SSM+EasyUI整合（OA办公系统）

一、EasyUI

1.简介

easyui是一种基于jQuery、Angular.、Vue和React的用户界面插件集合。

easyui为创建现代化，互动，JavaScript应用程序，提供必要的功能。

使用easyui你不需要写很多代码，你只需要通过编写一些简单HTML标记，就可以定义用户界面。

easyui是个完美支持HTML5网页的完整框架。

easyui节省您网页开发的时间和规模。

easyui很简单但功能强大的。

在jQuery 和 HTML5 上轻松使用EasyUI

jQuery EasyUI 提供易于使用的组件，它使 Web 开发人员能快速地在流行的 jQuery 核心和 HTML5 上建立程序页面。

1.1 两种使用方式：

直接在 HTML 声明组件

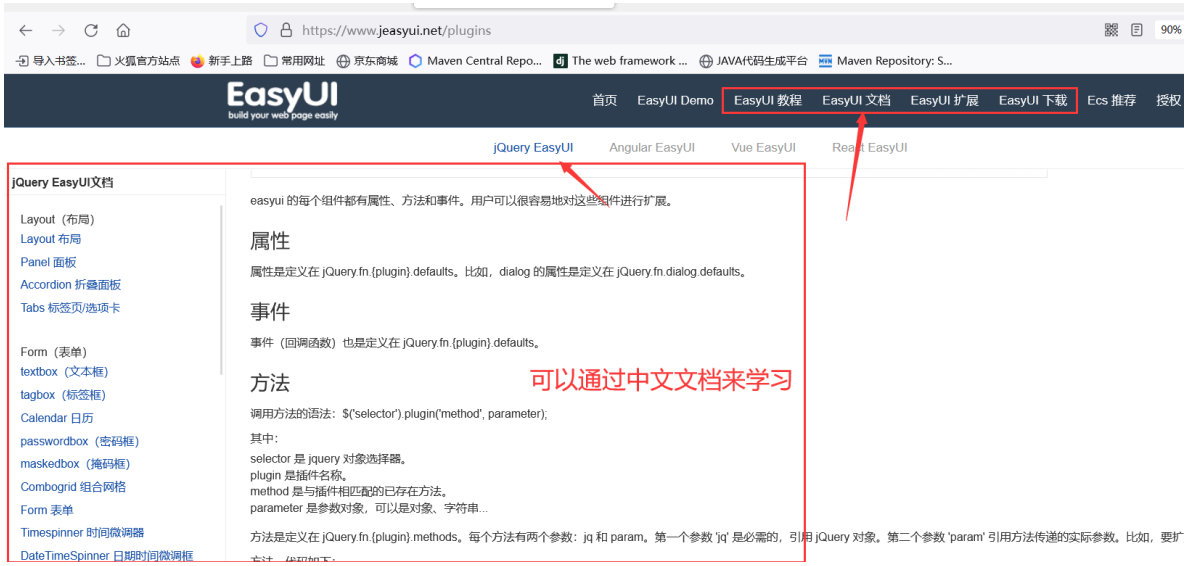
```
<div class="easyui-dialog" style="width:400px;height:200px"
    data-options="
        title:'My Dialog',
        iconCls:'icon-ok',
        onOpen:function(){}">
    dialog content.
</div>
```

编写 JS 代码来创建组件

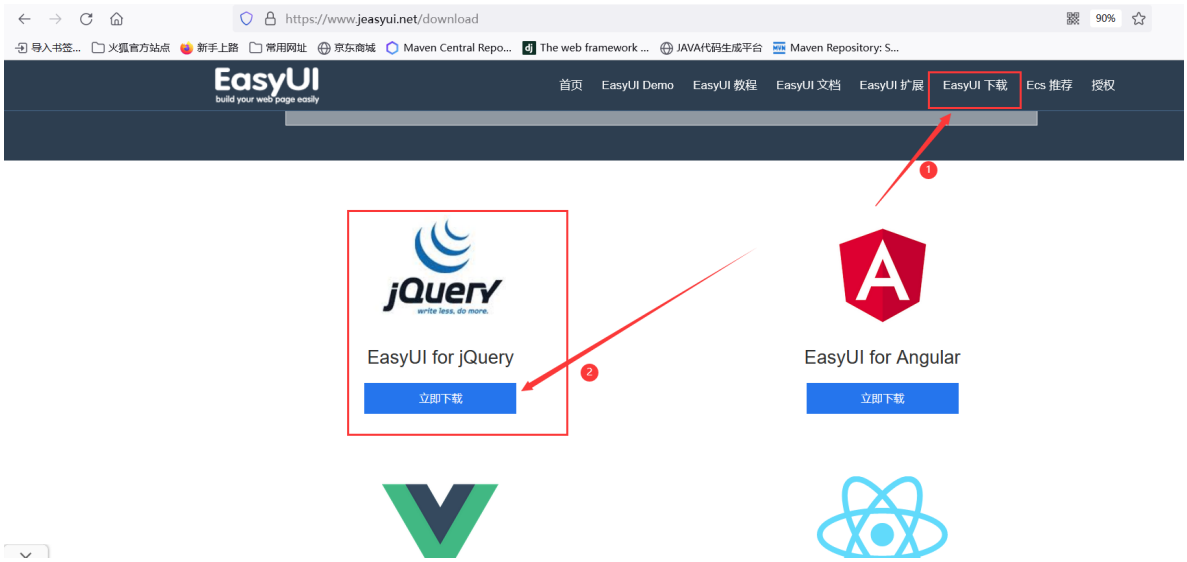
```
<input id="cc" style="width:200px" />

$('#cc').combobox({
    url: ...,
    required: true,
    valueField: 'id',
    textField: 'text'
});
```

接下来我们一起来学习EasyUI， just do it!

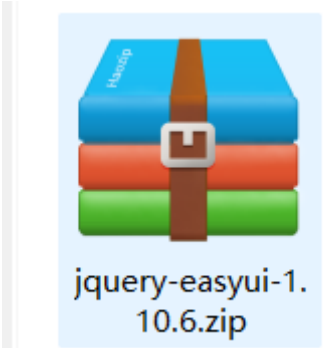
> EasyUI官网：https://www.jeasyui.net/



EasyUI下载



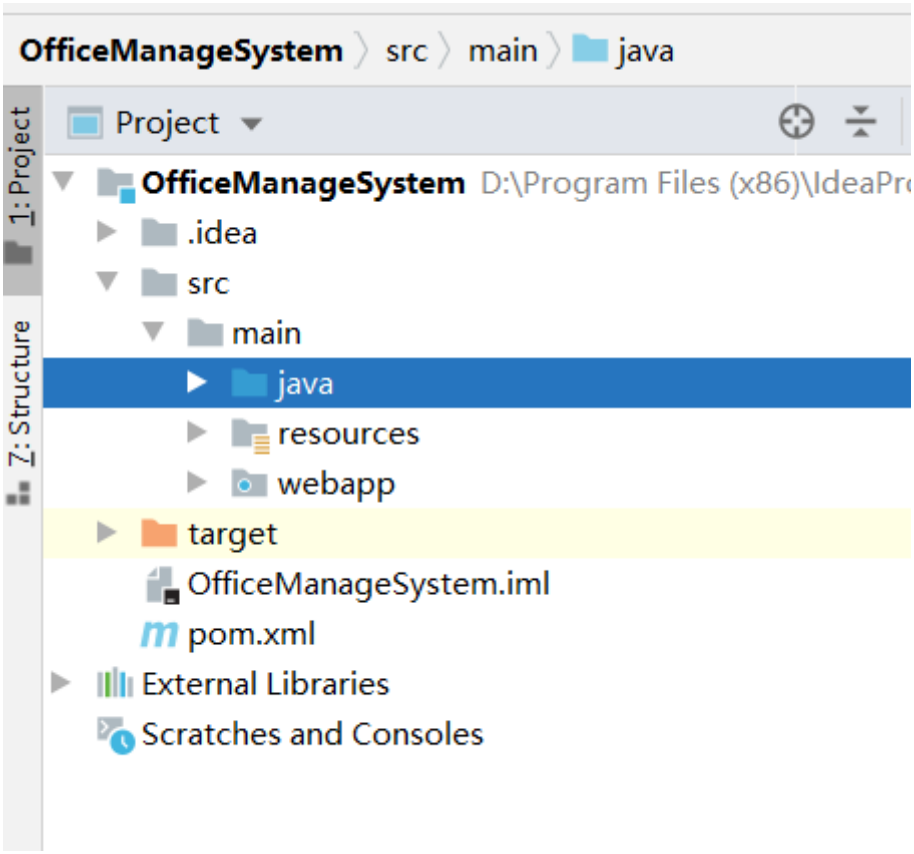
下载好的压缩包 解压即可



接下来我们一起来使用EasyUI， just do it!

二、SSM+EasyUI整合（OA办公系统）

1.创建项目结构(Maven的Web项目)

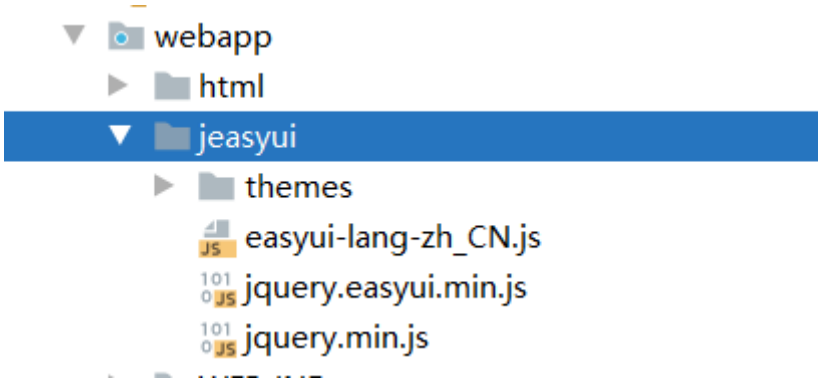


2.pom.xml（Maven依赖）

```
<dependencies>
    <!-- 整合spring框架（包含springmvc）这个jar文件包含springmvc开发时的核心类，
    同时也会将依赖的相关jar文件引入进来(spring的核心jar文件也包含在内)
     -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>4.1.3.RELEASE</version>
    </dependency>

    <!--这个jar文件包含对Spring对JDBC数据访问进行封装的所有类 -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-jdbc</artifactId>
      <version>4.1.3.RELEASE</version>
    </dependency>

    <!-- 整合mybatis框架 -->
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
      <version>3.2.8</version>
    </dependency>
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis-spring</artifactId>
      <version>1.2.2</version>
    </dependency>

    <!--连接池-->
    <dependency>
```

```
      <groupId>com.alibaba</groupId>
      <artifactId>druid</artifactId>
      <version>1.1.16</version>
    </dependency>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.29</version>
    </dependency>

    <!-- 整合log4j -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
      <version>1.7.36</version>
    </dependency>
    <!-- Jackson Json处理工具包 -->
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.13.3</version>
    </dependency>
    <!-- Servlet/JSP/JSTL -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>servlet-api</artifactId>
      <version>2.4</version>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jsp-api</artifactId>
      <version>2.0</version>
    </dependency>

    <!--jstl-->
    <dependency>
      <groupId>jstl</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>

    <!--单元测试-->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.13.2</version>
      <scope>test</scope>
    </dependency>

    <!--lombok-->
    <dependency>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
      <version>1.18.24</version>
    </dependency>
  </dependencies>
```

