《计算机网络》课程设计

实验报告书

班 级： 1618201

学 号： 011810402

姓 名： 张嘉慧

2020 年 12 月 10日

**目 录**

1. 平台环境
2. 架构设计
3. 实现界面或测试结果
4. 总结与展望。

网络爬虫

1. 平台环境

* python3.6
* Google chrome
* chromedriver(必须与Google chrome版本匹配)
* pycharm
* pip install selenium
* time

1. 架构设计

* 爬虫模块

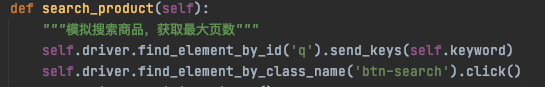
①功能：输入商品关键字和需要爬取的商品页数，可以自动爬取淘宝中该商品的名称、价格、付款人数、店铺名称、地址。

②实现原理：首先我们需要借助selenium，selenium是一个用于Web应用程序测试的[工具](https://baike.baidu.com/item/%E5%B7%A5%E5%85%B7/81891" \t "/Users/zjh20000508/Documents\\x/_blank)。selenium测试直接运行在浏览器中，就像真正的用户在操作一样。于是利用selenium我们可以自动的登录淘宝界面，但遗憾的是，由于利用selenium，会被淘宝发现，自动输入用户名和密码需要身份验证，所以登录时必须进行一步人工操作，自己扫码登录。接下来就是全自动的。

A：实现过程其实就是模拟人工操作的过程，首先我们需要实例化对象driver，它实际上就是一个模拟的谷歌浏览器，然后让它请求我们需要的地址”www.taobao.com“。



B：请求到地址之后，就需要在搜索框中输入我们需要的商品的关键字，然后按登录按钮。为了实现自动化操作，我们需要定位搜索框和登录按钮的元素，查看可以看到如下元素，于是我们可以利用driver自动找到该标签，进行对应操作。

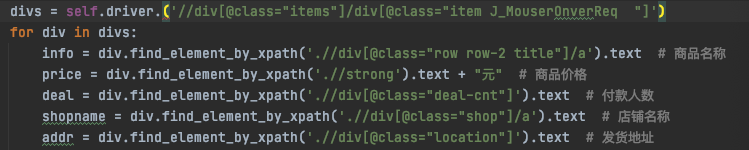
98B5E974-CA20-4A6F-A116-695AEC2C23C4B1702E1F-D9A4-4EAB-B18C-343FA0D7299F

C：点击搜索之后，由于利用了selenium，被自动检测到了，于是我们需要重新登录淘宝，且由于该网站的反爬技术十分完善，基本难以破解，所以需要我们手动扫码登录一下，注意由于扫码需要时间，我们需要调用time.sleep( )函数进行等待。

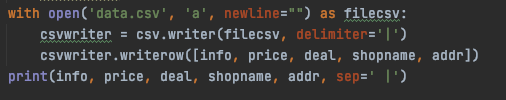
D：登录完毕后，我们需要获取数据，和第二部相同的原理，我们需要查看各个标签的关联元素。例如：商品名称、价格、付款人数等等，更加标签，我们可以利用find\_elements\_by\_xpath( )函数解析我们需要的信息.







E:解析完需要的数据之后，我们就需要把这些数据写入文件。可以写入csv文件，以便以表格的形式打开。



F：循环提取，会把第一页的数据全部提取，这时我们需要点击下一页，还需要提取总页数，于是我们可以通过总页数关联的元素，提取页数标签，然后解析。下一页相当于需要跳转到另一个地址，根据这些不同页面的地址，我们可以总结出地址的规律：



