目录

[一．创建ORACLE数据库 2](#_Toc271186874)

[二．Myeclipse下新建工程SVI 2](#_Toc271186875)

[三．配置ibatis 4](#_Toc271186876)

[四．配置web.xml 5](#_Toc271186877)

[五 配置VSITest-servlet.xml 5](#_Toc271186878)

[六 对应实例代码 11](#_Toc271186879)

[七.创建velocity模板 25](#_Toc271186880)

[八 .测试 28](#_Toc271186881)

# 一．创建ORACLE数据库

数据库名为ORCL，并创建表T\_USER

CREATE TABLE T\_USER

(

USERNAME VARCHAR2(20) NOT NULL,

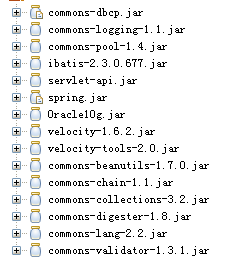
PASSWORD VARCHAR2(20) NOT NULL

)

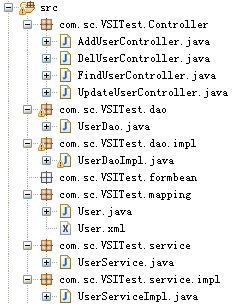
插入些数据

# 二．Myeclipse下新建工程SVI

用myeclipse新建一个web工程，导入所需的jar包，包括



包结构为



上图为本测试的WEB工程结构，可以看出WEB工程采用流行的三层框架，即MVC，这里我们采用的是Spring MVC，即表示层采用velocity，运用ibatis作为持久化对象，采用springIOC配置javabean的实际调用，以spring相关类作为控制器控制页面转发，核心是spring MVC

对应包说明如下：

com.sc.VSITest.dao------------------------------------系统DAO接口

com.sc.VSITest.dao.impl------------------------------------系统DAO实现

com.sc.VSITest..mapping------------------------------------bean文件及其映射文件

com.sc.VSITest.service------------------------------------系统服务层接口

com.sc.VSITest.service.impl------------------------------------系统服务实现

com.sc.VSITest.Controller------------------------------------系统控制层，为页面跳转提供支持

com.sc.VSITest.formbean------------------------------------系统传输对象目录

# 三．配置ibatis

1.在web-inf下配置ibatis总配置文件SqlMapConfig.xml

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

<!DOCTYPE sqlMapConfig PUBLIC "-//ibatis.apache.org//DTD SQL Map Config 2.0//EN"

"http://ibatis.apache.org/dtd/sql-map-config-2.dtd">

<sqlMapConfig>

<settings cacheModelsEnabled="true" enhancementEnabled="false"

lazyLoadingEnabled="false" maxRequests="3000" maxSessions="3000"

maxTransactions="300" useStatementNamespaces="true"/>

<!-- 配置ibatis -->

<sqlMap resource="com/sc/VSITest/mapping/User.xml"/>

</sqlMapConfig>

2.在web-inf下配置数据文件SqlMapForOracle.properties

driverName=oracle.jdbc.driver.OracleDriver

url=jdbc:oracle:thin:@localhost:1521:ORCL

username=qbstk

password=liusheng

# 四．配置web.xml

<!-- 1.配置Sring MVC的核心控制器DispatcherServlet

DispatcherServlet从一个xml文件中装入Spring应用程序上下文，xml文件的名称是servlet名称后面加上-sevlet，

在这个示例中,DispatcherServlet会从VSITest-servlet.xml文件装入应用程序上下文

-->

<servlet>

<servlet-name>VSITest</servlet-name>

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

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

</servlet>

<!--为VSITest建立映射 -->

<servlet-mapping>

<servlet-name>VSITest</servlet-name>

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

</servlet-mapping>

所有.do的请求都会转到DispatcherServlet

# 五 配置VSITest-servlet.xml

当前用的spring版本为2.0，可以考虑升级到2.5或3.0

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

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN" "http://www.springframework.org/dtd/spring-beans-2.0.dtd">

1. 数据源配置

<!-- =================================================================================== -->

<!-- 数据源配置 -->

<!-- =================================================================================== -->

<!-- DataSourse定义 -->

<bean id="propertyConfigure" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="locations">

<list>

<value>/WEB-INF/SqlMapForOracle.properties</value>

</list>

</property>

</bean>

<bean id="dataSource"

class="org.apache.commons.dbcp.BasicDataSource"

destroy-method="close">

<property name="driverClassName">

<value>${driverName}</value>

</property>

<property name="url">

<value>${url}</value>

</property>

<property name="username">

<value>${username}</value>

</property>

<property name="password">

<value>${password}</value>

</property>

</bean>

2. 配置ibatis

<!-- =================================================================================== -->