3.在webapp目录下创建jeasyui文件夹，导入需要的包和文件




```
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="jeasyui/easyui-lang-zh_CN.js"></script>
```

4.自己使用EasyUI编写页面

首页页面代码（index.html）

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>OA首页</title>
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="jeasyui/easyui-lang-zh_CN.js"></script>
    <script>
        function openTabs(url,text) {
            if($("#tabs").tabs('exists',text)){
                $("#tabs").tabs('select',text);
            }else{
                let myContext =
                    "<iframe frameborder='0' scrolling='auto' " +
                    "style='width:100%;height:100%' src="+url+"></iframe>";
                $("#tabs").tabs('add', {
                    title: text,
                    closable: true,
```

```
                    content: myContext
                })
            }
        }
    </script>
</head>

<body class="easyui-layout">
    <div data-options="region:'north',title:'OA办公系统',split:true"
style="height:120px;">

    </div>

    <div data-options="region:'west',title:'导航菜单',split:true"
style="width:150px;">
        <div  class="easyui-accordion" fit="true">
            <div title="系统信息管理">
                <a href="#" class="easyui-linkbutton" style="width:100%;"
                   plain="true" onclick="openTabs('html/user.html','用户信息管
理')">
                    用户信息管理
                </a>
                <a href="#" class="easyui-linkbutton" style="width:100%;"
                   plain="true" onclick="openTabs('html/logs.html','日志信息管
理')">
                    日志信息管理
                </a>
            </div>

        </div>

    </div>
    <div data-options="region:'center',title:'显示'"
style="padding:5px;background:#eee;">
        <div id="tabs"  class="easyui-tabs" fit="true">
            <!--<closable="true" 关闭按钮 panel-->
            <div title="首页">
                欢迎使用OA办公管理系统
            </div>
        </div>
    </div>
</body>
</html>
```

然后在webapp下创建html文件夹创建两个新的html页面user.html、logs.html

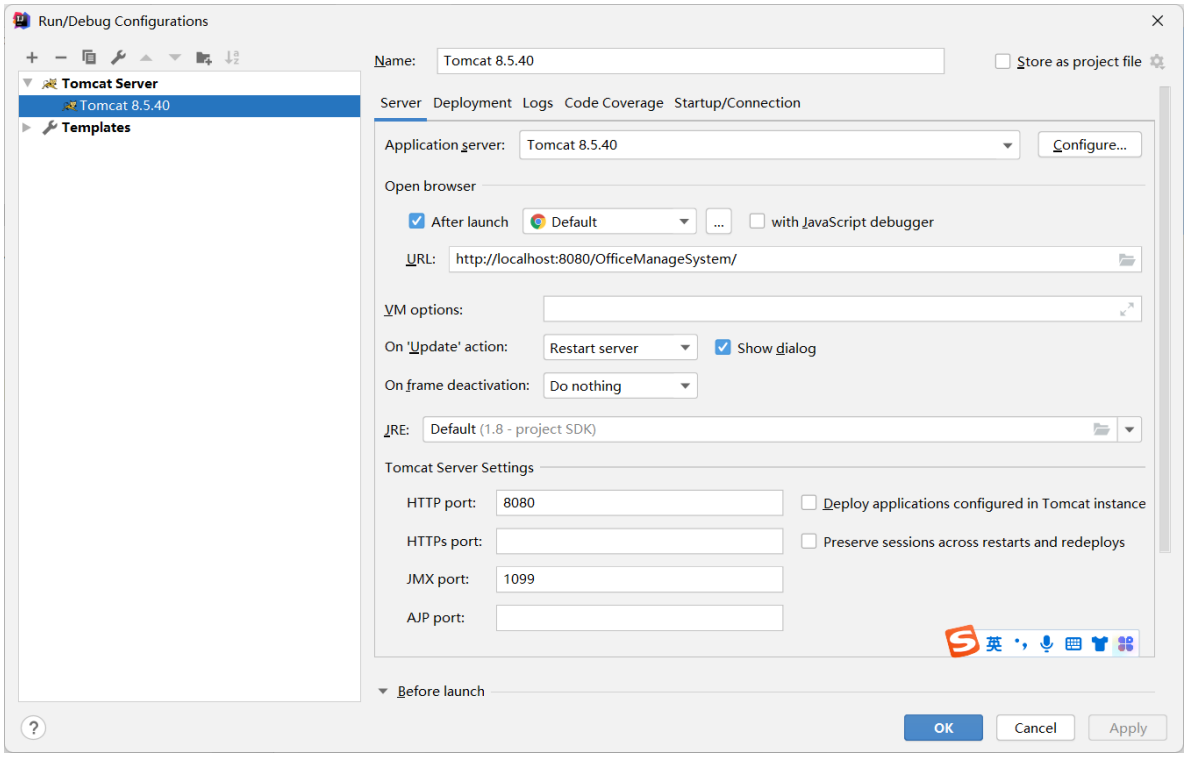
user.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>用户信息管理</title>
</head>
<body>
    用户信息管理
</body>
</html>
```

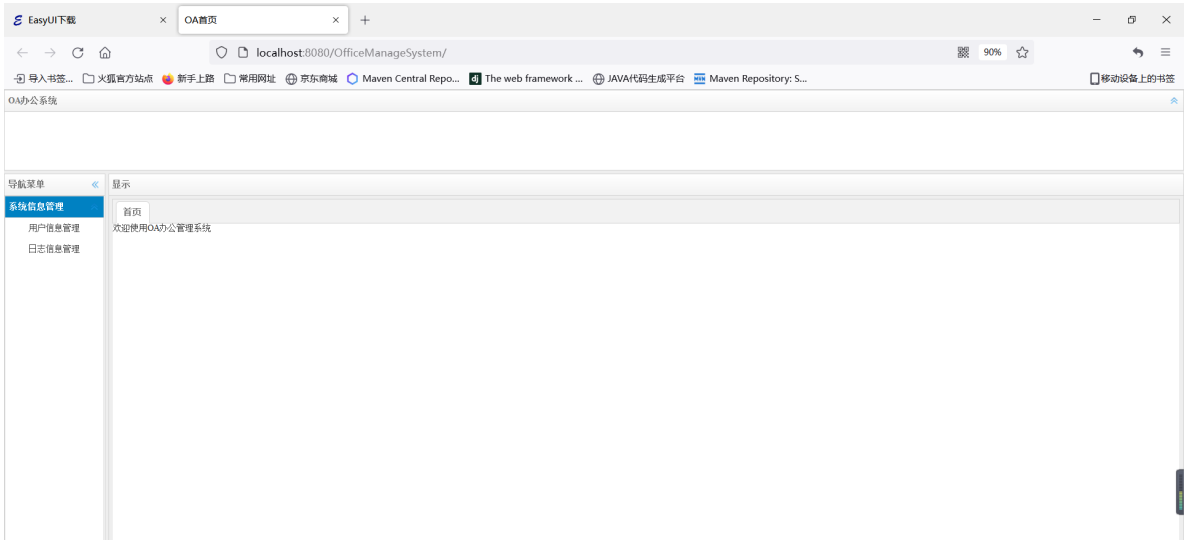
logs.html

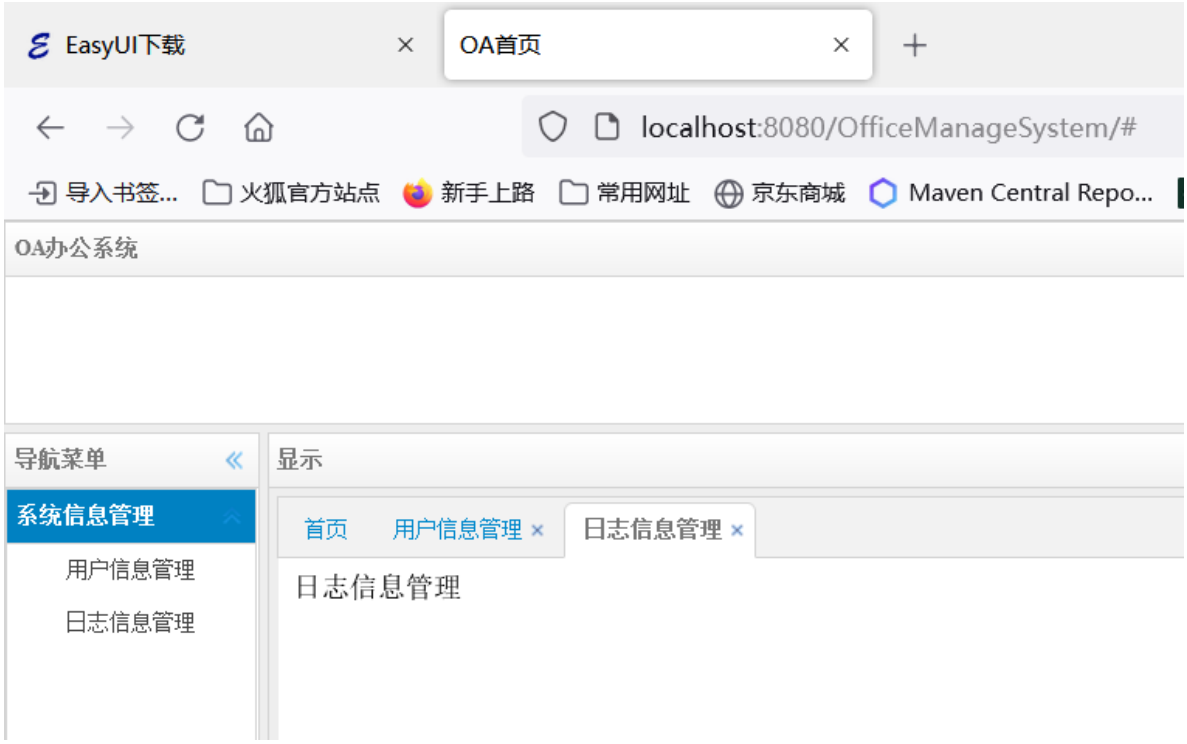
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>日志信息管理</title>
</head>
<body>
    日志信息管理
</body>
</html>
```

5.配置Tomcat(自行配置)



启动后访问首页查看效果





6.创建数据库

```
DROP DATABASE IF EXISTS `oadb`;
CREATE DATABASE  `oadb`  CHARSET utf8;
USE `oadb`;

DROP TABLE IF EXISTS `tb_user`;
CREATE TABLE `tb_user` (
  `uid` int NOT NULL AUTO_INCREMENT COMMENT '主键',
  `uname` varchar(50) DEFAULT NULL COMMENT '名称',
  `nickname` varchar(50) DEFAULT NULL COMMENT '昵称',
  `password` varchar(50) DEFAULT NULL COMMENT '密码',
  `images` text COMMENT '头像',
  `stat` int DEFAULT '0' COMMENT '0启用 1禁用',
  `createtime` varchar(50) DEFAULT NULL COMMENT '修改时间',
  `modifytime` varchar(50) DEFAULT NULL COMMENT '创建时间',
  `deleted` int DEFAULT '0' COMMENT '删除标记 0未删除 1 已删除',
  PRIMARY KEY (`uid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```sql
DROP TABLE IF EXISTS `tb_logs`;
CREATE TABLE `tb_logs` (
  `operid` INT NOT NULL AUTO_INCREMENT,
  `opertime` VARCHAR(50) COLLATE utf8_bin DEFAULT NULL,
  `opername` VARCHAR(50) COLLATE utf8_bin DEFAULT NULL,
  `ip` VARCHAR(50) COLLATE utf8_bin DEFAULT NULL,
  `methods` TEXT COLLATE utf8_bin,
  `ddesc` TEXT COLLATE utf8_bin,
  PRIMARY KEY (`operid`)
) ENGINE=INNODB AUTO_INCREMENT=29 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

