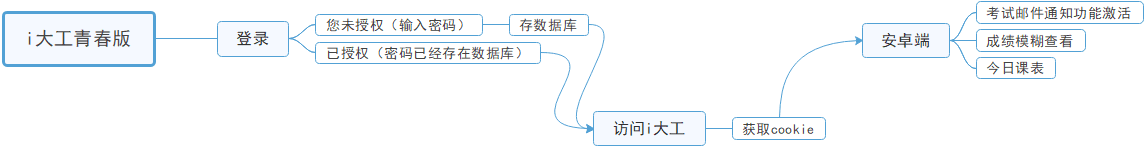
I大工Lite开源软件大作业

|  |  |
| --- | --- |
| 系别 | 软件工程 |
| 学生 | 杨岩 |
| 学号 | 201941325 |
| 班级 | 软测1901 |
| 成绩 |  |

2022 年 1 月 4 日

1. **项目背景**

**本次大作业针对于i大工app时常出现bug，程序崩溃，网络连接不上的问题，使用selenium + requests +beautifulsoup4实现同学们使用i大工所需的最基础功能：成绩查询与考试查询，帮助同学们在不适用i大工app或WebVPN的情况下实现该功能。**

**二、使用技术**

**1.** **使用MySql数据库对Python的Mysql操作进行封装，方便快捷**

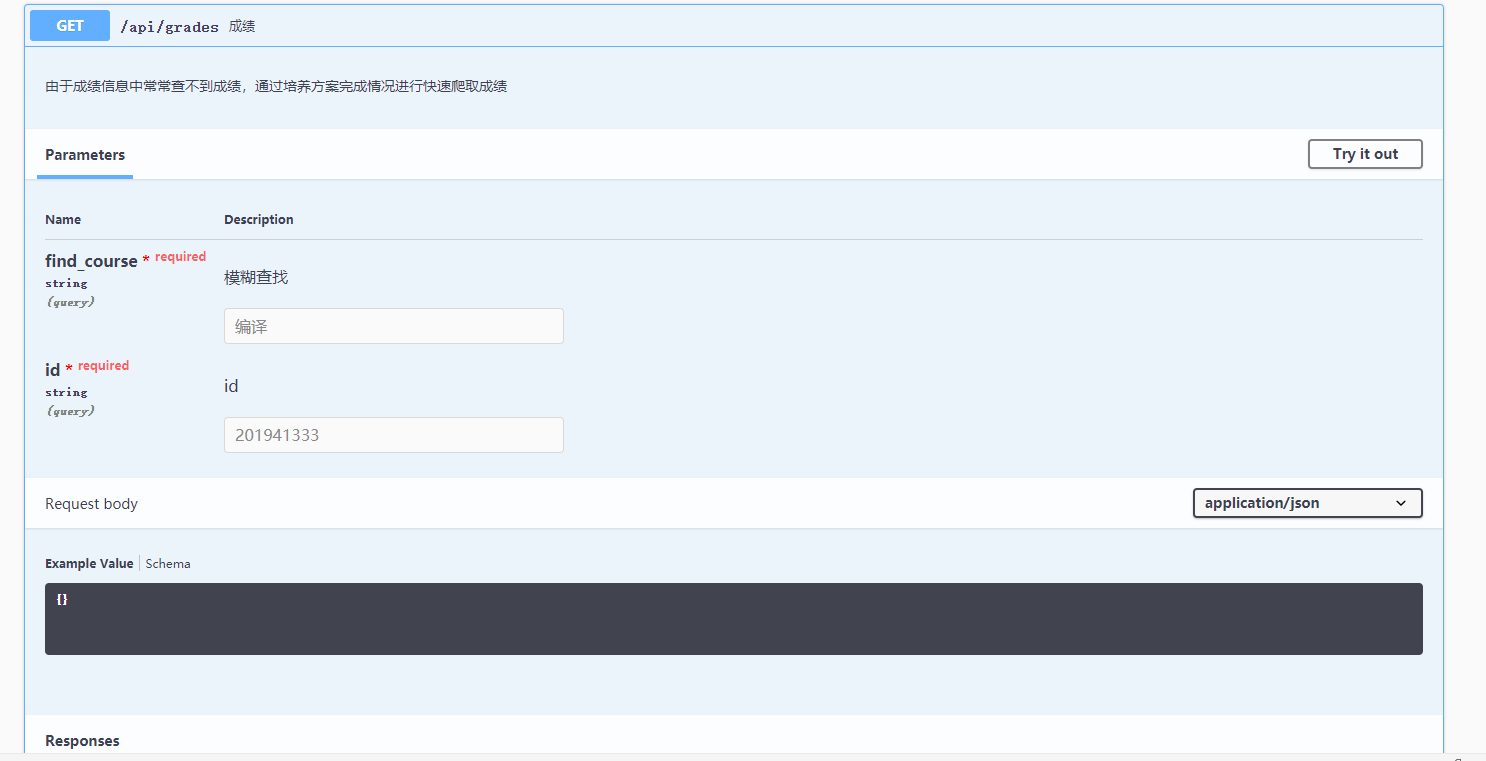
**2.使用Selenium进行电脑模拟人操作浏览器网页，可以实现自动化，测试等，通过requests +beautifulsoup4对大工综合教学管理系统信息进行爬取。**