<!-- ibatis 配置 -->

<!-- =================================================================================== -->

<!-- 注入ibatis总配置文件和数据源 让它们给我生成 SqlMapClient-->

<bean id="sqlMapClient" class="org.springframework.orm.ibatis.SqlMapClientFactoryBean">

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

<property name="configLocation" value="WEB-INF/SqlMapConfig.xml"/>

</bean>

3.配置velocity

<!-- =================================================================================== -->

<!-- Velocity 配置 -->

<!-- =================================================================================== -->

<!-- 配置Velocity引擎

VelocityConfigurer负责在Spring中设置Velocity引擎，这里，我们通过属性resourceLoaderPath告诉Velocity到哪里寻找它的模板

建议将模板放到web-inf的某个子目录里，这样可以保证这些模板不能被直接访问

-->

<bean id="velocityConfigurer" class="org.springframework.web.servlet.view.velocity.VelocityConfigurer">

<!-- resourceLoaderPath属性通知Velocity到哪里找到velocity模板 -->

<property name="resourceLoaderPath" value="/WEB-INF/velocity/"/>

<!-- 处理中文 -->

<property name="velocityProperties">

<props>

<prop key="input.encoding">gbk</prop>

<prop key="output.encoding">gbk</prop>

</props>

</property>

</bean>

<!-- 配置velocity视图解析器 -->

<bean id="viewResolver" class="org.springframework.web.servlet.view.velocity.VelocityViewResolver">

<!-- 配置模板的后缀 -->

<property name="suffix">

<value>.vm</value>

</property>

<!-- 处理中文乱码 -->

<property name="contentType">

<value>text/html;charset=gbk</value>

</property>

</bean>

4.注入dao和service

<beans>

<!-- =================================================================================== -->

<!-- 系统DAO层注入 -->

<!-- =================================================================================== -->

<!-- dao定义 -->

<bean id="userDao" class="com.sc.VSITest.dao.impl.UserDaoImpl">

<!-- 为UserDaoImpl注入 SqlMapClient-->

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

</bean>

<!-- =================================================================================== -->

<!-- Service层注入 -->

<!-- =================================================================================== -->

<bean id="userService" class="com.sc.VSITest.service.impl.UserServiceImpl">

<!-- 注入 UserDao -->

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

</bean>

</beans>

5．配置Controller

<!-- =================================================================================== -->

<!-- Controller层注入 本层实现Spring MVC 核心控制-->

<!-- =================================================================================== -->

<!-- 配置方法名解析器 -->

<bean id="methodNameResolver" class="org.springframework.web.servlet.mvc.multiaction.PropertiesMethodNameResolver">

<property name="mappings">

<props>

<prop key="/addUser.do">addUser</prop>

<prop key="/addUserForm.do">addUserForm</prop>

<prop key="/updateUser.do">updateUser</prop>

<prop key="/updateUserForm.do">updateUserForm</prop>

<prop key="/findUser.do">findUserByName</prop>

<prop key="/listUser.do">listUser</prop>

</props>

</property>

</bean>

<!-- 配置控制器 url映射 -->

<bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">

<property name="mappings">

<props>

<prop key="/findUser.do">findUserController</prop>

<prop key="/listUser.do">findUserController</prop>

<prop key="/addUser.do">addUserController</prop>

<prop key="/addUserForm.do">addUserController</prop>

<prop key="/delUser.do">delUserController</prop>

<prop key="/updateUser.do">updateUserController</prop>

<prop key="/updateUserForm.do">updateUserController</prop>

</props>

</property>

</bean>

<!-- 查询用户控制器 -->

<bean id="findUserController" class="com.sc.VSITest.Controller.FindUserController">

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

<!-- 注入模板名称 -->

<property name="page\_listUser">

<value>listUser</value>

</property>

<!-- 注入 userService -->

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

</bean>

<!-- 增加用户控制器 -->

<bean id="addUserController" class="com.sc.VSITest.Controller.AddUserController">

<property name="methodNameResolver">

<ref bean="methodNameResolver"/>

</property>

<!-- 注入模板名称 -->

<property name="page\_listUser">

<value>listUser</value>

</property>

<property name="page\_addUser">

<value>addUser</value>

</property>

<!-- 注入 userService -->

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

</bean>

<!-- 删除用户控制器 -->

<bean id="delUserController" class="com.sc.VSITest.Controller.DelUserController">

