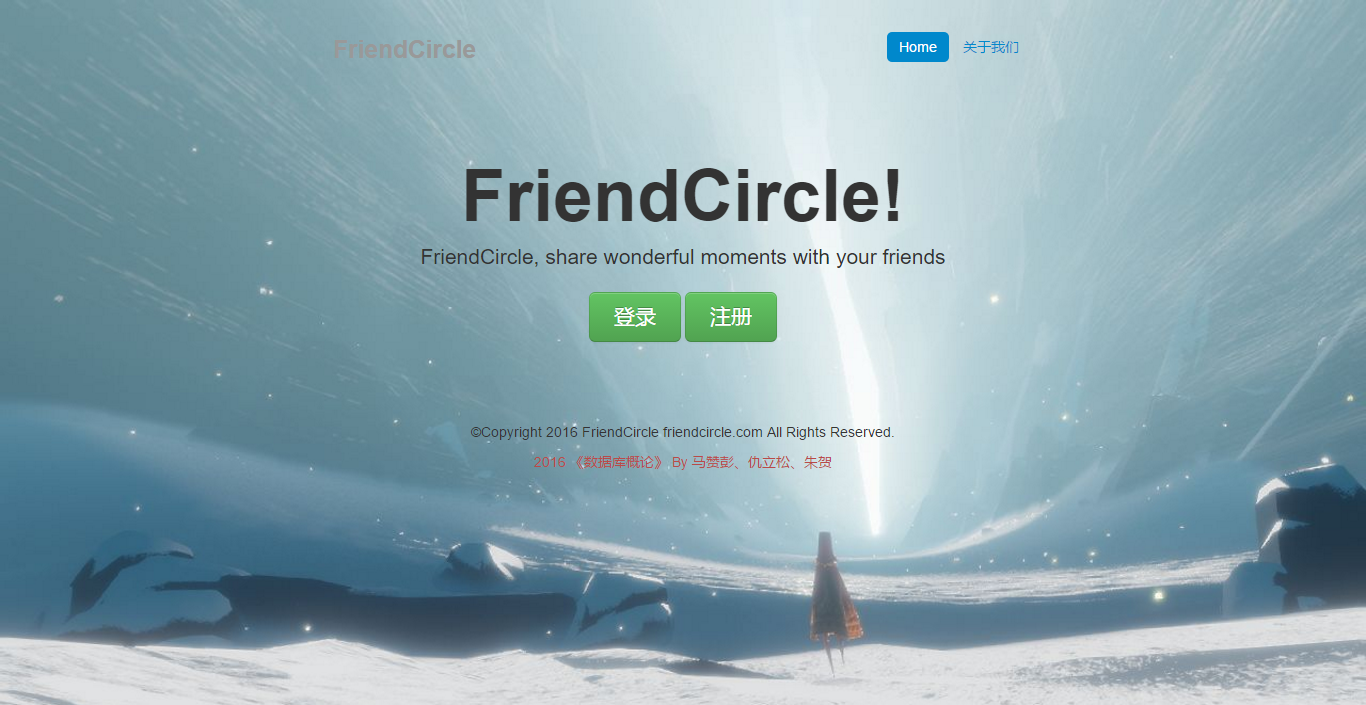
FriendCircle系统使用说明

By 仇立松，朱贺

以下对WebContent目录下各个jsp文件分别介绍：

## home.jsp：

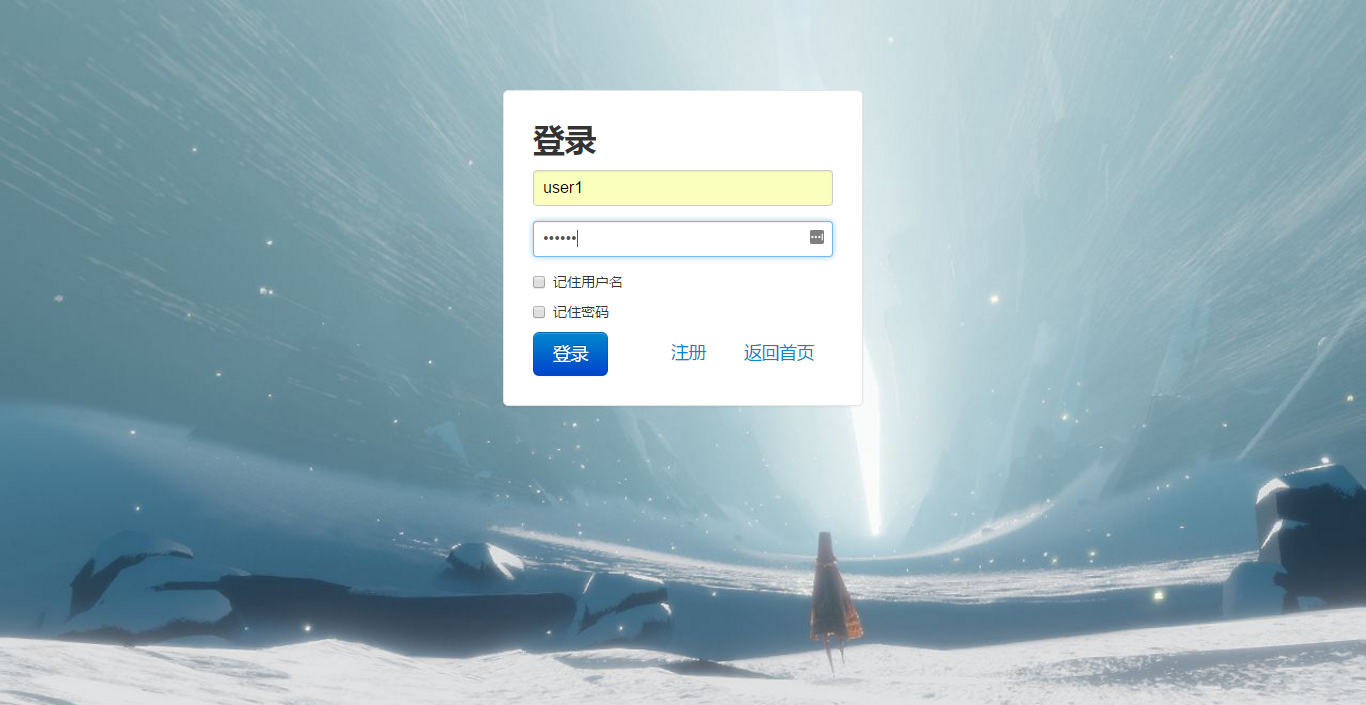
基本界面：



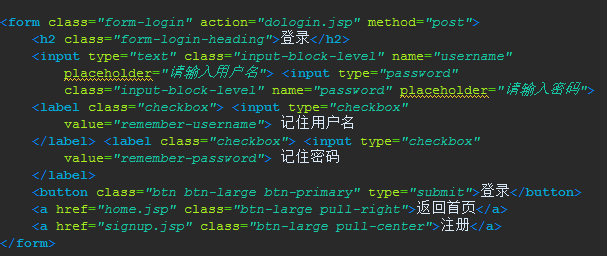
内部用了bootstrap前端框架，同时手动实现了一些元素的基本格式，<body>标签设置了页面背景图片，以及一些超链接——登录按钮和注册按钮，登录跳转到login.jsp页面，注册按钮跳转到signup.jsp页面。

## login.jsp：

基本界面：



输入用户名和密码实现登录，注册按钮跳转到signup.jsp页面，返回首页跳转到home.jsp页面。<head></head>间设置了基本格式，<body></body>中用表单(form)来实现用户名和密码的输入和处理。