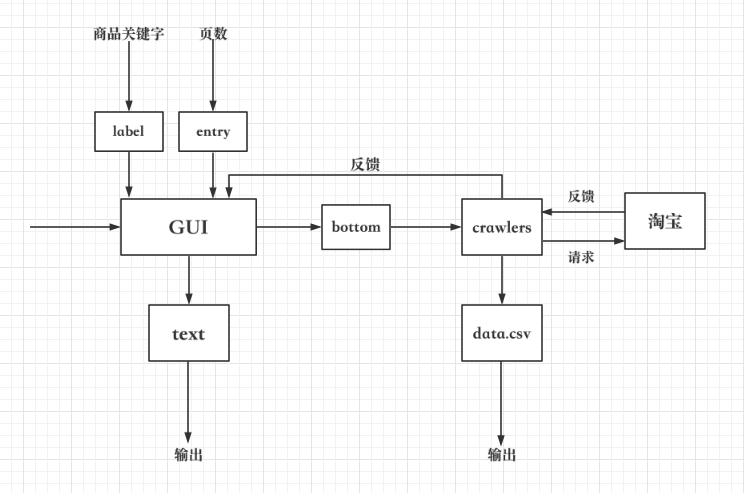
G：根据总的页面数，或者我们输入的页面数，最后可以循环提取所有页面的信息，实现功能。

* GUI模块

利用tkinter进行窗口视窗设计,主要的组件包括Label、Bottom、Text、Entry。Entry用于接收商品名称和页数，Bottom用于触发事件，当点击搜索按钮时，进入爬虫模块，爬取信息，最后反馈到Text输入框中；当点击退出按钮时，退出整个程序。



* 整体流程图



1. 实现界面或测试结果



聊天室系统

1. 平台环境

* vue
* node
* Socket
* 依赖:前端：npm install socket.io-client
* 依赖:后端：npm install socket npm install express

1. 架构设计

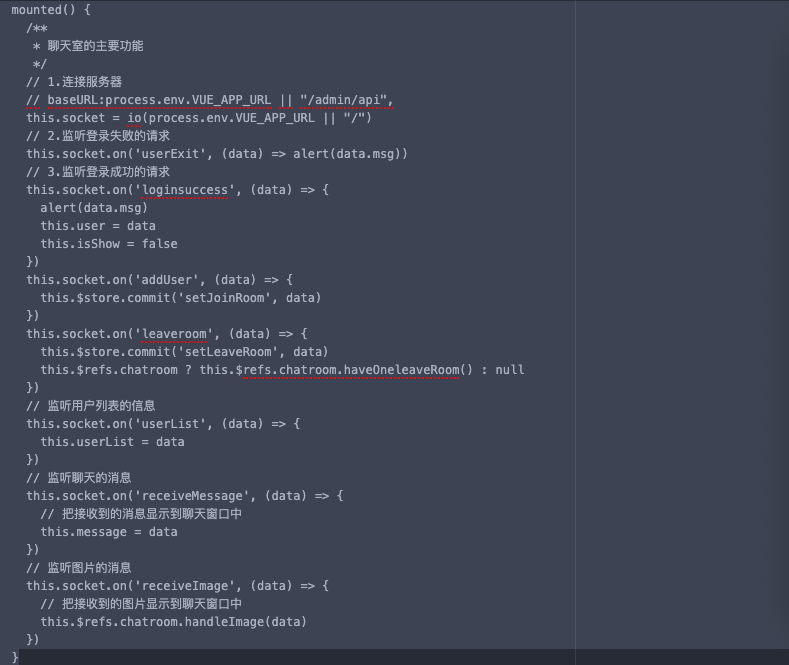
1、实现的功能：①、用户加入或退出聊天室都会有提示

②、发送消息

③、截图

④、发送表情包

⑤、图片实现大图预览



1. 基本的socket.io的demo



1. 用户加入或退出聊天室的提示



1. 事件具体实现流程（以截屏为例，其他类似）

①、监听事件(点击截图的按钮)

</template>

<span class="iconfont icon-jietu" @click="handleCanvas"></span>

</template>

②、子组件methods处理事件

handleCanvas() {

const room = this.$refs.room

html2canvas(room).then((canvas) => {

const imgUrl = canvas.toDataURL()