<!-- 注入模板名称 -->

<property name="page\_listUser">

<value>listUser</value>

</property>

<!-- 注入 userService -->

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

</bean>

<!-- 修改用户控制器 -->

<bean id="updateUserController" class="com.sc.VSITest.Controller.UpdateUserController">

<property name="methodNameResolver">

<ref bean="methodNameResolver"/>

</property>

<!-- 注入模板名称 -->

<property name="page\_listUser">

<value>listUser</value>

</property>

<property name="page\_updateUser">

<value>updateUser</value>

</property>

<!-- 注入 userService -->

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

</bean>

# 六 对应实例代码

1.User.java

**package** com.sc.VSITest.mapping;

**public** **class** User {

**private** String username ;

**private** String password ;

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

}

2.User.xml, 与User.java在同一个目录下

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

<!DOCTYPE sqlMap

PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN"

"http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMap namespace="USER\_SPACE">

<typeAlias alias="User" type="com.sc.VSITest.mapping.User"/>

<!-- 负责将结果集的列值映射成 Java Bean的属性值 -->

<resultMap id="result\_user" class="com.sc.VSITest.mapping.User">

<result property="username" column="username"/>

<result property="password" column="password"/>

</resultMap>

<!-- 查询所有用户 -->

<select id="get\_user\_list" parameterClass="java.util.Map" resultMap="result\_user">

select username,password from t\_user

</select>

<!-- 按姓名查询 -->

<select id="find\_user\_by\_username" parameterClass="java.lang.String" resultMap="result\_user">

select username,password from t\_user where username=#username#

</select>

<!-- 添加一个用户 -->

<insert id="add\_user" parameterClass="User">

insert into t\_user values(#username#,#password#)

</insert>

<!-- 删除用户 -->

<delete id="del\_user\_by\_name" parameterClass="java.lang.String">

delete from t\_user where username=#username#

</delete>

<!-- 修改用户 -->

<update id="update\_user" parameterClass="User">

update t\_user set password=#password# where username=#username#

</update>

</sqlMap>

3.UserDao.java

**package** com.sc.VSITest.dao;

**import** java.util.Collection;

**import** com.sc.VSITest.mapping.User;

/\*\*

\* 系统DAO接口

\*

\* **@author** Administrator

\*

\*/

**public** **interface** UserDao {

**public** Collection<User> getUserList();

**public** **void** delUserByName(String username);

**public** **void** updateUser(User user);

**public** **void** addUser(User user);

**public** User findUserByName(String username);

}

4. UserDaoImpl.java

**package** com.sc.VSITest.dao.impl;

**import** java.util.Collection;

**import** org.springframework.orm.ibatis.support.SqlMapClientDaoSupport;

**import** com.sc.VSITest.dao.UserDao;

**import** com.sc.VSITest.mapping.User;

**public** **class** UserDaoImpl **extends** SqlMapClientDaoSupport **implements** UserDao {

/\*\*

\* 注意：此处要在配置文件中注入SqlMapClient

\*/

**public** Collection<User> getUserList() {

**return** **this**.getSqlMapClientTemplate().queryForList("USER\_SPACE.get\_user\_list");

}

**public** **void** addUser(User user) {

**this**.getSqlMapClientTemplate().insert("USER\_SPACE.add\_user",user);

}

**public** **void** delUserByName(String username) {

**this**.getSqlMapClientTemplate().delete("USER\_SPACE.del\_user\_by\_name", username) ;

}

**public** **void** updateUser(User user) {

**this**.getSqlMapClientTemplate().update("USER\_SPACE.update\_user", user) ;

}

**public** User findUserByName(String username) {

**return** (User)**this**.getSqlMapClientTemplate().queryForObject("USER\_SPACE.find\_user\_by\_username",username) ;

}

}

5. UserService.java

**package** com.sc.VSITest.service;

**import** java.util.Collection;

**import** com.sc.VSITest.mapping.User;

/\*\*

\* 服务层接口

\*

\* **@author** Administrator

\*

\*/

**public** **interface** UserService {

**public** Collection<User> getAllUsers();

**public** **void** delUserByName(String username);

**public** **void** updateUser(User user);

**public** **void** addUser(User user);

**public** User findUserByName(String username);

}

**6.** UserServiceImpl.java

**package** com.sc.VSITest.service.impl;

**import** java.util.Collection;

**import** com.sc.VSITest.dao.UserDao;

**import** com.sc.VSITest.mapping.User;

**import** com.sc.VSITest.service.UserService;

**public** **class** UserServiceImpl **implements** UserService {

/\*\*

\* 注入 UserDao

\*/

**private** UserDao userDao;

**public** Collection<User> getAllUsers() {

Collection<User> userResult = userDao.getUserList();