7.配置文件

7.1 数据配置文件和日志文件

jdbc.properties

```
jdbc.driver=com.mysql.cj.jdbc.Driver
#jdbc.url=jdbc:mysql://localhost:3306/oadb?
serverTimezone=Asia/Shanghai&useSSL=false
jdbc.url=jdbc:mysql:///oadb
jdbc.username=root
jdbc.password=root
```

log4j.properties

```
log4j.rootLogger=INFO,A1
log4j.logger.cn.yhmis.mapper =DEBUG
log4j.appender.A1=org.apache.log4j.ConsoleAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss,SSS} [%t]
[%c]-[%p] %m%n
```

7.2 编写spring核心配置文件

在resources/spring目录下，创建spring的核心配置文件：**applicationContext.xml**

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
 xmlns:p="http://www.springframework.org/schema/p"
    xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
    http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.0.xsd">



</beans>
```

7.3 测试spring框架:

(1)创建com.tedu.pojo.User类

```
public class User { }
```

(2)在applicationContext.xml配置User类的bean实例

```
<!-- 声明bean对象 -->
<bean id="user" class="com.tedu.pojo.User"></bean>
```

(3)编写测试类: TestSpring.java

```
package com.tedu.test;
/** 测试类: 测试spring开发环境的类 */
public class  TestSpring {

  @Test
  public void testSpring(){

   //1.加载Spring的核心配置文件
   ClassPathXmlApplicationContext ac =
                new ClassPathXmlApplicationContext(
                            "spring/applicationContext.xml");
  //2.获取bean实例
   User user = (User) ac.getBean("user");
    System.out.println(user);

  }
}
```

7.3 整合springmvc框架

前面在添加spring的jar包同时, 将springmvc的jar包也引入了，因此这里不再引入springmvc的jar包

7.3.1、创建springmvc的核心配置文件

在resources/spring目录下，创建springmvc的核心配置文件：**springmvc-config.xml**，内容配置如下：

```
<?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/mvc
                        http://www.springframework.org/schema/mvc/spring-mvc-
4.0.xsd
                        http://www.springframework.org/schema/beans
                        http://www.springframework.org/schema/beans/spring-
beans-4.0.xsd
                        http://www.springframework.org/schema/context
                        http://www.springframework.org/schema/context/spring-
context-4.0.xsd">
    <!-- 1.配置注解驱动，用于识别注解（比如@Controller） -->
    <mvc:annotation-driven></mvc:annotation-driven>

    <!-- 2.配置需要扫描的包：spring自动去扫描 base-package 下的类，
        如果扫描到的类上有 @Controller、@Service、@Component等注解，
        将会自动将类注册bean对象
     -->
    <context:component-scan base-package="com.tedu.controller">
    </context:component-scan>

    <!--3.放行静态资源-->
    <mvc:default-servlet-handler></mvc:default-servlet-handler>

    <!-- 4.配置内部资源视图解析器
        prefix:配置路径前缀
        suffix:配置文件后缀
        由于使用的是EasyUI 所有前端页面全部是html为主。
        视图解析器用不上。可以注释掉
     -->
    <!--<bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/pages/"/>
        <property name="suffix" value=".jsp"/>
    </bean> -->


</beans>
```

7.3.2、在web.xml中配置springmvc乱码处理过滤器

```
<!-- 乱码处理过滤器 -->
<filter>
    <filter-name>encodingFilter</filter-name>
    <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <!-- 指定编码集 -->
    <init-param>
```

```
        <param-name>encoding</param-name>
        <param-value>utf-8</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <!-- 指定拦截方式为拦截所有请求 -->
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

7.3.3、在web.xml中配置springmvc

```
<!-- 配置springmvc, 将所有请求交给springmvc来处理 -->
<servlet>
    <servlet-name>springmvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
    <!-- 指定spring及springmvc配置文件的位置 -->
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:spring/*.xml</param-value>
    </init-param>
</servlet>
<!-- 其中的斜杠( / )表示拦截所有请求, 所有请求都要经过springmvc -->
<servlet-mapping>
    <servlet-name>springmvc</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

7.3.4、测试springmvc框架:

(1)创建home.jsp页面

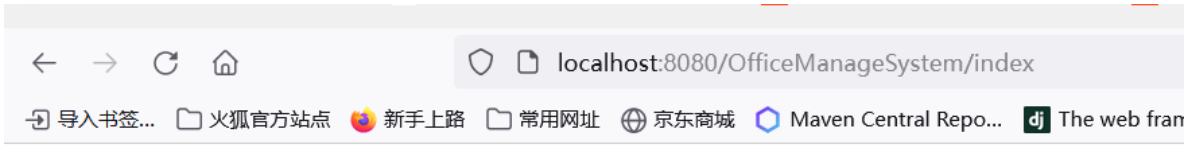
在WEB-INF/pages/目录下，创建index.jsp页面。

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>测试</title>
</head>
<body>
    <h1>SpringMVC</h1>
</body>
</html>
```

(2)创建测试Controller: TestController

```
package com.tedu.controller;

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

/** 测试类：测试springmvc开发环境 */
@Controller
```

```
public class TestController {
    @RequestMapping("/index")
    public String test(){
        return "index";
    }
}
```



SpringMVC

7.4 配置Mybatis框架

7.4.1、创建Mybatis的核心配置文件

在resources/mybatis目录下，创建mybatis的核心配置文件：mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">

<!-- MyBatis的全局配置文件 -->
<configuration >
    <!-- 1.配置开发环境 -->
    <environments default="develop">
        <!-- 这里可以配置多个环境，比如develop，test等 -->
        <environment id="develop">
            <!-- 1.1.配置事务管理方式：JDBC：将事务交给JDBC管理（推荐） -->
            <transactionManager type="JDBC"></transactionManager>
            <!-- 1.2.配置数据源，即连接池方式:JNDI/POOLED/UNPOOLED -->
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.cj.jdbc.Driver"/>
                <property name="url" value="jdbc:mysql://localhost:3306/oadb?
characterEncoding=utf-8"/>
                <property name="username" value="root"/>
                <property name="password" value="root"/>
            </dataSource>
        </environment>
    </environments>

    <!-- 2.加载Mapper配置文件,路径以斜杠间隔: xx/xx/../xx.xml -->
    <mappers>
        <mapper resource="xx/xx/../xx.xml"/>
    </mappers>
</configuration>
```

7.4.2、创建实体类User，用于封装所有的用户信息

```java
package com.tedu.pojo;

import lombok.Data;
import java.io.Serializable;

@Data
public class User implements Serializable {
    private Integer uid;
    private String uname;
    private String password;
    private String nickName;
    private String images;
    private Integer stat;
    private String createTime;
    private String modifyTime;
    private Integer deleted;
}
```

7.4.3、创建实体类映射文件UserMapper.xml

在src/main/resources/mybatis/mapper目录下，创建User实体类的映射文件—UserMapper.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<!-- 用户表的映射文件    namespace值为对应接口的全路径 -->
<mapper namespace="com.tedu.dao.UserDao">
    <!-- 1.查询所有用户信息，id值为对应接口中方法的名字
        resultType指定将查询的结果封装到哪个pojo对象中
     -->
    <select id="findAll" resultMap="userMapper">
        select * from tb_user where deleted=0
    </select>
    <resultMap id="userMapper" type="com.tedu.pojo.User">
        <id property="uid" column="uid"/>
        <result property="uname" column="uname"/>
        <result property="nickName" column="nickname"/>
        <result property="password" column="password"/>
        <result property="stat" column="stat"/>
        <result property="images" column="images"/>
        <result property="createTime" column="createtime"/>
        <result property="modifyTime" column="modifytime"/>
    </resultMap>
</mapper>
```

7.4.4、在mybatis的全局配置文件中引入UserMapper.xml映射文件

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
        PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-config.dtd">

<!-- MyBatis的全局配置文件 -->
```

```
<configuration >
    <!-- 1.配置开发环境 -->
    <environments default="develop">
        <!-- 这里可以配置多个环境，比如develop，test等 -->
        <environment id="develop">
            <!-- 1.1.配置事务管理方式：JDBC：将事务交给JDBC管理（推荐） -->
            <transactionManager type="JDBC"></transactionManager>
            <!-- 1.2.配置数据源，即连接池方式:JNDI/POOLED/UNPOOLED -->
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.cj.jdbc.Driver"/>
                <property name="url" value="jdbc:mysql://localhost:3306/oadb?
characterEncoding=utf-8"/>
                <property name="username" value="root"/>
                <property name="password" value="root"/>
            </dataSource>
        </environment>
    </environments>

    <!-- 2.加载Mapper配置文件,路径以斜杠间隔: xx/xx/../xx.xml -->
    <mappers>
        <mapper resource="mybatis/mapper/UserMapper.xml"/>
    </mappers>
</configuration>
```

7.4.5、创建dao接口，编写查询所有用户信息方法

创建com.tedu.dao.UserDao接口，并根据UserMapper.xml文件中的sql语句，提供findAll方法

```
package com.tedu.dao;

import com.tedu.pojo.User;
import org.springframework.stereotype.Repository;
import java.util.List;

/**
 * UserDao接口
 * 声明增删改查方法,对用户信息进行操作
 */
@Repository
public interface UserDao {
    /**
     * 1.查询所有用户信息
     */
    public List<User> findAll();

}
```

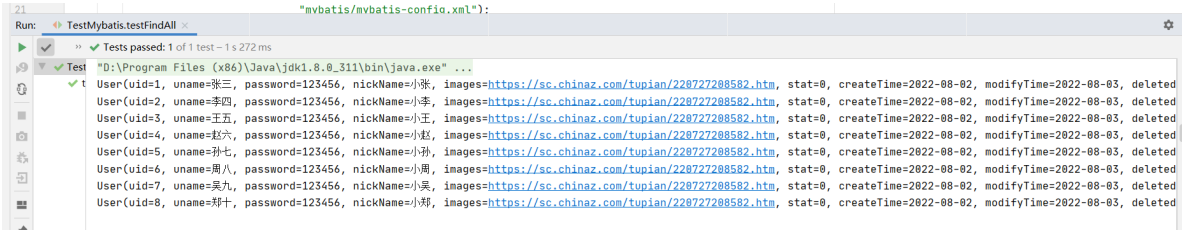
7.4.6、测试mybatis内容

创建测试com.tedu.controller.TestMybatis类，对mybatis开发环境进行测试

```
package cn.tedu.test;

import java.io.InputStream;
import java.util.List;
```

```
import com.tedu.dao.UserDao;
import com.tedu.pojo.User;
import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

/** 测试类：测试mybatis开发环境 */
public class TestMybatis {
    public static void main(String[] args) throws Exception {
        //1.读取mybatis-config.xml核心文件
        InputStream in = Resources.getResourceAsStream(
                                    "mybatis/mybatis-config.xml");
        //2.获取SqlSessionFactory工厂
        SqlSessionFactory factory =
                new SqlSessionFactoryBuilder()
                                            .build(in);
        //3.获取SqlSession对象
        SqlSession session = factory.openSession();