客户端post之后，跳转到dologin.jsp进行处理。

dologin.jsp:

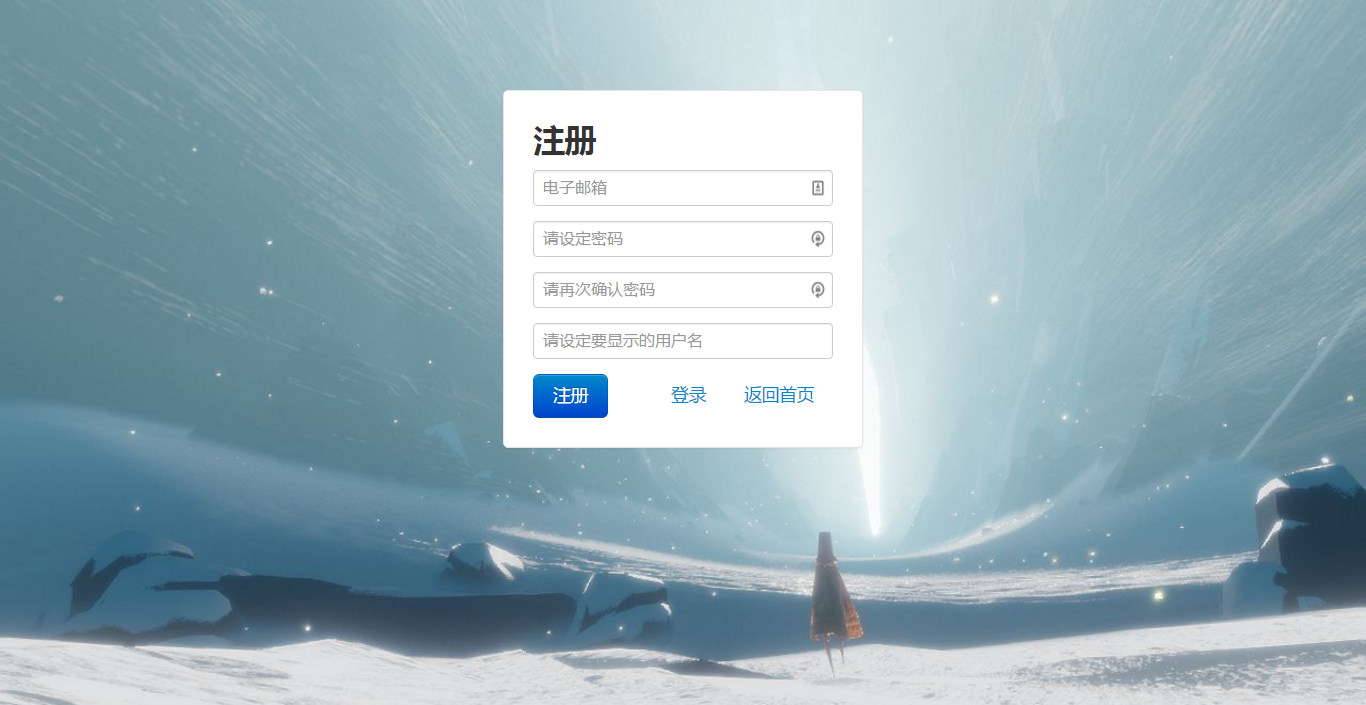
调用(import)了src包下的”friendCircle.\*”,”util.Util”。首先设置request的encoding为utf-8，以防止中文乱码。之后用request.getParameter()方法获得用户输入的username和password，并调用userAPI中的方法进行合法性判断，如果合法，则用getRequestDispatcher().forward()跳转到user-info.jsp页面，并将userName属性加入session中，以备后续网页处理使用，如果不合法，则跳转到login\_failure.jsp页面。