/\*\*

\* do some otherthings

\*/

**return** userResult;

}

**public** User findUserByName(String username) {

**return** userDao.findUserByName(username);

}

**public** **void** addUser(User user) {

userDao.addUser(user);

}

**public** **void** delUserByName(String username) {

userDao.delUserByName(username);

}

**public** **void** updateUser(User user) {

userDao.updateUser(user);

}

**public** UserDao getUserDao() {

**return** userDao;

}

**public** **void** setUserDao(UserDao userDao) {

**this**.userDao = userDao;

}

}

7. AddUserController.java

**package** com.sc.VSITest.Controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

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

**import** javax.servlet.http.HttpServletResponse;

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

**import** org.springframework.web.servlet.mvc.multiaction.MultiActionController;

**import** com.sc.VSITest.mapping.User;

**import** com.sc.VSITest.service.UserService;

**public** **class** AddUserController **extends** MultiActionController {

/\*\*

\* 注入UserService

\*/

**protected** UserService userService;

/\*\*

\* 注入viewPage viewPage 代表处理完后要返回的页面

\*/

**private** String page\_listUser;

**private** String page\_addUser;

**public** ModelAndView addUser(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

User user = **new** User();

user.setUsername(request.getParameter("username"));

user.setPassword(request.getParameter("password"));

userService.addUser(user);

**return** **new** ModelAndView(page\_listUser, "users", userService.getAllUsers());

}

**public** ModelAndView addUserForm(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

**return** **new** ModelAndView(page\_addUser);

}

**public** UserService getUserService() {

**return** userService;

}

**public** **void** setUserService(UserService userService) {

**this**.userService = userService;

}

**public** String getPage\_listUser() {

**return** page\_listUser;

}

**public** **void** setPage\_listUser(String page\_listUser) {

**this**.page\_listUser = page\_listUser;

}

**public** String getPage\_addUser() {

**return** page\_addUser;

}

**public** **void** setPage\_addUser(String page\_addUser) {

**this**.page\_addUser = page\_addUser;

}

}

8 DelUserController.java

**package** com.sc.VSITest.Controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

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

**import** javax.servlet.http.HttpServletResponse;

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

**import** org.springframework.web.servlet.mvc.multiaction.MultiActionController;

**import** com.sc.VSITest.service.UserService;

**public** **class** DelUserController **extends** MultiActionController {

/\*\*

\* 注入UserService

\*/

**protected** UserService userService;

/\*\*

\* 注入viewPage viewPage 代表处理完后要返回的页面

\*/

**private** String page\_listUser;

**public** ModelAndView delUser(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

// 处理中文问题

String username = **new** String(request.getParameter("username").getBytes(

"ISO-8859-1"), "gbk");

userService.delUserByName(username);

**return** **new** ModelAndView(page\_listUser, "users", userService

.getAllUsers());

}

**public** UserService getUserService() {

**return** userService;

}

**public** **void** setUserService(UserService userService) {

**this**.userService = userService;

}

**public** String getPage\_listUser() {

**return** page\_listUser;

}

**public** **void** setPage\_listUser(String page\_listUser) {

**this**.page\_listUser = page\_listUser;

}

}

9 FindUserController.java

**package** com.sc.VSITest.Controller;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.servlet.ServletException;

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

**import** javax.servlet.http.HttpServletResponse;

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

**import** org.springframework.web.servlet.mvc.multiaction.MultiActionController;

**import** com.sc.VSITest.mapping.User;

**import** com.sc.VSITest.service.UserService;

**public** **class** FindUserController **extends** MultiActionController {

/\*\*

\* 注入UserService

\*/

**protected** UserService userService;

/\*\*

\* 注入viewPage

\* viewPage 代表处理完后要返回的页面

\*/

**private** String page\_listUser ;

**public** ModelAndView findUserByName(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

//接收输入域数据

String username = request.getParameter("input") ;

User user = userService.findUserByName(username) ;

System.*out*.println("user-----"+user);

**if**(**null**==username||username.trim().length()==0){

**return** **new** ModelAndView(page\_listUser,"users",userService.getAllUsers()) ;

}

List<User> list = **null** ;

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

list = **new** ArrayList<User>() ;

list.add(user);

}

System.*out*.println("list---"+list);