        //4.获取DoorMapper接口的实例
        UserDao userDao = session.getMapper(UserDao.class);
        //5.调用findAll方法查询所有用户信息
        List<User> list = userDao.findAll();
        //6.遍历所有用户信息
        for(User user : list){
            System.out.println(user);
        }
    }
}
```

执行结果：



7.5 整合spring和mybatis

7.5.1、修改mybatis核心配置文件

修改mybatis-config.xml文件，将连接池等配置移除，在spring中配置

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">

<!-- MyBatis的全局配置文件 -->
<configuration >
    <!-- 1.配置开发环境 -->
```

```
    <!-- 1.1.配置事务管理方式：JDBC：将事务交给JDBC管理（推荐） -->
    <!-- 1.2.配置数据源，即连接池方式:JNDI/POOLED/UNPOOLED -->
    <!-- 2.加载Mapper配置文件,路径以斜杠间隔: xx/xx/../xx.xml -->

</configuration>
```

7.5.2、在applicationContext.xml中配置druid连接池

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
       xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
    http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.0.xsd">

    <!-- 测试spring开发环境，声明User类的bean实例 -->
    <bean id="user" class="com.tedu.pojo.User"></bean>

    <!-- 加载jdbc.properties文件的位置 -->
    <context:property-placeholder location="classpath:jdbc.properties"/>

    <!-- 配置druid连接池 ，id是固定值，class是druid连接池类的全路径 -->
    <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource">
        <!-- 配置连接数据库的基本信息 -->
        <property name="driverClassName" value="${jdbc.driver}"></property>
        <property name="url" value="${jdbc}"></property>
        <property name="username" value="${jdbc.username}"></property>
        <property name="password" value="${jdbc.password}"></property>
    </bean>

    <!-- 3.配置需要扫描的包(service层)：spring自动去扫描 base-package下的类，
        如果扫描到的类上有 @Controller、@Service、@Component等注解，
        将会自动将类注册为bean（即由spring创建实例）
     -->
    <context:component-scan base-package="com.tedu.service"></context:component-
scan>

</beans>
```

7.5.3、在resources目录下创建jdbc.properties文件，将连接数据库的基本信息提取到文件中

```
jdbc.driver=com.mysql.cj.jdbc.Driver
#jdbc.url=jdbc:mysql://localhost:3306/oadb?
serverTimezone=Asia/Shanghai&useSSL=false
jdbc.url=jdbc:mysql:///oadb
jdbc.username=root
jdbc.password=root
```

7.5.4、创建applicationContext-mybatis.xml文件：

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
    xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
    http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.0.xsd">

    <!-- 整合spring和mybatis框架 id值是固定值
        将SqlSession等对象的创建交给Spring容器
     -->
    <bean id="sqlSessionFactory"
class="org.mybatis.spring.SqlSessionFactoryBean">
        <!-- 1.指定mybatis核心配置文件的位置 -->
        <property name="configLocation"
                value="classpath:mybatis/mybatis-config.xml"></property>

        <!-- 2.配置连接池(数据源) ref指向连接池bean对象的id值 -->
        <property name="dataSource" ref="dataSource"></property>

        <!-- 3、扫描所有的 XxxMapper.xml映射文件，读取其中配置的SQL语句 -->
        <property name="mapperLocations"
value="classpath:mybatis/mapper/*.xml"/>

    </bean>

    <!-- 4、定义mapper接口扫描器 -->
    <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
        <!-- 扫描所有XxxMapper接口，将接口实例的创建交给spring容器 -->
        <property name="basePackage" value="com.tedu.dao"/>
    </bean>
</beans>
```

7.5.5、复制TestController类，改名为TestSSM

```
package com.tedu.controller;
```

```java
import java.util.List;
import com.tedu.dao.UserDao;
import com.tedu.pojo.User;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;


/** 测试类：测试SSM开发环境 */
@Controller /* 这个注解表示当前类属于Controller层代码 */
public class TestSSM {

    /** 自动装配：由spring自动为属性赋值(对象)  */
    @Autowired
    UserDao mapper;

    @RequestMapping("/testssm")
    public String testSSM(){
        //1.调用findAll方法查询所有用户信息
        List<User> list = mapper.findAll();
        //2.遍历所有用户信息
        for(User user : list){
            System.out.println(user);
        }
        return "index";
    }
}
```

其实还有一种情况就是将applicationContext-mybatis.xml文件中的整合内容放在applicationContext.xml也是可以的

```xml
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
       xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
    http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.0.xsd">

    <!-- 测试spring开发环境，声明User类的bean实例 -->
    <bean id="user" class="com.tedu.pojo.User"></bean>

    <!-- 加载jdbc.properties文件的位置 -->
    <context:property-placeholder location="classpath:jdbc.properties"/>
```

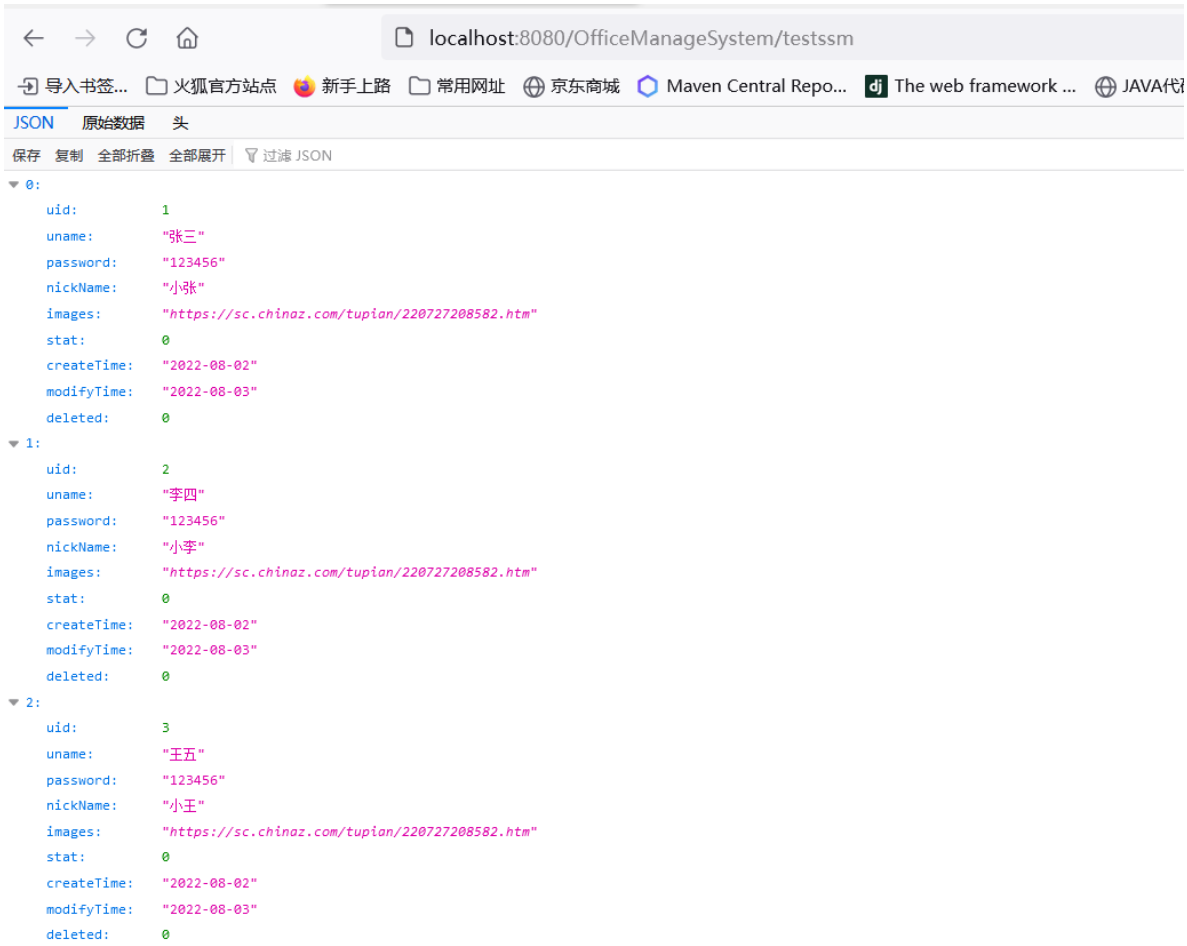
```
    <!-- 配置druid连接池 ，id是固定值，class是druid连接池类的全路径 -->
    <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource">
        <!-- 配置连接数据库的基本信息 -->
        <property name="driverClassName" value="${jdbc.driver}"></property>
        <property name="url" value="${jdbc.url}"></property>
        <property name="username" value="${jdbc.username}"></property>
        <property name="password" value="${jdbc.password}"></property>
    </bean>

    <!-- 3.配置需要扫描的包(service层)：spring自动去扫描 base-package下的类，
        如果扫描到的类上有 @Controller、@Service、@Component等注解，
        将会自动将类注册为bean（即由spring创建实例）
     -->
    <context:component-scan base-package="com.tedu.service"></context:component-
scan>


    <!-- 整合spring和mybatis框架 id值是固定值
       将SqlSession等对象的创建交给Spring容器
    -->
    <bean id="sqlSessionFactory"
class="org.mybatis.spring.SqlSessionFactoryBean">
        <!-- 1.指定mybatis核心配置文件的位置 -->
        <property name="configLocation"
                  value="classpath:mybatis/mybatis-config.xml"></property>

        <!-- 2.配置连接池(数据源) ref指向连接池bean对象的id值 -->
        <property name="dataSource" ref="dataSource"></property>

        <!-- 3、扫描所有的 XxxMapper.xml映射文件，读取其中配置的SQL语句 -->
        <property name="mapperLocations"
value="classpath:mybatis/mapper/*.xml"/>

    </bean>

    <!-- 4、定义mapper接口扫描器 -->
    <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
        <!-- 扫描所有XxxMapper接口，将接口实例的创建交给spring容器 -->
        <property name="basePackage" value="com.tedu.dao"/>
    </bean>

</beans>
```

然后调整Controller类代码 将数据返回到页面(后期可以利用Ajax来解析数据)

```
package com.tedu.controller;

import java.util.List;
import com.tedu.dao.UserDao;
import com.tedu.pojo.User;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```java
/** 测试类：测试SSM开发环境 */
@RestController /* 这个注解表示当前类属于Controller层代码 */
public class TestSSM {

    /** 自动装配：由spring自动为属性赋值(对象)  */
    @Autowired
    UserDao mapper;

    @RequestMapping("/testssm")
    public List<User> testSSM(){
        //1.调用findAll方法查询所有用户信息
        List<User> list = mapper.findAll();
        //2.遍历所有用户信息
        for(User user : list){
            System.out.println(user);
        }
        return list;
    }
}
```

测试结果：



7.6 整合log4j

以上两步操作如果前面已经完成的就不用再次操作！

7.6.1、在pom.xml文件中，引入log4j及其他依赖包

```
    <!-- 整合log4j -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
      <version>1.7.36</version>
    </dependency>
    <!-- Jackson Json处理工具包 -->
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.13.3</version>
    </dependency>
    <!-- Servlet/JSP/JSTL -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>servlet-api</artifactId>
      <version>2.4</version>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jsp-api</artifactId>
      <version>2.0</version>
    </dependency>
```

7.6.2、在resources目录下创建log4j.properties文件，配置内容如下:

```
log4j.rootLogger=INFO,A1
log4j.logger.cn.yhmis.mapper =DEBUG
log4j.appender.A1=org.apache.log4j.ConsoleAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss,SSS} [%t]
[%c]-[%p] %m%n
```