## login\_failure.jsp:

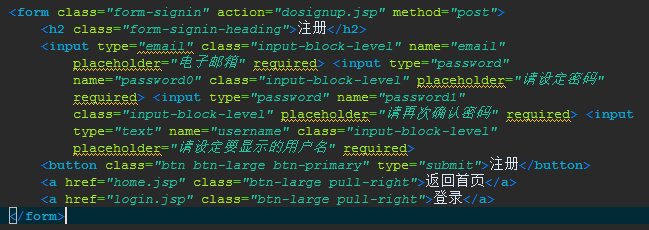
界面和功能与login基本相同，多了错误提示信息：

## signup.jsp:

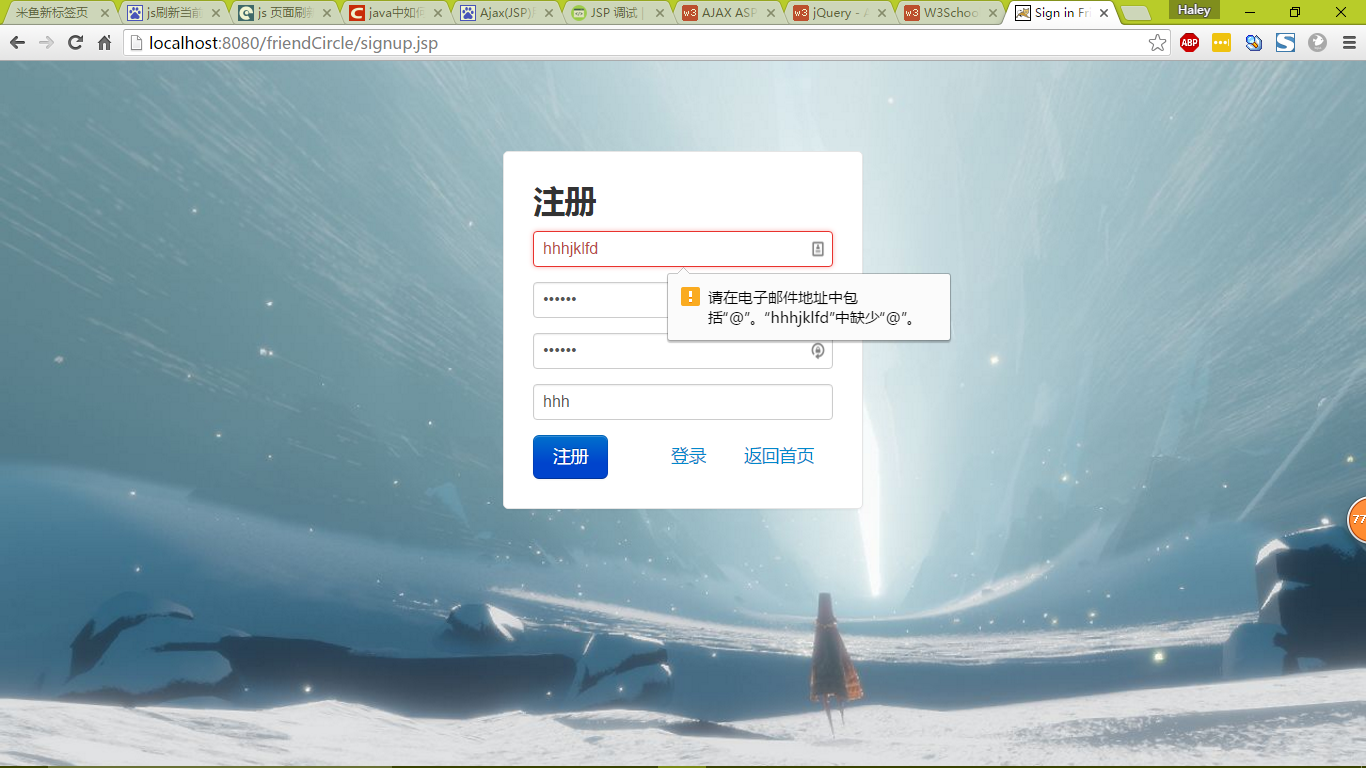
基本界面：



用于用户注册，<head></head>中定义了界面格式css，并对于电子邮箱的输入做了格式的处理，如果格式错误会提示。用<form>来post注册信息，交与dosignup.jsp进行处理。

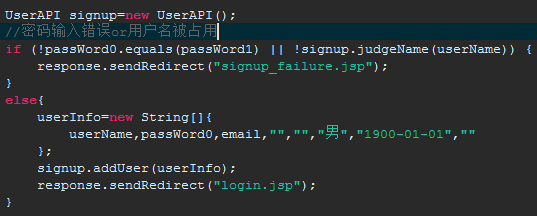


signup表单拥有输入合法性检查的功能，电子邮箱必须格式正确才能注册：



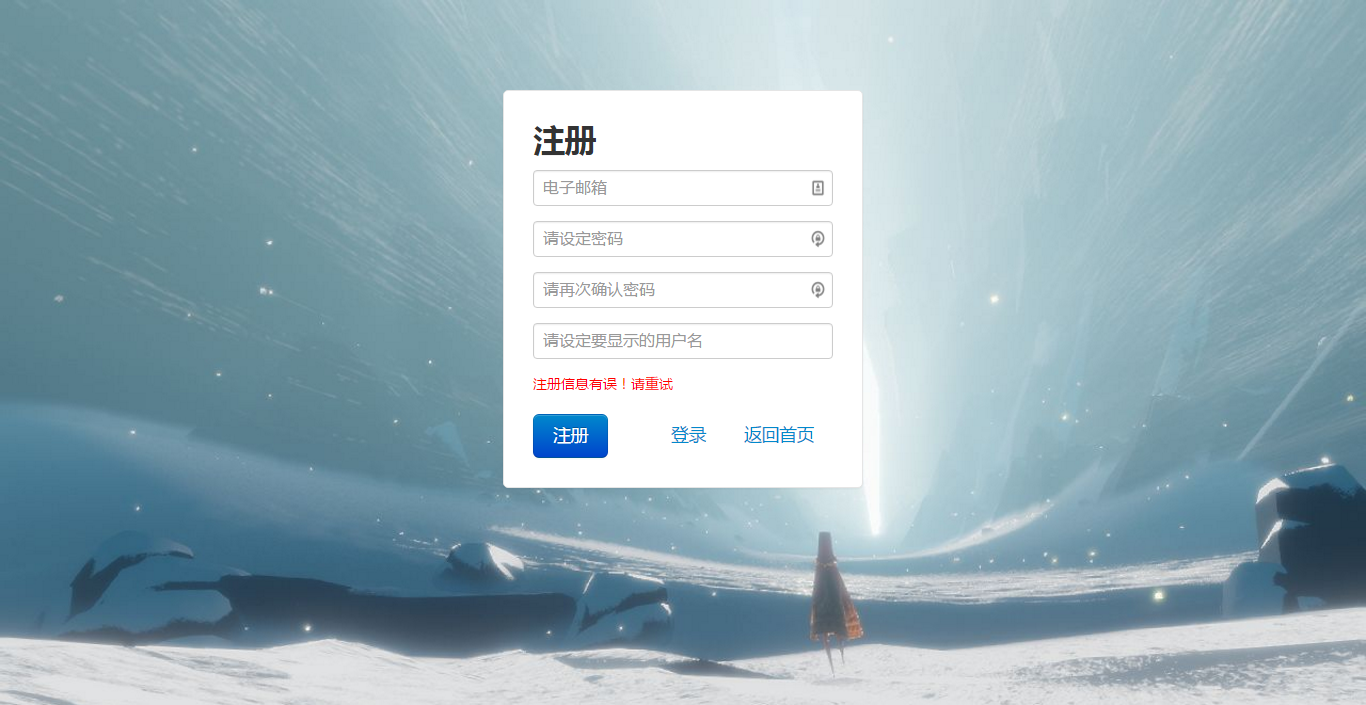
## dosignup.jsp:

用request.getParameter()获取post的信息，对于密码，使用了Util.getMD5()方法进行处理，在数据库中存储其加密后的值。之后检测输入信息的合法性，包括密码和用户名，成功则跳转到login.jsp页面，并在数据库中添加对应项(注册中未涉及的值赋予默认值)，失败则返回signup\_failure.jsp页面。



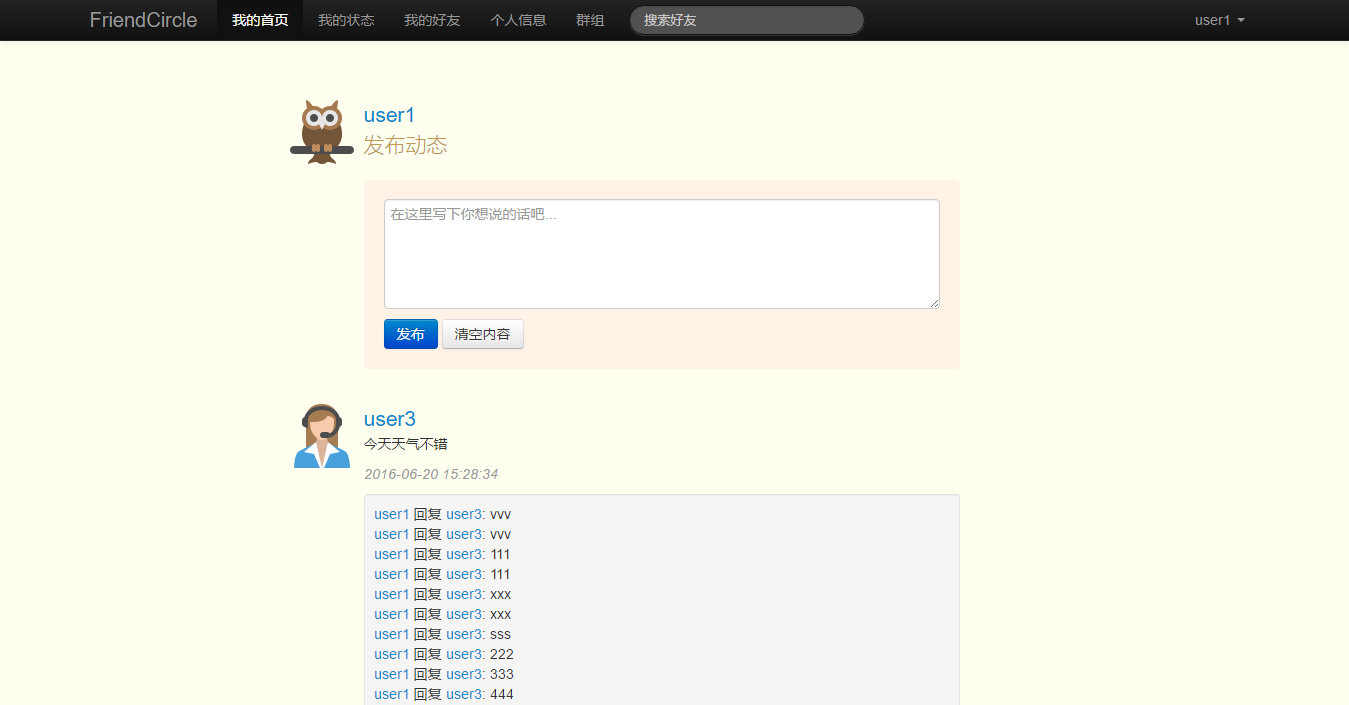
## signup\_failure.jsp

注册失败时返回的页面，基本功能与signup.jsp相同。



## user-home.jsp

1. 界面如下：



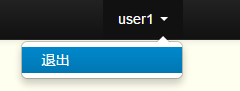
指向首页

当前用户头像

当前用户名

下拉菜单

可退出登陆



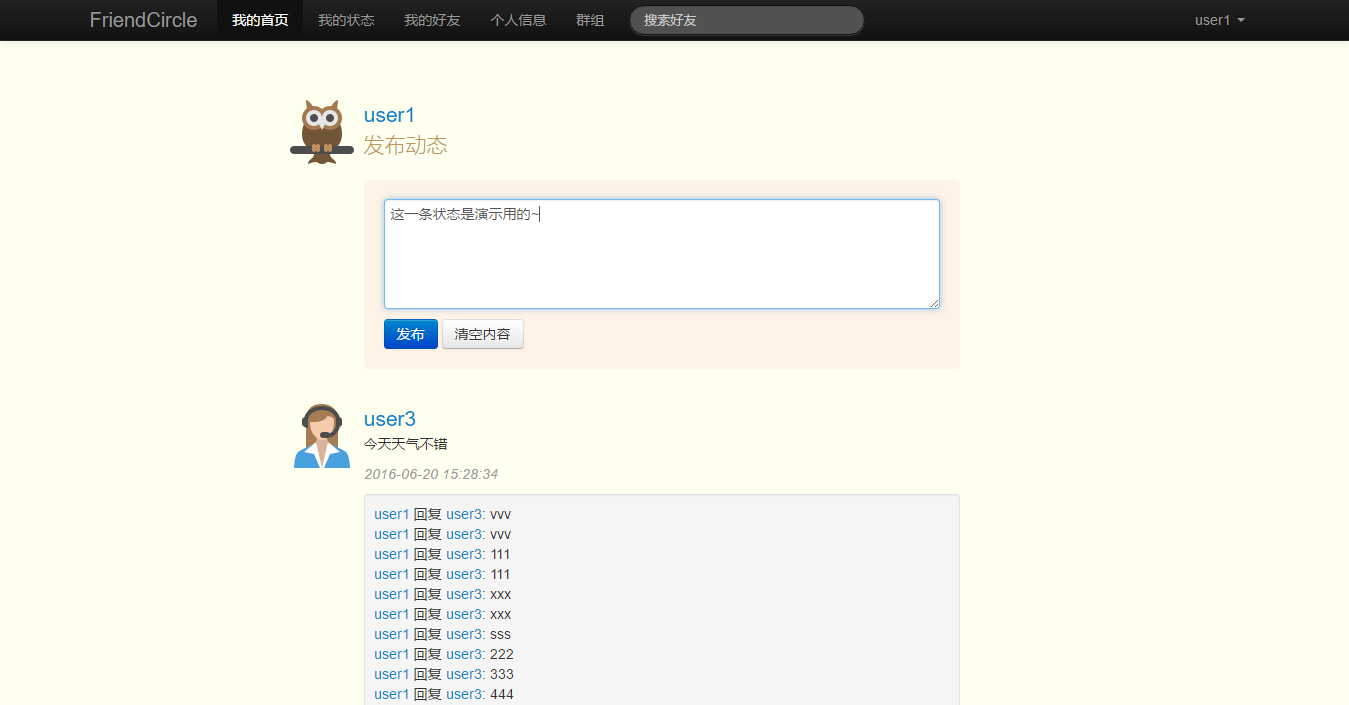