**return** **new** ModelAndView(page\_listUser,"users",list);

}

**public** ModelAndView listUser(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

**return** **new** ModelAndView(page\_listUser,"users",userService.getAllUsers()) ;

}

**public** UserService getUserService() {

**return** userService;

}

**public** **void** setUserService(UserService userService) {

**this**.userService = userService;

}

**public** String getPage\_listUser() {

**return** page\_listUser;

}

**public** **void** setPage\_listUser(String page\_listUser) {

**this**.page\_listUser = page\_listUser;

}

}

10 UpdateUserController.java

**package** com.sc.VSITest.Controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

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

**import** javax.servlet.http.HttpServletResponse;

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

**import** org.springframework.web.servlet.mvc.multiaction.MultiActionController;

**import** com.sc.VSITest.mapping.User;

**import** com.sc.VSITest.service.UserService;

**public** **class** UpdateUserController **extends** MultiActionController {

/\*\*

\* 注入UserService

\*/

**protected** UserService userService;

/\*\*

\* 注入viewPage viewPage 代表处理完后要返回的页面

\*/

**private** String page\_listUser;

**private** String page\_updateUser;

**public** ModelAndView updateUser(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

String usrename = request.getParameter("username");

String password = request.getParameter("password");

User user = userService.findUserByName(usrename);

user.setPassword(password);

userService.updateUser(user);

**return** **new** ModelAndView(page\_listUser, "users", userService

.getAllUsers());

}

**public** ModelAndView updateUserForm(HttpServletRequest request,

HttpServletResponse response) **throws** ServletException, IOException {

// 处理中文问题

String username = **new** String(request.getParameter("username").getBytes(

"ISO-8859-1"), "gbk");

User user = userService.findUserByName(username);

**return** **new** ModelAndView(page\_updateUser, "user", user);

}

**public** UserService getUserService() {

**return** userService;

}

**public** **void** setUserService(UserService userService) {

**this**.userService = userService;

}

**public** String getPage\_listUser() {

**return** page\_listUser;

}

**public** **void** setPage\_listUser(String page\_listUser) {

**this**.page\_listUser = page\_listUser;

}

**public** String getPage\_updateUser() {

**return** page\_updateUser;

}

**public** **void** setPage\_updateUser(String page\_updateUser) {

**this**.page\_updateUser = page\_updateUser;

}

}

# 七.创建velocity模板

在WEB-INF下建立目录velocity交将以下文件放下其中

1.listUser.vm

<html>

<head>

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

<title>test</title>

</head>

<body>

<h1>Test Web Application System</h1>

<hr>

<h1>velocity模板本身中文</h1>

<form action="/VSITest\_v5\_1/findUser.do" method="post" name="listform">

<input type="text" name="input">&nbsp;<input type="submit" value=" find ">&nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<a href="addUserForm.do">addUser</a>

<table border="1">

<tr>

<td>username</td>

<td>password</td>

<td>update</td>

<td>delete</td>

</tr>

#if(!$users)

<tr>

<td colspan="4">no data</td>

</tr>

#else

#foreach($user in $users)

<tr>

<td>$user.username</td>

<td>$user.password</td>

<td><a href="updateUserForm.do?username=${user.username}">update</a></td>

<td><a href="delUser.do?username=${user.username}">delete</a></td>

</tr>

#end

#end

</table>

</form>

</body>

</html>

2.addUser.vm

<html>

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

<head>

<title>test</title>

</head>

<body>

<h1>add a User</h1>

<hr>

<form action="/VSITest\_v5\_1/addUser.do" method="post" name="addUserForm">

username:<input type="text" name="username"/><br><br>

password:<input type="text" name="password"/><br><br>

<input type="submit" name="submit" value="submit"/>&nbsp;&nbsp;

<input type="reset" name="reset" value="reset"/>

</form>

</body>

</html>

3.updateUser.vm

<html>

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

<head>

<title>test</title>

</head>

<body>

<h1>update User</h1>

<hr>

<form action="/VSITest\_v5\_1/updateUser.do" method="post" name="updateUserForm">

username:<input type="text" name="username" value="${user.username}" readonly="readonly" /><br><br>

new password:<input type="text" name="password"/><br><br>

<input type="submit" name="submit" value="submit"/>&nbsp;&nbsp;

<input type="reset" name="reset" value="reset"/>

</form>

</body>

</html>

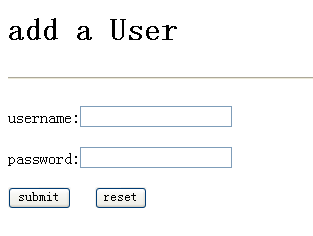
# 八 .测试

在浏览器中敲入http://localhost:8080/VSI/listUser.do

出现以下界面则配置成功



点击addUser出现界面



点击update出现界面