至此，OA办公系统的SSM整合以及EasyUI内容第一步就已经完成。接下来就是**页面以及后端的增删改查功能。**

三、系统业务功能代码

1.查询用户代码

dao层由于上面代码以及完成,故不需要再次完成。

service业务层

```
package com.tedu.service;

import com.tedu.pojo.User;
import java.util.List;
/**
 * UserService接口
 * 声明增删改查方法,对用户信息进行操作
 */
public interface UserService {
```

```
    /**
     * 1.查询所有用户信息
     */
    public List<User> findAll();
}
```

service业务层实现类

```
package com.tedu.service.impl;

import com.tedu.dao.UserDao;
import com.tedu.pojo.User;
import com.tedu.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserDao userDao;
    @Override
    public List<User> findAll() {
        return userDao.findAll();
    }
}
```

controller控制器层

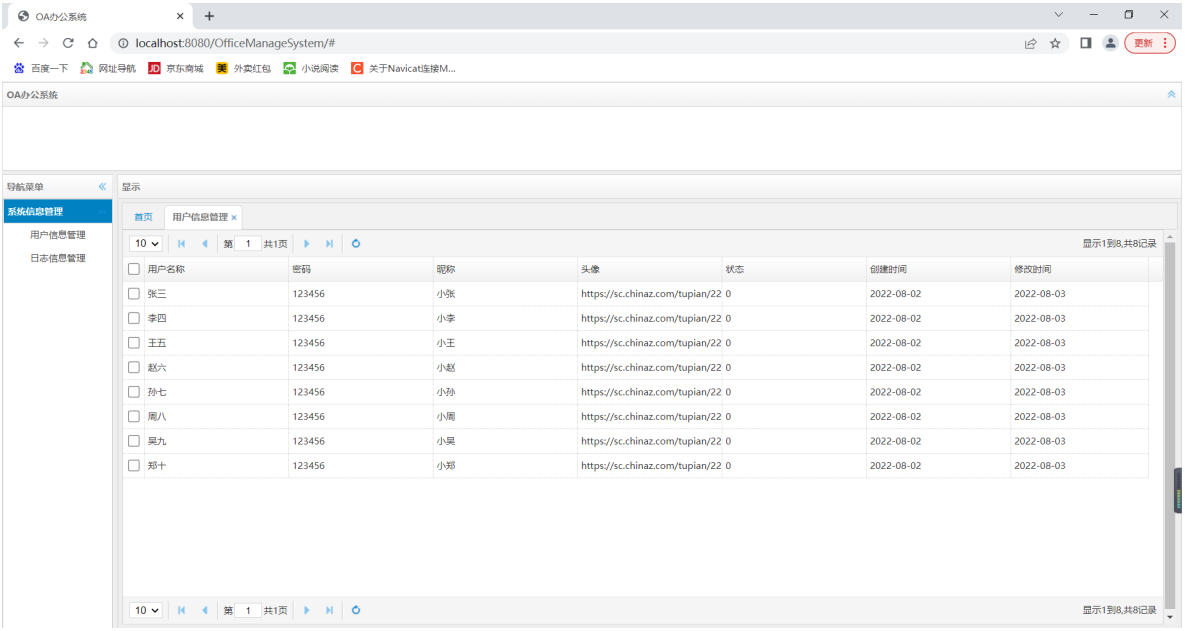
```
package com.tedu.controller;

import com.tedu.pojo.User;
import com.tedu.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;

    @RequestMapping("/findAll")
    public List<User> findAll(){
        //1.调用findAll方法查询所有用户信息
        List<User> list = userService.findAll();
        return list;
```

```
    }

}
```

前端页面user.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>用户信息管理</title>
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="../jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="../jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="../jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="../jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="../jeasyui/easyui-lang-zh_CN.js"></script>

    <script>
       $(function () {
           //表格数据
           $('#dg').datagrid({
               url: '../user/findAll',//远程请求地址
               fit: true,//适应父容器
               fitColumns: true,//自适应
               toolbar: '#tb',//表格的工具栏
               pagination: true,//分页工具栏
               pagePosition: 'both',//分页工具栏的位置 both 上下
               columns:[[//列
                   {field:'uid',title:'xxx',width:80,checkbox:true},
                   {field:'uname',title:'用户名称',width:80},
                   {field:'password',title:'密码',width:80},
                   {field:'nickName',title:'昵称',width:80},
                   {field:'images',title:'头像',width:80},
                   {field:'stat',title:'状态',width:80},
                   {field:'createTime',title:'创建时间',width:80},
                   {field:'modifyTime',title:'修改时间',width:80},
               ]]
           })
       })

    </script>

</head>
<body>
    <table id="dg"></table>
</body>
</html>
```

访问效果：



工具栏 增删改按钮 Linkbutton 链接按钮

```
<!-- user.html -->
<body>
    <!--表格工具栏链接-->
    <div id="tb">
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
add',plain:true">添加</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
edit',plain:true">修改</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
remove',plain:true">删除</a>
    </div>
    <table id="dg"></table>
</body>
```

工具栏 查询框

```
<!--查询框-->
        <table>
            <tr>
                <td>
                    <input id="uname" class="easyui-validatebox" placeholder="请
输入用户名称查询">
                    <a href="#" class="easyui-linkbutton" data-
options="iconCls:'icon-search',plain:true">查询</a>
                </td>
            </tr>
        </table>
```

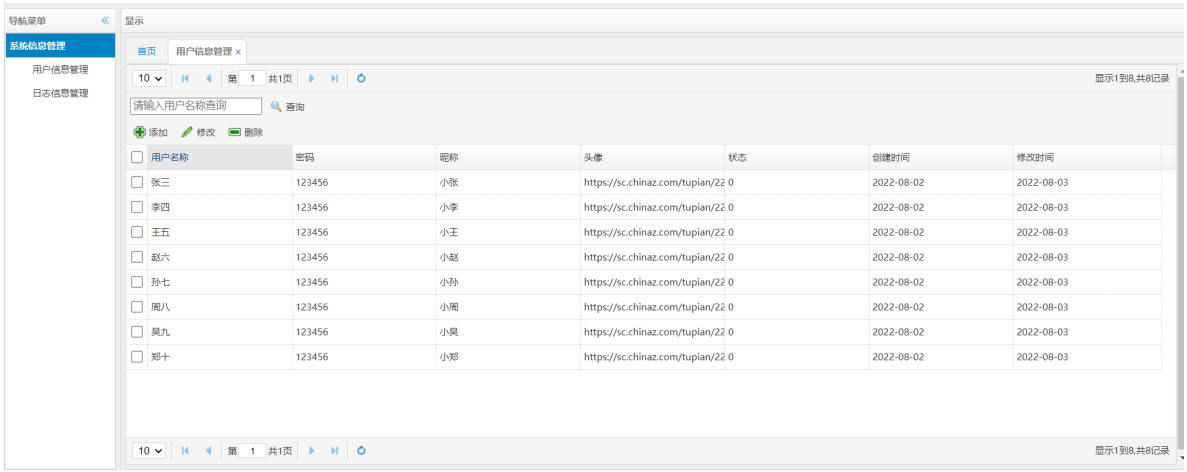
完整代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>用户信息管理</title>
```

```
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="../jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="../jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="../jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="../jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="../jeasyui/easyui-lang-zh_CN.js"></script>

    <script>
       $(function () {
           //表格数据
           $('#dg').datagrid({
               url: '../user/findAll',//远程请求地址
               fit: true,//适应父容器
               fitColumns: true,//自适应
               toolbar: '#tb',//表格的工具栏
               pagination: true,//分页工具栏
               pagePosition: 'both',//分页工具栏的位置 both 上下
               columns:[[//列
                   {field:'uid',title:'xxx',width:80,checkbox:true},
                   {field:'uname',title:'用户名称',width:80},
                   {field:'password',title:'密码',width:80},
                   {field:'nickName',title:'昵称',width:80},
                   {field:'images',title:'头像',width:80},
                   {field:'stat',title:'状态',width:80},
                   {field:'createTime',title:'创建时间',width:80},
                   {field:'modifyTime',title:'修改时间',width:80},
               ]]
           })
       })

    </script>

</head>
<body>
    <!--表格工具栏链接-->
    <div id="tb">
        <!--查询框-->
        <table>
            <tr>
                <td>
                    <input id="uname" class="easyui-validatebox" placeholder="请
输入用户名称查询">
                    <a href="#" class="easyui-linkbutton" data-
options="iconCls:'icon-search',plain:true">查询</a>
                </td>
            </tr>
        </table>
        <!--增删改按钮-->
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
add',plain:true">添加</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
edit',plain:true">修改</a>
```

```
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
remove',plain:true">删除</a>
    </div>
    <table id="dg"></table>
</body>
</html>
```

效果如下:



分页查询以及按照名称查询

1.UserMapper.xml

```
<!-- 1.查询所有用户信息，id值为对应接口中方法的名字
        resultType指定将查询的结果封装到哪个pojo对象中
     -->
    <select id="findAll" resultMap="userMapper">
        select * from tb_user
        <where>
            <if test="uname!=null and uname!=''">
                uname like '%${uname}%'
            </if>
            AND  deleted = 0
        </where>
        limit #{page},#{rows}
    </select>
    <!--分页统计-->
    <select id="count" resultType="java.lang.Integer">
        select count(*)
        from tb_user
        where deleted = 0
    </select>
```

2.UserDao接口

```
package com.tedu.dao;

import com.tedu.pojo.User;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;

import java.util.List;
```

```
/**
 * UserDao接口
 * 声明增删改查方法,对用户信息进行操作
 */
@Repository
public interface UserDao {
    /**
     * 1.查询所有用户信息
     */
    public List<User> findAll(@Param("uname")String uname,
                              @Param("page") Integer page,
                              @Param("rows") Integer rows);
    /**
     * 2.统计所有用户信息数量
     */
    public int count(@Param("uname")String uname);

}
```

3.UserService接口

```
package com.tedu.service;

import com.tedu.pojo.User;
import org.apache.ibatis.annotations.Param;

import java.util.List;
/**
 * UserService接口
 * 声明增删改查方法,对用户信息进行操作
 */
public interface UserService {
    /**
     * 1.查询所有用户信息
     */
    public List<User> findAll(String uname,Integer page,Integer rows);
    /**
     * 2.统计所有用户信息数量
     */
    public int count(@Param("uname")String uname);
}
```

4.UserServiceImpl实现类

```
package com.tedu.service.impl;

import com.tedu.dao.UserDao;
import com.tedu.pojo.User;
import com.tedu.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
```

```
@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserDao userDao;
    @Override
    public List<User> findAll(String uname,Integer page,Integer rows) {
        return userDao.findAll(uname,(page-1)*rows,rows);
    }

    @Override
    public int count(String uname) {
        return userDao.count(uname);
    }
}
```

5.UserController.java

```
package com.tedu.controller;

import com.tedu.pojo.User;
import com.tedu.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@RestController
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;

    @RequestMapping("/findAll")
    public Map<String,Object> findAll(String uname, Integer page, Integer rows){
        Map<String,Object>  map =  new HashMap<>();
        map.put("rows",userService.findAll(uname,page,rows));
        map.put("total",userService.count(uname));

        return map;
    }

}
```

6.user.html

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <title>用户信息管理</title>
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="../jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="../jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="../jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="../jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="../jeasyui/easyui-lang-zh_CN.js"></script>