/ /发事件让父组件处理，imgUrl是图片的base64编码

this.$emit('handleFile', imgUrl)

})

}

③、父组件处理子组件传递的事件(methods中)

handleFile(file) {

this.socket.emit('sendImage', { ...this.user, file })

}

④、后端处理(io.on(‘connection’)中)

*// 接受图片信息*

socket.on("sendImage",data => {

*//广播给所有用户*

io.emit("receiveImage",data)

})

⑤、父组件处理后端广播的receiveImage事件

// 监听图片的消息

this.socket.on('receiveImage', (data) => {

//this.$refs.chatroom指的是子组件，调用子组件的handleImage

来处理

// 把接收到的图片显示到聊天窗口中

this.$refs.chatroom.handleImage(data)

})

⑥、子组件中的handleImage事件

handleMessageBox(newValue) {

*//判断是自己发出的信息，还是别人发出的信息*

if (newValue.username === this.user.username) {

*//是自己发的信息*

this.messageContent.push({ ...newValue, type: 1 })

}

else {

*//是别人发的信息*

this.messageContent.push({ ...newValue, type: 2 })

}

},

⑦、子组件渲染

<div class="chatcontent">

<ul class="join" ref="joinUs">

<li v-for="item in joinRoom" :key="item.username">欢迎{{item.username}}加入群聊</li>

<!-- <li v-show="leaveRoom.username">{{leaveRoom.username}}离开了群聊</li> -->

<li

v-for="(item1,index) in messageContent"

:key="index"

:class="{'my-message':item1.type===1?true:false,'other-message':ite

m1.type===2?true:false}"

>

<div v-if="item1.type === 3">

{{item1.username}}离开了群聊

</div>

<div v-if="item1.type === 1">

<img

:src="item1.file"

alt

class="file"

v-if="item1.file"

@load="loadOverImg"

preview="1"

/>

<span v-else>{{item1.msg}}</span>

<img:src="require('@/assets/avatar/'+item1.avatar)"

class="avatar" />

</div>

<div v-if="item1.type ===2">

<img:src="require('@/assets/avatar/'+item1.avatar)"alt

class="avatar" />

<p class="username">{{item1.username}}</p>

<img

:src="item1.file"

alt

class="file"

v-if="item1.file"

@load="loadOverImg"

preview="1"

/>

<p class="content" v-else>{{item1.msg}}</p>

</div>

</li>

</ul>

</div>

1. 实现界面或设计结果

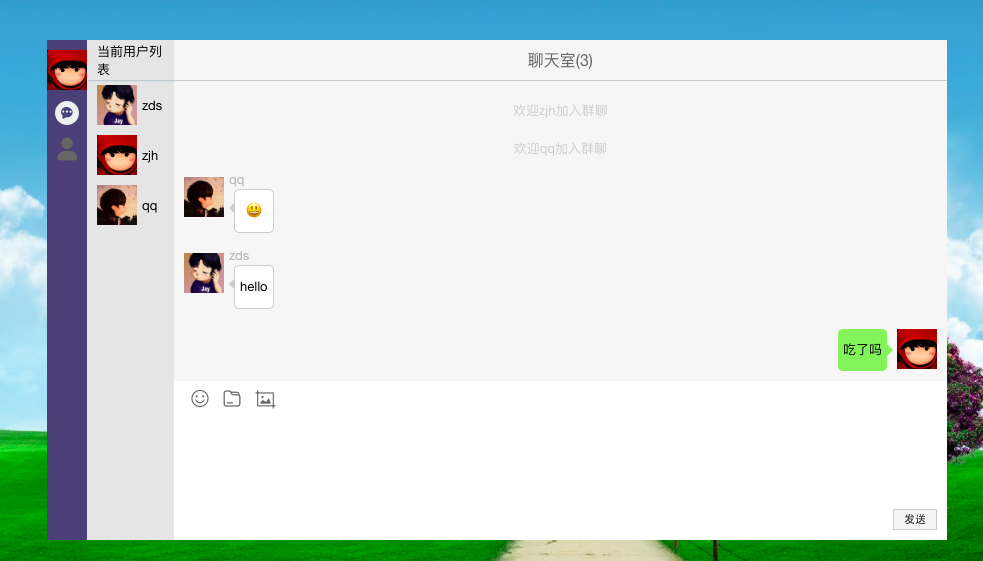
前端(chatroom文件)运行：npm install npm  run serve