发布状态，支持一键清空编辑区内容

好友状态

好友头像

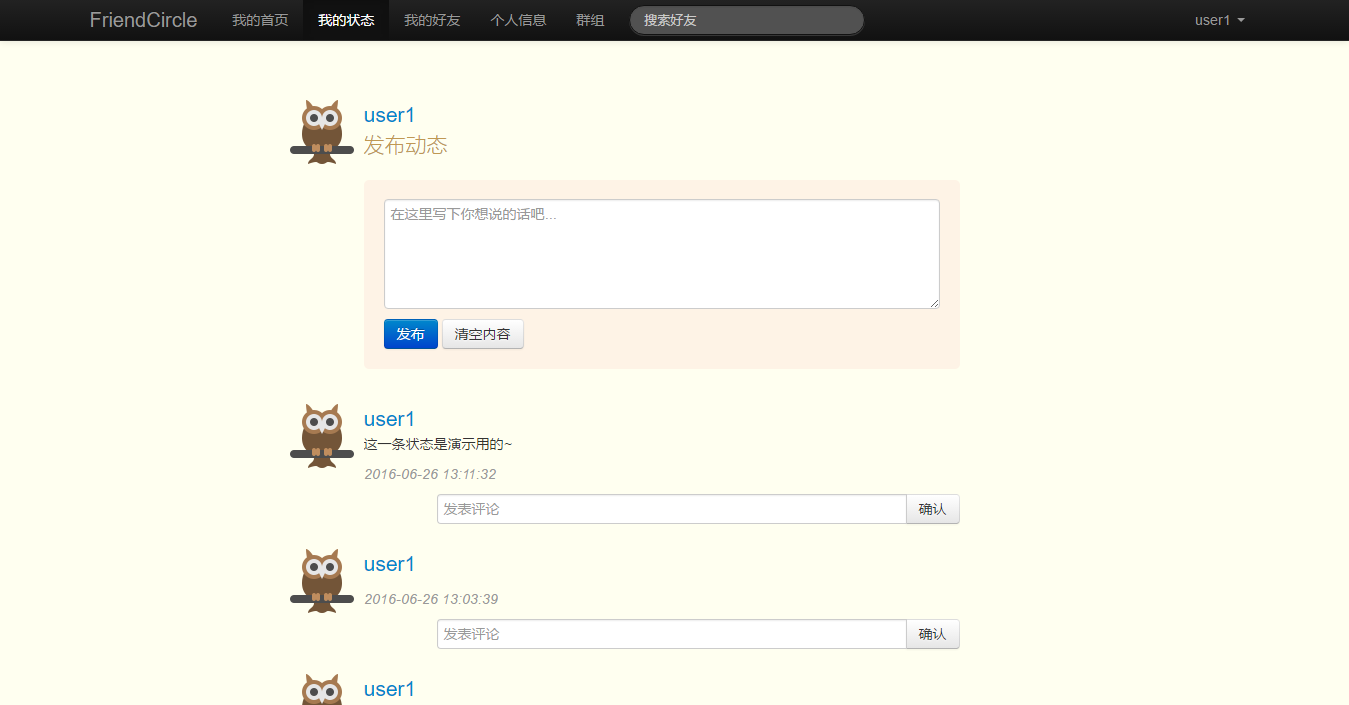
好友用户名

1. 在user-home界面发布状态将立即上传到数据库，并在自己的user-status页面看到，如下：



1.编辑消息

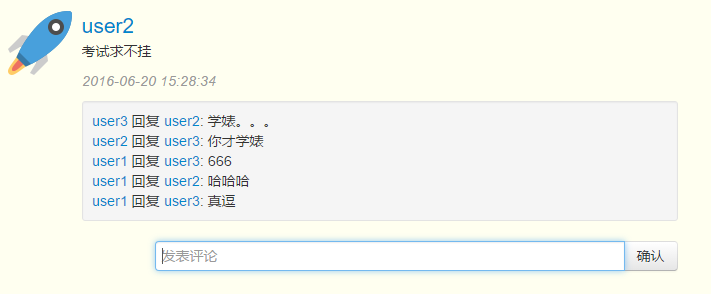
2.点击“发布”按钮



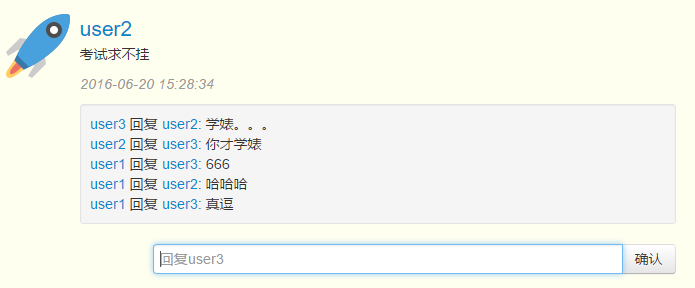
1.跳转到“我的状态”页面

2.看到刚刚发布的状态

1. user-home界面还提供了非常灵活的回复好友动态的功能，介绍如下：

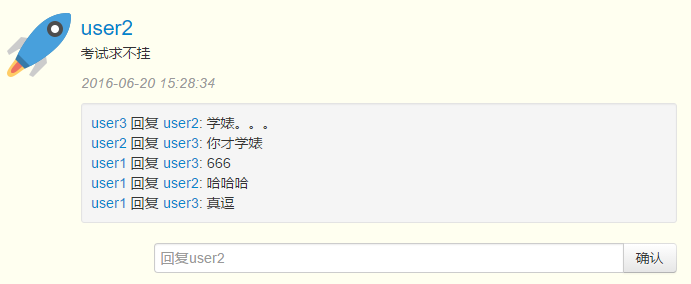


“发表评论”默认评论发状态的好友



2. 点击某个好友的名字之后，变为回复该好友

1. 点击一个好友的名字