    <script>
       $(function () {
           /*查询框*/
           $("#btn-search").click(function () {
               $('#dg').datagrid('load', {
                   uname: $("#uname").val()
               })
           })
           //表格数据
           $('#dg').datagrid({
               url: '../user/findAll',//远程请求地址
               fit: true,//适应父容器
               fitColumns: true,//自适应
               toolbar: '#tb',//表格的工具栏
               pagination: true,//分页工具栏
               pagePosition: 'both',//分页工具栏的位置 both 上下
               columns:[[//列
                   {field:'uid',title:'xxx',width:80,checkbox:true},
                   {field:'uname',title:'用户名称',width:80},
                   {field:'password',title:'密码',width:80},
                   {field:'nickName',title:'昵称',width:80},
                   {field:'images',title:'头像',width:80},
                   {field:'stat',title:'状态',width:80},
                   {field:'createTime',title:'创建时间',width:80},
                   {field:'modifyTime',title:'修改时间',width:80},
               ]]
           })
       })

    </script>

</head>
<body>
    <!--表格工具栏链接-->
    <div id="tb">
        <!--查询框-->
        <table>
            <tr>
                <td>
                    <input id="uname" class="easyui-validatebox" placeholder="请
输入用户名称查询">
                    <a href="#" id="btn-search" class="easyui-linkbutton" data-
options="iconCls:'icon-search',plain:true">查询</a>
                </td>
```

```
            </tr>
        </table>
        <!--增删改按钮-->
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
add',plain:true">添加</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
edit',plain:true">修改</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
remove',plain:true">删除</a>
    </div>
    <table id="dg"></table>
</body>
</html>
```

2.添加用户代码

UserDao.java

```
/**
     * 3.添加用户
     */
    int save(User user);
```

UserMapper.xml

```
 <insert id="save">
        insert into tb_user(uname,nickname,password,images,createtime)
        values (#{uname},#{nickName},#{password},#{images},now())
    </insert>
```

UserService.java

```
 /**
     * 3.添加用户
     */
    boolean save(User user);
```

UserServiceImpl.java

```
 @Override
    public boolean save(User user) {
        return userDao.save(user)>0;
    }
```

UserController.java

```
 @RequestMapping("/save")
    public boolean save(User user){
        return  userService.save(user);
    }
```

user.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>用户信息管理</title>
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="../jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="../jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="../jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="../jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="../jeasyui/easyui-lang-zh_CN.js"></script>

    <script>
       $(function () {
           //添加
           $("#btn-add").click(function () {
               $("#add-dialog").dialog({
                   closed:false,
                   buttons:[{
                       text:'保存',
                       iconCls:'icon-save',
                       handler:function () {
                           $("#add-form").form("submit",{
                               url:"../user/save",
                               onSubmit:function () {
                                   return $("#add-form").form("validate");
                               },
                               success:function (flag) {
                                   if (flag){
                                       $("#dg").datagrid("load");
                                       $("#add-dialog").dialog({closed:true});
                                       $.messager.alert("消息","添加成
功！","info")
                                   }
                               }

                           })
                       }
                   },{
                       text:'重置',
                       iconCls:'icon-redo',
                       handler:function () {
                           $("#add-form").form('clear');
                       }
                   }]
               })
           })

           /*查询框*/
           $("#btn-search").click(function () {
               $('#dg').datagrid('load', {
                   uname: $("#uname").val()
               })
```

```
            })
            //表格数据
            $('#dg').datagrid({
                url: '../user/findAll',//远程请求地址
                fit: true,//适应父容器
                fitColumns: true,//自适应
                toolbar: '#tb',//表格的工具栏
                pagination: true,//分页工具栏
                pagePosition: 'both',//分页工具栏的位置 both 上下
                columns:[[//列
                    {field:'uid',title:'xxx',width:80,checkbox:true},
                    {field:'uname',title:'用户名称',width:80},
                    {field:'password',title:'密码',width:80},
                    {field:'nickName',title:'昵称',width:80},
                    {field:'images',title:'头像',width:80},
                    {field:'stat',title:'状态',width:80},
                    {field:'createTime',title:'创建时间',width:80},
                    {field:'modifyTime',title:'修改时间',width:80},
                ]]
            })
        })

    </script>

</head>
<body>
    <!--表格工具栏链接-->
    <div id="tb">
        <!--查询框-->
        <table>
            <tr>
                <td>
                    <input id="uname" class="easyui-validatebox" placeholder="请
输入用户名称查询">
                    <a href="#" id="btn-search" class="easyui-linkbutton" data-
options="iconCls:'icon-search',plain:true">查询</a>
                </td>
            </tr>
        </table>
        <!--增删改按钮-->
        <a id="btn-add" href="#" class="easyui-linkbutton" data-
options="iconCls:'icon-add',plain:true">添加</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
edit',plain:true">修改</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-
remove',plain:true">删除</a>
    </div>
    <table id="dg"></table>

    <div id="add-dialog" class="easyui-dialog" closed="true" title="添加用户"
style="width: auto;height: 200px">
        <form id="add-form" method="post">
            <table>
                <tr>
                    <td>用户名：</td>
                    <td>
                        <input name="uname" class="easyui-validatebox"
placeholder="请输入用户名称"
```

```
                                data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>密码：</td>
                    <td>
                        <input name="password" class="easyui-validatebox"
placeholder="请输入密码"
                                data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>昵称：</td>
                    <td>
                        <input name="nickName" class="easyui-validatebox"
placeholder="请输入昵称"
                                data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>头像</td>
                    <td>
                        <input name="images" class="easyui-validatebox"
placeholder="请输入头像"
                                data-options="required:true">
                    </td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>
```

3.删除用户代码

UserDao.java

```
/**
     * 4.批量删除用户
     */
    int remove(List<Integer> list);
```

UserMapper.xml

```
<update id="remove">
        update tb_user set deleted=1 where uid in
        <foreach item="item" index="index" collection="list"
                 open="(" separator="," close=")">
            #{item}
        </foreach>
    </update>
```

UserService.java

```
 /**
     * 4.批量删除用户
     */
    boolean batchRemove(List<Integer> list);
```

UserServiceImpl.java

```
@Override
    public boolean batchRemove(List<Integer> list) {
        return userDao.remove(list)>0;
    }
```

UserController.java

```
 @RequestMapping("/batchRemove")
    public boolean  batchRemove(@RequestParam("uids[]")List<Integer> list){
        return  userService.batchRemove(list);
    }
```

user.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>用户信息管理</title>
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="../jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="../jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="../jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="../jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="../jeasyui/easyui-lang-zh_CN.js"></script>

    <script>
       $(function () {
           //批量删除
           $('#btn-remove').click(function () {
               let arr = $("#dg").datagrid("getSelections");
               if(arr.length==0){
                   $.messager.alert('消息','请选择要删除的数据','error');
               }else{
                   let uids=[];
                   for(let i in arr){
                       uids.push(arr[i].uid);
                   }
                   $.post('../user/batchRemove',{"uids[]":uids},function (flag)
{
                       if(flag){
                           $.messager.alert("消息","删除数据成功！",'info');
                           $('#dg').datagrid('load');
```

```
                    }
                })
            }
        })
        //添加
        $("#btn-add").click(function () {
            $("#add-dialog").dialog({
                closed:false,
                buttons:[{
                    text:'保存',
                    iconCls:'icon-save',
                    handler:function () {
                        $("#add-form").form("submit",{
                            url:"../user/save",
                            onSubmit:function () {
                                return $("#add-form").form("validate");
                            },
                            success:function (flag) {
                                if (flag){
                                    $("#dg").datagrid("load");
                                    $("#add-dialog").dialog({closed:true});
                                    $.messager.alert("消息","添加成
功！","info")
                                }
                            }

                        })
                    }
                },{
                    text:'重置',
                    iconCls:'icon-redo',
                    handler:function () {
                        $("#add-form").form('clear');
                    }
                }]
            })
        })

        /*查询框*/
        $("#btn-search").click(function () {
            $('#dg').datagrid('load', {
                uname: $("#uname").val()
            })
        })
        //表格数据
        $('#dg').datagrid({
            url: '../user/findAll',//远程请求地址
            fit: true,//适应父容器
            fitColumns: true,//自适应
            toolbar: '#tb',//表格的工具栏
            pagination: true,//分页工具栏
            pagePosition: 'both',//分页工具栏的位置 both 上下
            columns:[[//列
                {field:'uid',title:'xxx',width:80,checkbox:true},
                {field:'uname',title:'用户名称',width:80},
                {field:'password',title:'密码',width:80},
                {field:'nickName',title:'昵称',width:80},
                {field:'images',title:'头像',width:80},
```

```
                    {field:'stat',title:'状态',width:80},
                    {field:'createTime',title:'创建时间',width:80},
                    {field:'modifyTime',title:'修改时间',width:80},
                ]]
            })
        })

    </script>

</head>
<body>
    <!--表格工具栏链接-->
    <div id="tb">
        <!--查询框-->
        <table>
            <tr>
                <td>
                    <input id="uname" class="easyui-validatebox" placeholder="请输入用户名称查询">
                    <a href="#" id="btn-search" class="easyui-linkbutton" data-options="iconCls:'icon-search',plain:true">查询</a>
                </td>
            </tr>
        </table>
        <!--增删改按钮-->
        <a id="btn-add" href="#" class="easyui-linkbutton" data-options="iconCls:'icon-add',plain:true">添加</a>
        <a href="#" class="easyui-linkbutton" data-options="iconCls:'icon-edit',plain:true">修改</a>
        <a id="btn-remove" href="#" class="easyui-linkbutton" data-options="iconCls:'icon-remove',plain:true">删除</a>
    </div>
    <table id="dg"></table>

    <div id="add-dialog" class="easyui-dialog" closed="true" title="添加用户" style="width: auto;height: 200px">
        <form id="add-form" method="post">
            <table>
                <tr>
                    <td>用户名：</td>
                    <td>
                        <input name="uname" class="easyui-validatebox" placeholder="请输入用户名称"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>密码：</td>
                    <td>
                        <input name="password" class="easyui-validatebox" placeholder="请输入密码"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>昵称：</td>
                    <td>
```

```
                        <input name="nickName" class="easyui-validatebox"
placeholder="请输入昵称"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>头像</td>
                    <td>
                        <input name="images" class="easyui-validatebox"
placeholder="请输入头像"
                               data-options="required:true">
                    </td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>
```

4.修改用户代码

UserDao.java

```
    /**
     * 5.修改用户
     */
    int edit(User user);
```

UserMapper.xml

```
    <update id="edit">
        update tb_user set
        uname=#{uname},nickname=#{nickName},password=#{password},images=#
{images},modifyTime=now()
        where uid=#{uid}
    </update>
```

UserService.java

```
  /**
     * 5.修改用户
     */
    boolean edit(User user);
```

UserServiceImpl.java

```
  @Override
    public boolean edit(User user) {
        return userDao.edit(user)>0;
    }
```

UserController.java