后端(service文件)运行：node app.js

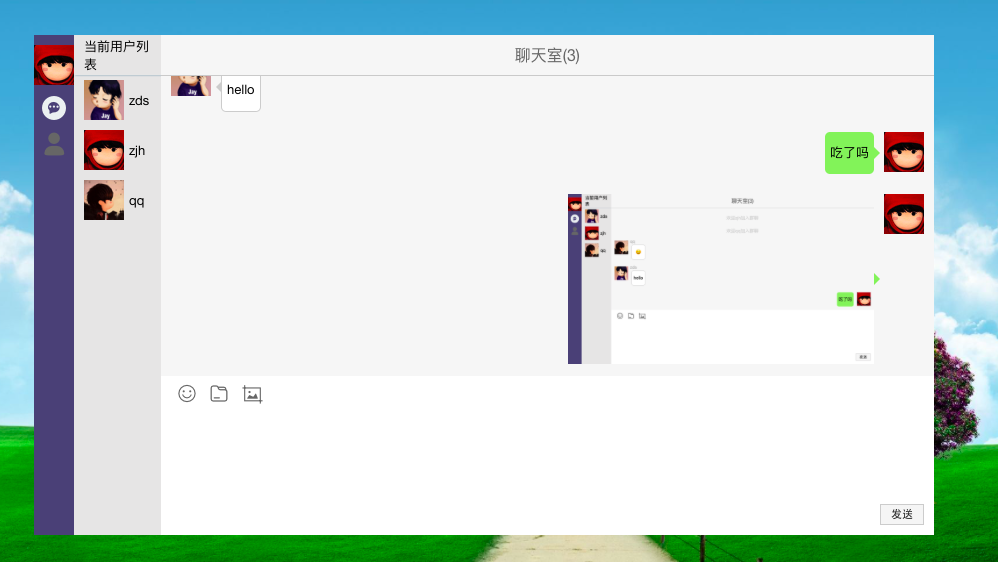
* 登录



* 发表情和消息



* 发截图



* 图片大图预览



* 退出聊天室



1. 总结与展望

这次课设无论是爬虫还是聊天室都十分有趣，不是枯燥无味的纯编程题，而是结合了很多实践技能。以前的课设基本上是给你一个完全定好的目标，自己发挥的空间很有限，但是这次计算机网络的课设却只给了一个大概的框架，有更多的自由发挥的空间，同时这也就意味着难度上去，没有了既定的目标，我们可以实现的东西太多了，可以是一个最基本的只能聊天聊天室，也可以是有很多功能的聊天室，可以是爬取一些最基本的页面信息的网络爬虫，也可以是爬取大量数据，并对这些数据进行处理的庞大的网络爬虫。虽然有难度，但是我们可以学习的途径特别多，不局限于老师上课传授的知识，GitHub、csdn等都有大量可以参考的资料，通过这次课设我认为我的技能的到了很大的提升，不局限于理论，而更到的是实践，我初步了解了HTML语言、爬虫的原理、GUI界面的设计方法、基于vue的前端开发方法以及基于Socket的编程，然而这也都是一些初步的浅陋的了解，想要更深入的学习还需要花更多的经历，目前我开发的爬虫也只是爬取淘宝页面上最简单的信息，聊天室也只是能多人简单的聊天，寒假时间，我会继续学习，去完善这两个项目，争取去开发更多的功能，进一步提升我的实践能力。