1. 点击发状态好友的名字

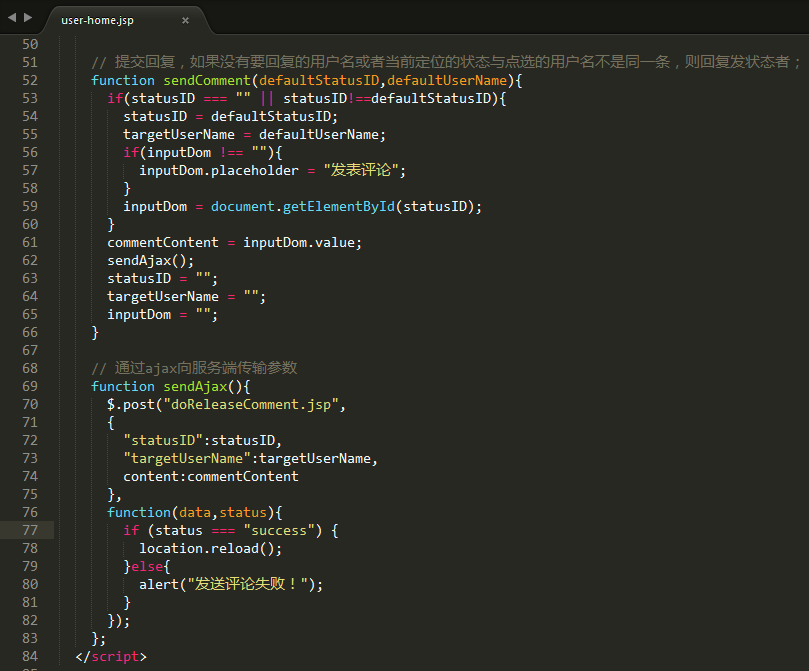
2. 重新变回评论发状态好友



1. 点击确认

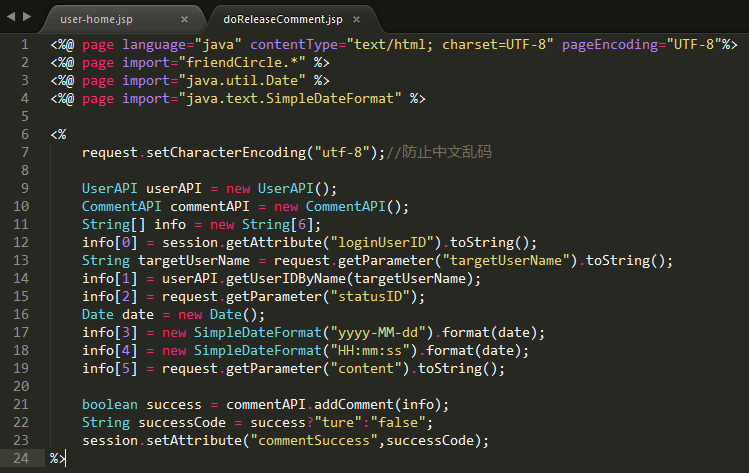
2. 页面自动刷新，评论立即显示

值得一提的是，回复功能传输的参数量比较大，所以前端使用ajax方式传输数据到服务器，服务器通过jsp处理参数：



**通过Ajax方法传输数据**

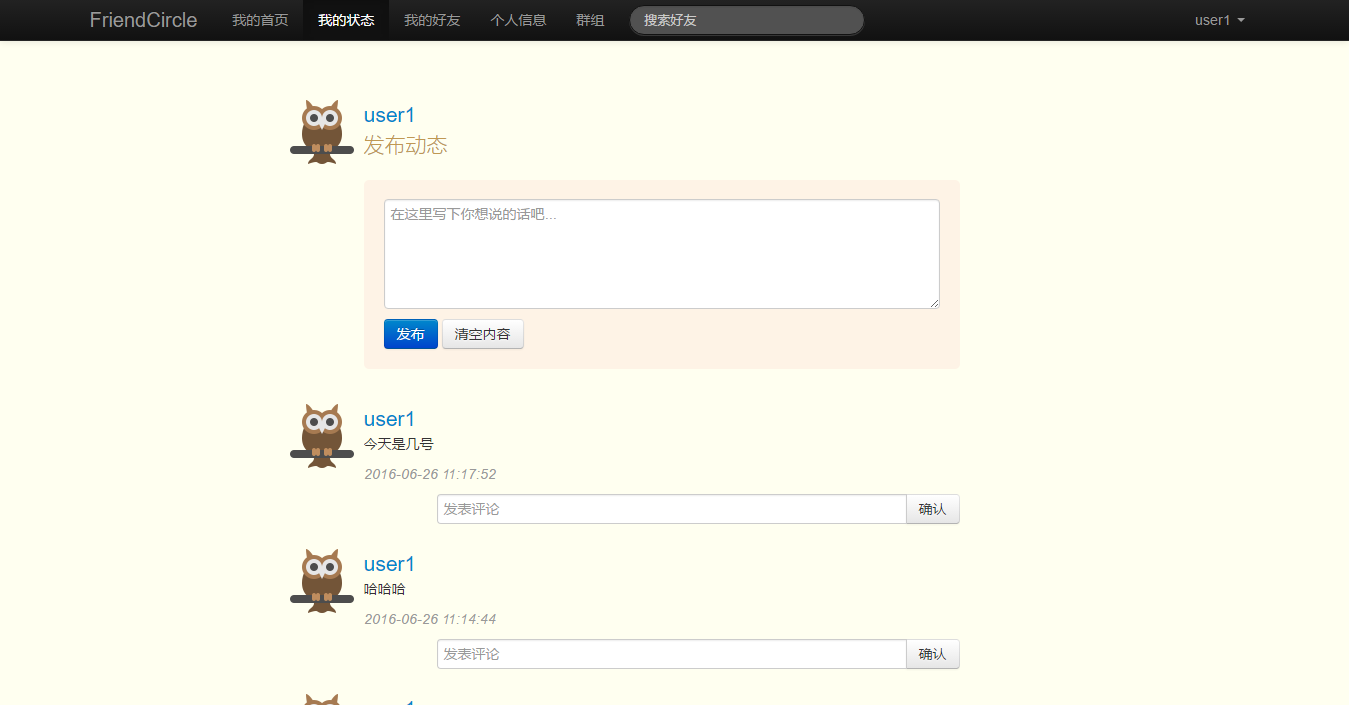
服务器端通过request.getParameter()方法接收参数：



**服务器端通过request.getParameter()方法接收参数**

## user-status.jsp

1. 界面如下：



发布状态

新的状态显示在这里

