=*"computer"* extends=*"struts-default"*>

<action name=*"computer\_\*"* class=*"computerAction"* method=*"{1}"*>

<result name=*"add\_success"* type=*"redirectAction"*>computer\_query</result>

<result name=*"query\_success"* type=*"redirect"*>/admin.jsp</result>

<result name=*"search\_success"* type=*"redirect"*>/edit.jsp</result>

<result name=*"update\_success"* type=*"redirectAction"*>computer\_query</result>

<result name=*"remove\_success"* type=*"redirectAction"*>computer\_query</result>

</action>

</package>

</struts>

1. **创建XML配置文件**
2. applicationContext.xml

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

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

<property name=*"url"* value=*"jdbc:mysql:///test"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"system"* />

</bean>

<bean id=*"sessionFactory"* class=*"org.springframework.orm.hibernate4.LocalSessionFactoryBean"*>

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

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.MySQLDialect</prop>

<prop key=*"hibernate.show\_sql"*>true</prop>

<prop key=*"hibernate.format\_sql"*>true</prop>

<prop key=*"hibernate.hbm2ddl.auto"*>update</prop>

</props>

</property>

<property name=*"mappingResources"*>

<list>

<value>admin.xml</value>

<value>user.xml</value>

<value>computer.xml</value>

</list>

</property>

</bean>

<bean id=*"adminDao"* class=*"exam.dao.impl.AdminDaoImpl"*>

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

</bean>

<bean id=*"userDao"* class=*"exam.dao.impl.UserDaoImpl"*>

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

</bean>

<bean id=*"computerDao"* class=*"exam.dao.impl.ComputerDaoImpl"*>

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

</bean>

<bean id=*"adminAction"* class=*"exam.action.AdminAction"*>

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

</bean>

<bean id=*"userAction"* class=*"exam.action.UserAction"*>

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

</bean>

<bean id=*"computerAction"* class=*"exam.action.ComputerAction"*>

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

</bean>

1. admin.xml

<hibernate-mapping>

<class name=*"exam.model.Admin"*>

<id name=*"id"*>

<generator class=*"native"* />

</id>

<property name=*"username"* />

<property name=*"password"* />

<property name=*"checkpwd"* />

<property name=*"email"* />

</class>

</hibernate-mapping>

1. computer.xml

<hibernate-mapping>

<class name=*"exam.model.Computer"*>

<id name=*"id"*>

<generator class=*"native"* />

</id>

<property name=*"type"* />

<property name=*"title"* />

<property name=*"content"* />

<property name=*"modify\_date"* />

</class>

</hibernate-mapping>

1. **创建实体类**

**demo.model**

1. Admin.java
2. Computer.java

demo.action

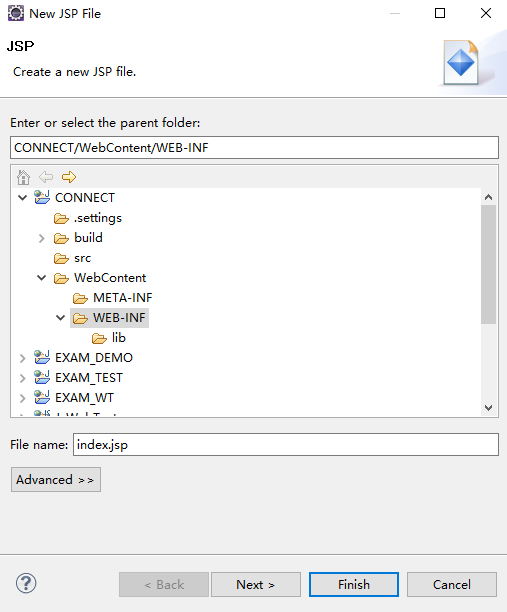
1. Adminaction.java
2. Computer action.java

demo.dao

1. Admindao.java
2. Computerdao.java

demo.dao.mpl

1. AdminDaoImpl.java
2. Computer DaoImpl.java
3. **创建JSP文件**
4. **创建引导页面Index.jsp**



<%@ page pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>login page</title>

</head>

<body>

<h1>Login Page</h1>

<form action=*"admin\_login"* method=*"post"*>

username: <input name=*"admin.username"*><br>

password: <input type=*"password"* name=*"admin.password"*><br>

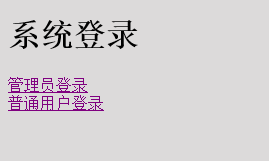
<input type=*"submit"* value=*"LOGIN"*>

</form>

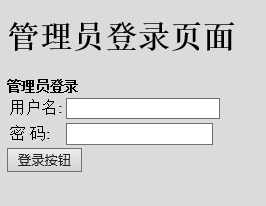
${requestScope.message }

</body>

</html>



1. **创建管理员登录页面login.jsp**

****

<body>

<h1>Login Page</h1>

<form action=*"admin\_login"* method=*"post"*>

username: <input name=*"admin.username"*><br>

password: <input type=*"password"* name=*"admin.password"*><br>

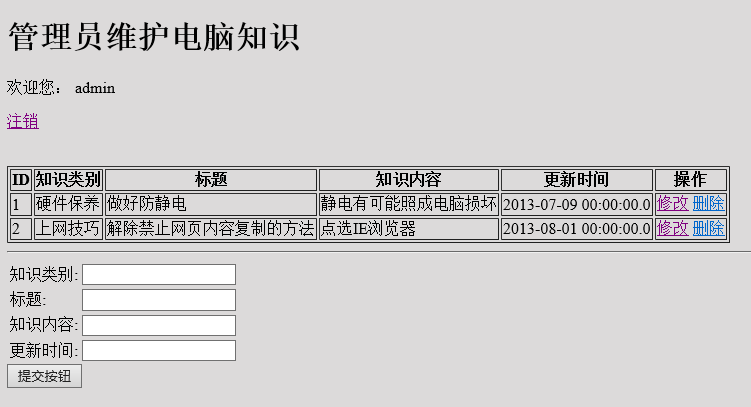
<input type=*"submit"* value=*"LOGIN"*>

</form>

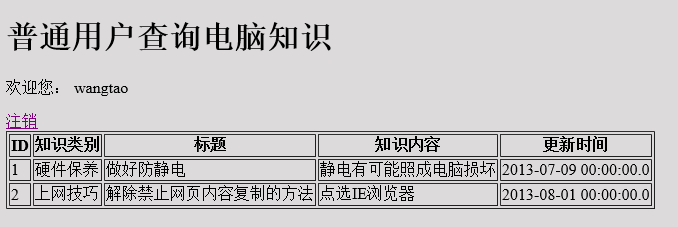
${requestScope.message }

</body>

1. **创建管理员维护页面admin.jsp**



1. **创建用户查询页面user.jsp**

****

1. **总结**

本案例介绍了基于JSP实现Web程序开发和实现过程，用JSP技术设计的系统，具有跨平台性，运行效率高的特点。JSP页面开发技术帮助开发者和设计者能够轻松实现业务的开发和维护工作。JSP实现了基于Web应用程序平台独立性，实现了用户界面和程序内容的分离，是的网页设计人员可以在不改变JSP程序的情况下更改用户界面。