```
@RequestMapping("/edit")
    public boolean edit(User user){
        return  userService.edit(user);
    }
```

user.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>用户信息管理</title>
    <!--    皮肤样式-->
    <link rel="stylesheet" type="text/css"
href="../jeasyui/themes/bootstrap/easyui.css">
    <!--    图片样式-->
    <link rel="stylesheet" type="text/css" href="../jeasyui/themes/icon.css">
    <!--    jquery环境-->
    <script src="../jeasyui/jquery.min.js"></script>
    <!--    easyui的js环境-->
    <script src="../jeasyui/jquery.easyui.min.js"></script>
    <!--    汉化js-->
    <script src="../jeasyui/easyui-lang-zh_CN.js"></script>

    <script>
       $(function () {
           //修改
           $("#btn-edit").click(function () {
               let arr=$("#dg").datagrid('getSelections');
               if(arr.length==0){
                   $.messager.alert("消息","请选择要修改的数据！","error");
               }else if(arr.length>1){
                   $.messager.alert("消息","请选择一条记录进行修改！","error");
               }else{
                   //填充修改表单数据
                   $("#edit-form").form('load',{
                       uid:arr[0].uid,
                       uname:arr[0].uname,
                       password:arr[0].password,
                       images:arr[0].images,
                       nickName:arr[0].nickName
                   })
                   //弹出修改对话框
                   $("#edit-dialog").dialog({
                       closed:false,
                       buttons:[{
                           text:'保存',
                           iconCls:'icon-save',
                           handler:function () {
                               $("#edit-form").form("submit",{
                                   url:"../user/edit",
                                   onSubmit:function () {
                                       return $("#edit-form").form("validate");
                                   },
                                   success:function (flag) {
                                       if (flag) {
```

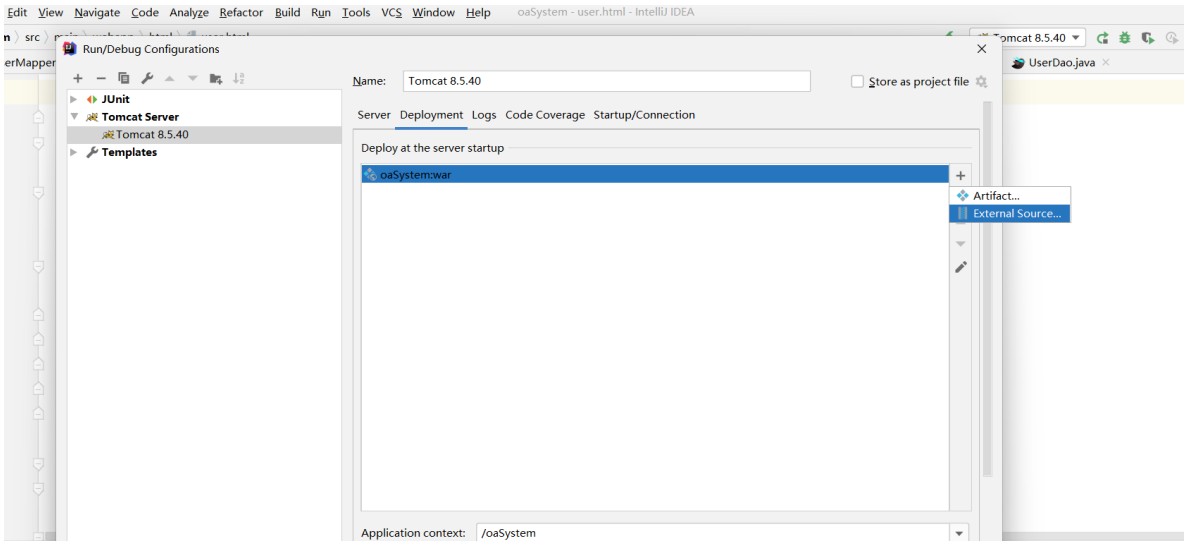
```
                                        //表格刷新
                                        $("#dg").datagrid("load");
                                        //修改对话框关闭
                                        $("#edit-dialog").dialog({closed:
true});
                                        $.messager.alert("消息", "修改成功！",
"info")
                                    }
                                }
                            })
                        }
                    },{
                        text:'重置',
                        iconCls:'icon-redo',
                        handler:function () {
                            $("#edit-form").form('clear');
                        }
                    }]
                })
            }
        })
        //批量删除
        $('#btn-remove').click(function () {
            let arr = $("#dg").datagrid("getSelections");
            if(arr.length==0){
                $.messager.alert('消息','请选择要删除的数据','error');
            }else{
                let uids=[];
                for(let i in arr){
                    uids.push(arr[i].uid);
                }
                $.post('../user/batchRemove',{"uids[]":uids},function (flag)
{
                    if(flag){
                        $.messager.alert("消息","删除数据成功！",'info');
                        $('#dg').datagrid('load');
                    }
                })
            }
        })
        //添加
        $("#btn-add").click(function () {
            $("#add-dialog").dialog({
                closed:false,
                buttons:[{
                    text:'保存',
                    iconCls:'icon-save',
                    handler:function () {
                        $("#add-form").form("submit",{
                            url:"../user/save",
                            onSubmit:function () {
                                return $("#add-form").form("validate");
                            },
                            success:function (flag) {
                                if (flag){
                                    $("#dg").datagrid("load");
                                    $("#add-dialog").dialog({closed:true});
```

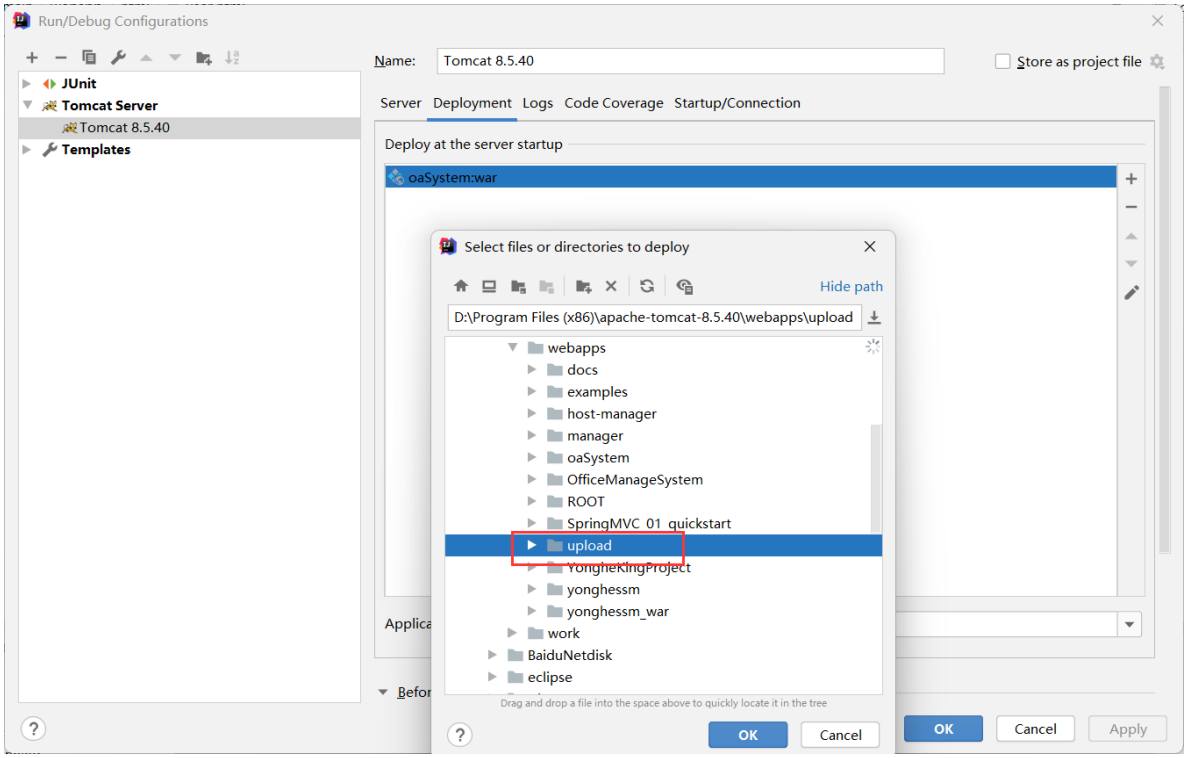
```
                                    $.messager.alert("消息","添加成功! ","info")
                                }
                            }

                        })
                    }
                },{
                    text:'重置',
                    iconCls:'icon-redo',
                    handler:function () {
                        $("#add-form").form('clear');
                    }
                }]
            })
        })

        /*查询框*/
        $("#btn-search").click(function () {
            $('#dg').datagrid('load', {
                uname: $("#uname").val()
            })
        })
        //表格数据
        $('#dg').datagrid({
            url: '../user/findAll',//远程请求地址
            fit: true,//适应父容器
            fitColumns: true,//自适应
            toolbar: '#tb',//表格的工具栏
            pagination: true,//分页工具栏
            pagePosition: 'both',//分页工具栏的位置 both 上下
            columns:[[//列
                {field:'uid',title:'xxx',width:80,checkbox:true},
                {field:'uname',title:'用户名称',width:80},
                {field:'password',title:'密码',width:80},
                {field:'nickName',title:'昵称',width:80},
                {field:'images',title:'头像',width:80},
                {field:'stat',title:'状态',width:80},
                {field:'createTime',title:'创建时间',width:80},
                {field:'modifyTime',title:'修改时间',width:80},
            ]]
        })
    })

</script>

</head>
<body>
    <!--表格工具栏链接-->
    <div id="tb">
        <!--查询框-->
        <table>
            <tr>
                <td>
                    <input id="uname" class="easyui-validatebox" placeholder="请输入用户名称查询">
                    <a href="#" id="btn-search" class="easyui-linkbutton" data-options="iconCls:'icon-search',plain:true">查询</a>
```

```
                </td>
            </tr>
        </table>
        <!--增删改按钮-->
        <a id="btn-add" href="#" class="easyui-linkbutton" data-
options="iconCls:'icon-add',plain:true">添加</a>
        <a id="btn-edit" href="#" class="easyui-linkbutton" data-
options="iconCls:'icon-edit',plain:true">修改</a>
        <a id="btn-remove" href="#" class="easyui-linkbutton" data-
options="iconCls:'icon-remove',plain:true">删除</a>
    </div>
    <table id="dg"></table>

    <div id="add-dialog" class="easyui-dialog" closed="true" title="添加用户"
style="width: auto;height: 200px">
        <form id="add-form" method="post">
            <table>
                <tr>
                    <td>用户名：</td>
                    <td>
                        <input name="uname" class="easyui-validatebox"
placeholder="请输入用户名称"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>密码：</td>
                    <td>
                        <input name="password" class="easyui-validatebox"
placeholder="请输入密码"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>昵称：</td>
                    <td>
                        <input name="nickName" class="easyui-validatebox"
placeholder="请输入昵称"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>头像</td>
                    <td>
                        <input name="images" class="easyui-validatebox"
placeholder="请输入头像"
                               data-options="required:true">
                    </td>
                </tr>
            </table>
        </form>
    </div>

    <div id="edit-dialog" class="easyui-dialog" closed="true" title="修改用户"
style="width: auto;height: 200px">
        <form id="edit-form" method="post">
            <!--修改的条件-->
            <input type="hidden" name="uid">
```

```
            <table>
                <tr>
                    <td>用户名：</td>
                    <td>
                        <input name="uname" class="easyui-validatebox"
placeholder="请输入用户名称"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>密码：</td>
                    <td>
                        <input name="password" class="easyui-validatebox"
placeholder="请输入密码"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>昵称：</td>
                    <td>
                        <input name="nickName" class="easyui-validatebox"
placeholder="请输入昵称"
                               data-options="required:true">
                    </td>
                </tr>
                <tr>
                    <td>头像</td>
                    <td>
                        <input name="images" class="easyui-validatebox"
placeholder="请输入头像"
                               data-options="required:true">
                    </td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>
```