1. 列出用户最近发出的所有状态，按照时间由近到远排序，同时支持评论自己的状态，以及回复自己和好友的评论，如下图：



查看并回复好友

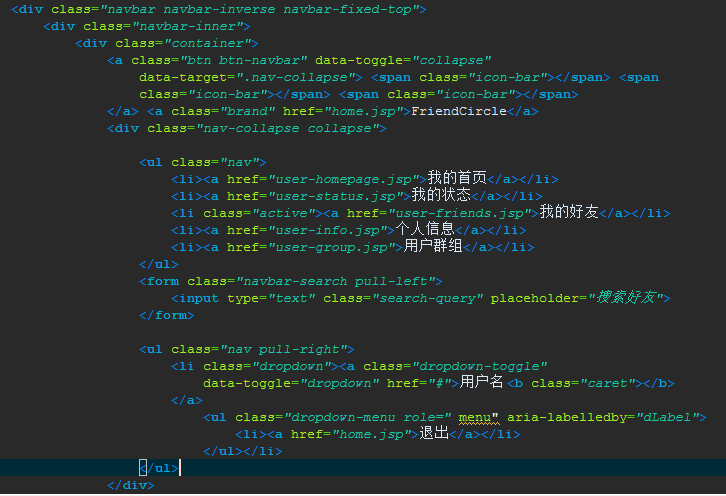
支持自我评论

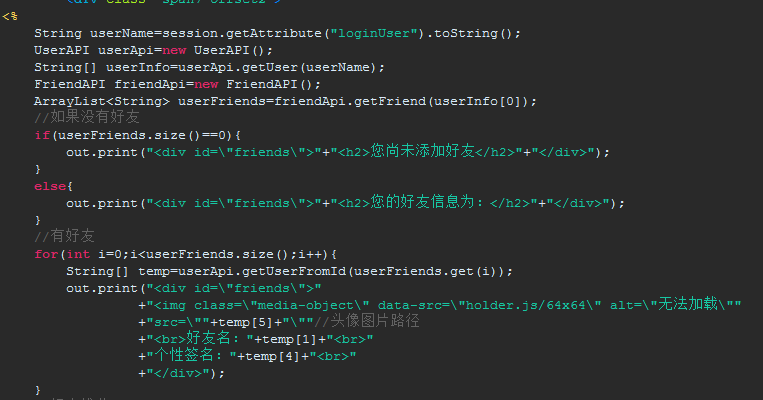
## user-friends.jsp

基本页面：

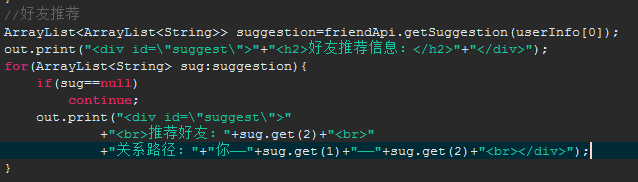


分别显示了用户目前的好友信息以及好友推荐信息（即一步好友、好友的好友）。界面顶部的分选框用bootstrap的框架实现，

。

获取好友信息import了src/friendCircle.\*中的类和getFriend方法，

好友推荐对应friendCircle.friendAPI下的getSuggestion方法(内部调用friendPath方法)，显示了推荐的好友名字和共同好友（即关系路径信息）。