5.头像上传





user.html

```
<!--头像上传-->
    <div id="upload-dialog" data-options="closed:true" title="头像上传"
class="easyui-dialog" style="width: auto;height: auto;">
        <form method="post" id="upload-form"   enctype="multipart/form-data">
            <input class="easyui-filebox"  data-options="required:true"
name="file" style="width:300px">
        </form>
    </div>
```

user.html 的上传js

```
 //头像上传
$("#btn-upload").click(function () {
    let arr = $("#dg").datagrid('getSelections');
    if(arr.length==0){
        $.messager.alert("消息","请选择上传头像的用户信息！","error");
    }else if(arr.length>1){
        $.messager.alert("消息","请选择一条记录进行头像上传！","error");
    }else{
        $("#upload-dialog").dialog({
            closed:false,
            buttons:[{
                text:'保存',
                iconCls:'icon-save',
                handler:function () {
                    $("#upload-form").form("submit",{
                        url:"../user/uploadImages?uid="+arr[0].uid,
                        onSubmit:function () {
                            return $("#upload-form").form("validate");
                        },
                        success:function (flag) {
                            if (flag) {
                                //表格刷新
```

```
                                $("#dg").datagrid("load");
                                $("#upload-dialog").dialog({
                                    closed:true
                                })
                                $.messager.alert("消息", "头像上传成功! ", "info")
                            }
                        }
                    })
                }
            }]
        })
    }
})
```

pomxml

```
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
    </dependency>
    <!--文件上传-->
    <dependency>
      <groupId>commons-fileupload</groupId>
      <artifactId>commons-fileupload</artifactId>
      <version>1.3.3</version>
    </dependency>
```

springmvc-config.xml

```
<!--配置文件上传-->
    <bean id="multipartResolver"

 class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
        <!--10mb-->
        <property name="maxUploadSize" value="10485760"></property>
        <!--解决上传文件过程中的中文乱码问题-->
        <property name="defaultEncoding" value="utf-8"></property>
    </bean>
```

UserDao.java

```
/**
     * 6.根据id查询用户
     */
    User getUserById(Integer uid);
```

UserMapper.xml

```
<select id="getUserById" resultType="com.tedu.pojo.User">
        select * from tb_user where uid = #{uid}
    </select>
```

UserService.java

```
/**
     * 6.根据id查询用户
     */
    User getUserById(Integer uid);
```

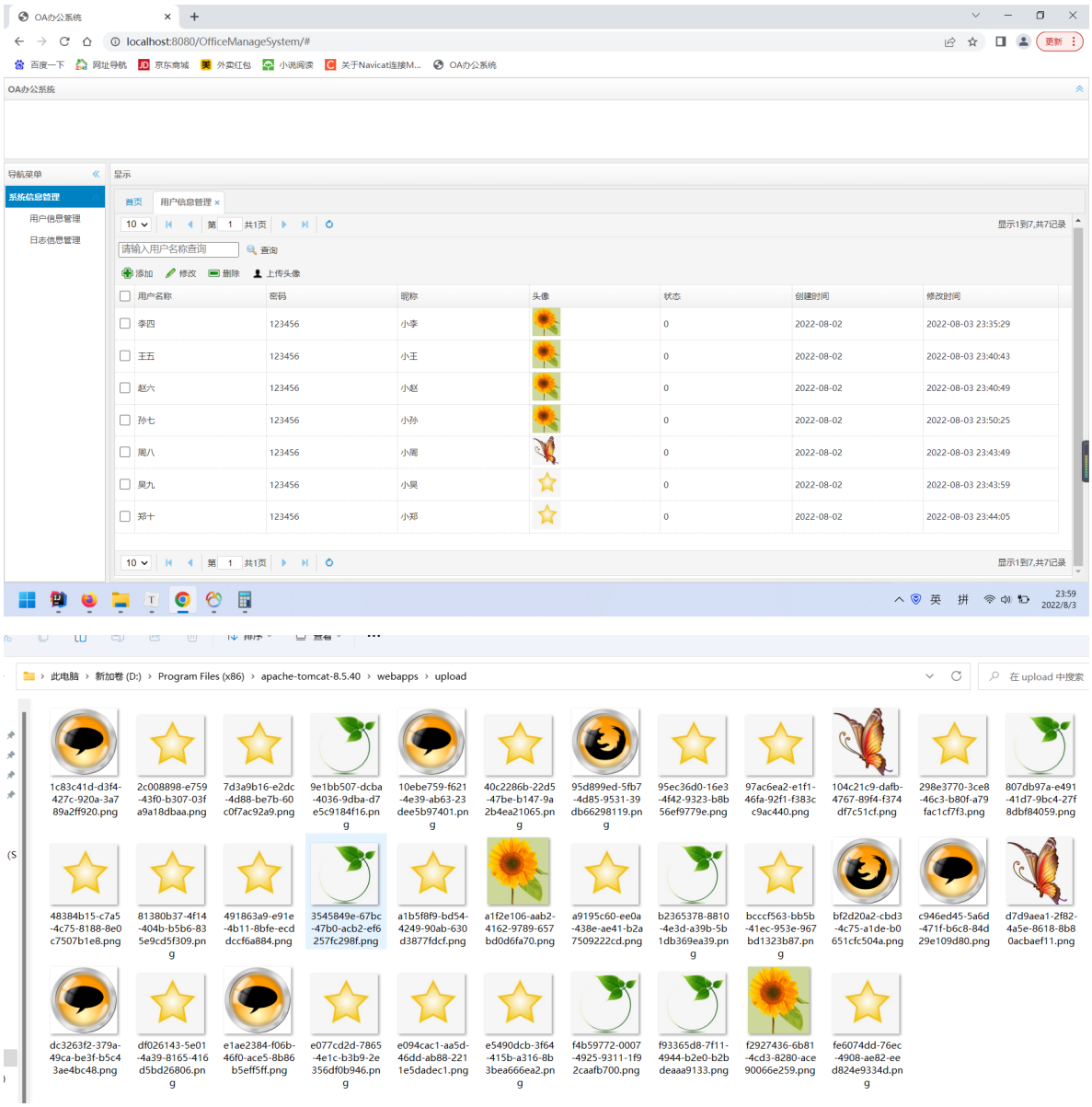
UserServiceImpl.java

```
@Override
    public User getUserById(Integer uid) {
        return userDao.getUserById(uid);
    }
```

UserController.java

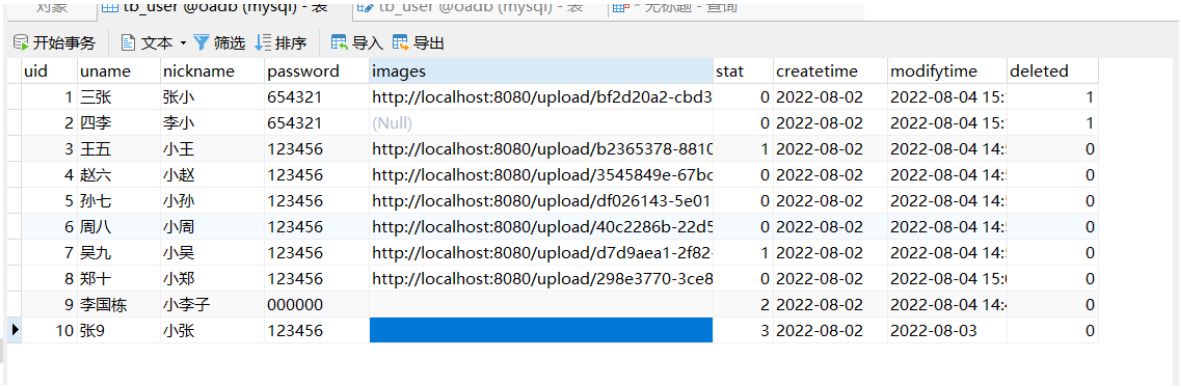
```
@RequestMapping("/uploadImages")
    public boolean uploadImages(Integer uid, MultipartFile file,
                                HttpServletRequest request) throws IOException {

        //获得上传文件名称，判断后缀是否.png名称，不是就不执行后面上传代码
        String originalFilename = file.getOriginalFilename();

        if (!originalFilename.endsWith(".png")) {
            return false;
        }
        //查询需要上传的对象的数据
        User user =userService.getUserById(uid);
        //获取当前项目部署的阵势路径
        String realPath = request.getServletContext().getRealPath("/");
        //截取除项目名以外的路径
        String substring = realPath.substring(0,
realPath.indexOf("OfficeManageSystem"));

        //文件上传路径 （需要在当前tomcat的webapps下创建新文件加upload）
        String uploadPath = substring + "upload\\";
        //上传文件的新名称
        String newFileName = UUID.randomUUID().toString()+".png";
        //创建文件
        File filePath = new File(uploadPath, newFileName);

        // 如果目录不存在则创建目录
        if(!filePath.getParentFile().exists()) {
            filePath.getParentFile().mkdirs();
            System.out.println("创建目录: " + filePath);
        }
        //真正的文件上传
        file.transferTo(filePath);
        //由于要请求所以需要加上请求地址
        String url = "http://localhost:8080/upload/"+newFileName;
        //将用户的头像数据设置进去
        user.setImages(url);
        //调用修改方法刷新头像
        return  userService.edit(user);
    }
```

效果如下：



6.用户状态

```
{field:'stat',title:'状态',width:80,
    formatter:function (stat, rows, index) {
        if (stat != null) {
            if(stat==0){
                return "<div style='font-weight: bold;color: green'>启用</div>"
            }else if(stat==1){
                return "<div style='font-weight: bold;color: red'>禁用</div>"
            }else{
                return "<div style='font-weight: bold;color: yellow'>未知</div>"
            }
        }
    }
},
```



开始事务 文本 筛选 排序 导入 导出

uid	uname	nickname	password	images	stat	createtime	modifytime	deleted
1	三张	张小	654321	http://localhost:8080/upload/bf2d20a2-cbd3	0	2022-08-02	2022-08-04 15:	1
2	四李	李小	654321	(Null)	0	2022-08-02	2022-08-04 15:	1
3	王五	小王	123456	http://localhost:8080/upload/b2365378-881(	1	2022-08-02	2022-08-04 14:	0
4	赵六	小赵	123456	http://localhost:8080/upload/3545849e-67bc	0	2022-08-02	2022-08-04 14:	0
5	孙七	小孙	123456	http://localhost:8080/upload/df026143-5e01	0	2022-08-02	2022-08-04 14:	0
6	周八	小周	123456	http://localhost:8080/upload/40c2286b-22d5	0	2022-08-02	2022-08-04 14:	0
7	吴九	小吴	123456	http://localhost:8080/upload/d7d9aea1-2f82	1	2022-08-02	2022-08-04 14:	0
8	郑十	小郑	123456	http://localhost:8080/upload/298e3770-3ce8	0	2022-08-02	2022-08-04 15:	0
9	李国栋	小李子	000000		2	2022-08-02	2022-08-04 14:	0
10	张9	小张	123456		3	2022-08-02	2022-08-03	0

请输入用户名称查询 查询

添加 修改 删除 上传头像

用户名称	密码	昵称	头像	状态	创建时间
王五	123456	小王		禁用	2022-08-02
赵六	123456	小赵		启用	2022-08-02
孙七	123456	小孙		启用	2022-08-02
周八	123456	小周		启用	2022-08-02
吴九	123456	小吴		禁用	2022-08-02
郑十	123456	小郑		启用	2022-08-02
李国栋	000000	小李子		未知	2022-08-02
张9	123456	小张		启用	2022-08-02