## user-info.jsp

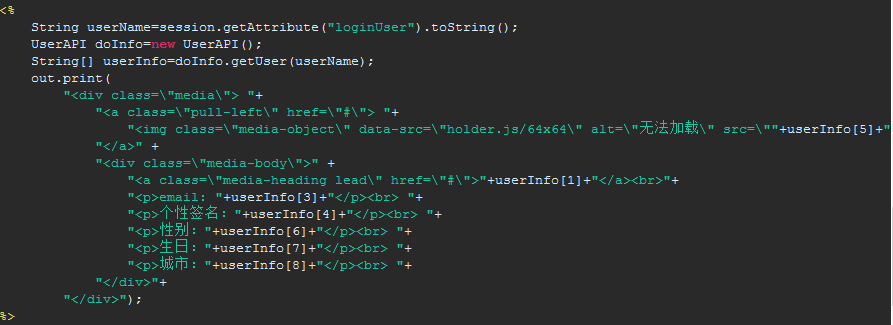
基本界面：



显示了个人的基本信息，并可点击“修改信息”按钮，在弹出框中进行个人资料修改，如图：



界面格式上顶部的分选框和user-friend.jsp中基本相同，这里不再介绍。个人信息部分调用userAPI.java并用session.getAttribute(“loginUser”)方法获取了当前登录的用户名信息（在登录时dologin.jsp中卫session添加了该信息），之后调用userAPI.getUser()方法获取了个人信息列表并输出：



按钮和弹出框的实现是设置了button的data-target属性，在弹出框中，也是用<form>来post对应的个人信息（方法同login和signup），之后再doinfo.jsp中处理提交的信息:



## doinfo.jsp

这里我一开始准备用外部类org.apache.commons.fileupload将图片上传到服务器中然后使用（我们将数据库放在了远端服务器上），但上传成功后在调用时出现了一系列问题，最终决定用本地的文件来代替，上传部分的代码作为注释保留在原文件中。之后处理修改的信息，对默认未修改的信息做了判断（源码中的a=b?c:d格式的代码），并最后修改数据库相应项。

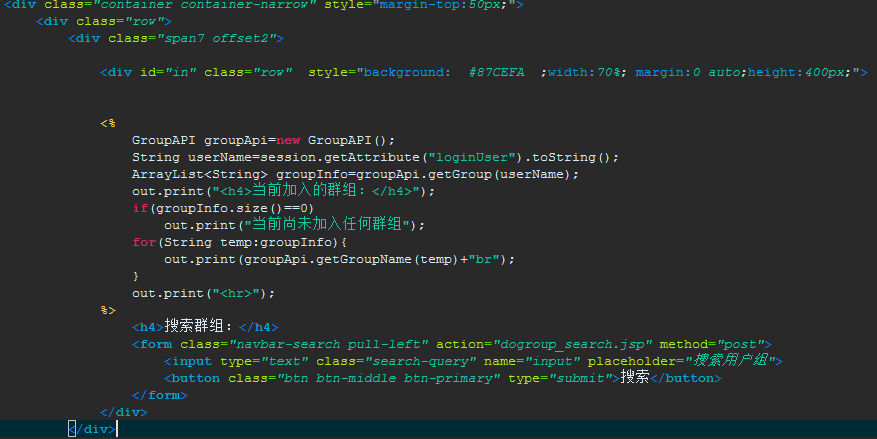


## user-group.jsp

基本界面（最后没时间美化界面了，但基本功能都实现了）：

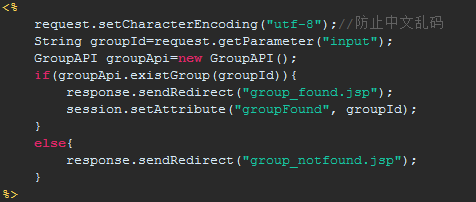


上部分显示用户当前加入的群组，后半部分处理群组的搜索。调用GroupAPI中的getGroup方法获取session中对应用户当前加入的群组并用out.print方法输出。搜索群组是用<form>来post所要搜索的群组的ID，并用dogroup\_search.jsp来返回。代码：



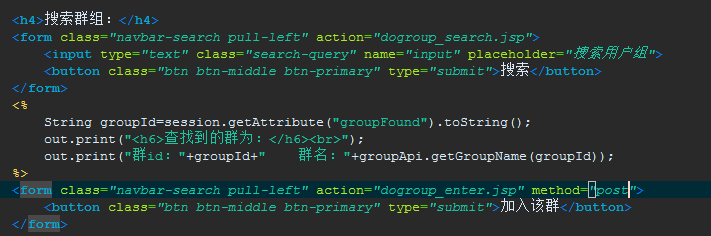
## dogroup\_search.jsp

检查groupId是否存在(对应GroupAPI.existGroup()方法)，如果存在，则跳转到group\_found.jsp页面，并在session中添加groupId属性到”groupFound”，以备后续加入群组使用，如果不存在，则跳转到group\_notfound.jsp界面。



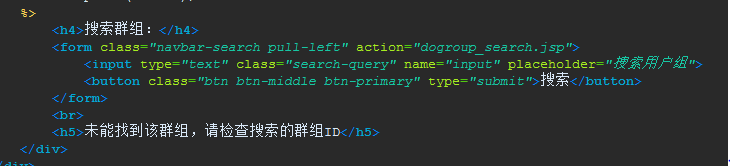
## group\_found.jsp

界面与user-group基本相同，在搜索栏底端返回搜索到的群组的结果，并添加了一个按钮，用于加入群组的处理，相应的处理在dogroup\_enter.jsp中。



## group\_notfound.jsp

界面与user-group基本相同，只是在搜索栏底端返回未搜索到群组的提示：



## dogroup\_enter.jsp

用于用户加入群组的处理，只是简单调用了gourpAPI.enterGroup()方法。