**3.实现了3个主要的功能：**

**1.登录大工大工综合教学管理系统并获取Cookie**

**2.由于成绩信息中常常查不到成绩，通过培养方案完成情况进行快速爬取成绩**

**3.爬取考试信息，并且通过SMTP和多线程处理技术 在考试前一天定时发送邮件提醒考生。**

**4.API文档：**

**前端**

** 前端通过unity+我们编写的api实现可视化功能**

**后端**

**1. 登录大工大工综合教学管理系统并获取Cookie**

**流程：**

**登录前，我们使用Selenium，Selenium是一个Web的自动化测试工具,最初是为网站自动化测试而开发的，类型像我们玩游戏用的按键精灵,Selenium 可以直接运行在浏览器上,Selenium 可以根据我们的指令，让浏览器自动加载页面，获取需要的数据，甚至页面截屏，或者判断网站上某些动作是否发生。**

**我们在linux系统下使用 chromedriver,并且配置chrome选项(--headless)以实现在linux无图形化界面运行Chrome。**

**登录过程中，我们使用RSA非对称式加密技术，在前端使用公钥进行加密密码，在后端使用私钥进行解密获得密码，防止密码泄露**

**登录后我们获取cookie,并将(账号(primary),密码,cookie)一齐上传数据库。**

**2.由于成绩信息中常常查不到成绩，通过培养方案完成情况进行快速爬取成绩**

**通过爬虫爬取培养方案中的成绩信息（支持模糊搜索），并且修正格式后发送至前端**

**3.爬取考试信息，并且通过SMTP和多线程处理技术 在考试前一天定时发送邮件提醒考生**

**流程：**

**通过爬虫爬取考试相关信息，并且判断考试日期。**

**通过多线程技术对每一项考试进行判断，若已经过期，则线程结束；**

**若尚未过期，则待到考试前1天通过SMTP对考生邮箱发送考试提醒邮件，以免错过考试。**

**三、队员分工**

**组长：杨岩**

**后端编写，服务器环境搭建，数据库管理及Api管理**

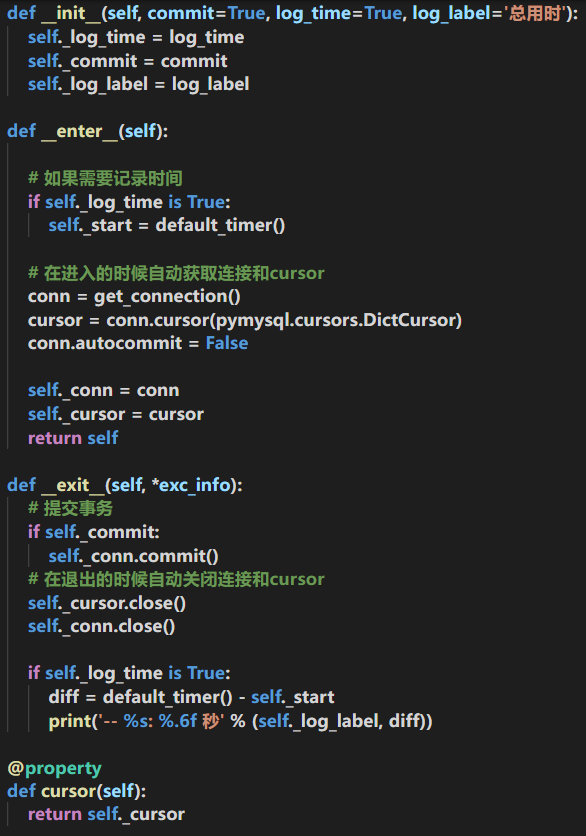
**组员：官星宇**

**使用unity编写前端，打包app**

**组员：蒲华强**

**使用unity编写前端，并完成报告的书写**

**四、代码**

**mysql.py**

**main.py**

def add\_cookies(cookiename,browser):   # broswer 加 cookies

    with open(cookiename, "r") as fp:

            cookies = json.load(fp)

            for cookie in cookies:

                cookie.pop('domain')  # 如果报domain无效的错误

                browser.add\_cookie(cookie)

@app.get('/')

def index():

    return {'message': '你已经正确创建 FastApi 服务！'}

@app.get('/query/{uid}')

def query(uid:int):

    print(uid)

    return {'success': True, 'msg': uid}

@app.get('/api/login')

def get\_Cookies(id:str,password:str):

    # 导入读取到的私钥数据, 生成 私钥对象, 返回类型为: <class 'Crypto.PublicKey.RSA.RsaKey'>

    #解密

    with open('private.pem', 'rb') as f:

        key = f.read()

        rsakey = RSA.importKey(key)  # 导入读取到的私钥

        cipher = Cipher\_pkcs1\_v1\_5.new(rsakey)  # 生成对象

        # 将密文解密成明文，返回的是bytes类型，需要自己转成str,主要是对中文的处理

        text = cipher.decrypt(base64.b64decode(password), "ERROR")

        text=text.decode(encoding='utf-8')

    print(text.replace(' ',''))

    password=text.replace(' ','')

    browser = webdriver.Chrome(chrome\_options=chrome\_opts)

    browser.get(url)

    print("输入账号: ")

    print("输入密码: ")

    try:

        browser.find\_element\_by\_xpath('/html/body/form[1]/div[2]/div/div[2]/div[2]/div[1]/input[1]').send\_keys(id)

        # 输入密码

        browser.find\_element\_by\_xpath('/html/body/form[1]/div[2]/div/div[2]/div[2]/div[1]/input[2]').send\_keys(password)

        # click登录

        browser.find\_element\_by\_xpath('/html/body/form[1]/div[2]/div/div[2]/div[2]/div[1]/span/input').click()

        cookies1=browser.get\_cookies()

        browser.find\_element\_by\_xpath('/html/body/div/div[2]/div/div/div/div[2]/div[3]/a').click()

        cookies = browser.get\_cookies()

    except:

        print('账号密码输入错误')

        return ''

    with open(id+"cookies.txt", "w") as fp:

        json.dump(cookies, fp)

    with open(id+"cookies1.txt", "w") as fp:

        json.dump(cookies1, fp)

    with UsingMysql(log\_time=True) as um:

        searchsql1 = 'select \* from user as total where id=%s'

        searchparams = [id]

        um.cursor.execute(searchsql1, searchparams)

        data = um.cursor.fetchone()

        print(data)

        if (len(data) !=0):

            sql = "update user set id=%s,password=%s,cookie1=%s,cookie2=%s where id=%s"

            params = (id, password, str(cookies), str(cookies1), id)

        else:

            sql = "insert into user(id, password,cookie1,cookie2) values(%s, %s,%s,%s)"

            params = (id, password, str(cookies), str(cookies1))

        um.cursor.execute(sql, params)

    return 'success'

@app.get('/api/setmail')

def set\_Mail(id:str,mail:str):

    with UsingMysql(log\_time=True) as um:

        sql='update user set mail=%s where id=%s'

        params=(mail,id)

        um.cursor.execute(sql, params)

@app.get('/api/grades')

def find\_Grade(find\_course:str,id:str):

    browser = webdriver.Chrome(chrome\_options=chrome\_opts)

    browser.get('http://jxgl.dlut.edu.cn/student/for-std/program-completion-preview/')

    add\_cookies(cookiename=id+'cookies.txt',browser=browser)

    browser.get('http://jxgl.dlut.edu.cn/student/for-std/program-completion-preview/')

    soup=BeautifulSoup(browser.page\_source,'html.parser',from\_encoding="gb18030")

    try:

        # find\_course=input()

        course\_list=soup.find('div',{"class":"tab-content"}).find\_all(text=re.compile(find\_course))

        if(len(course\_list))==0:

            print('没找到课程')

        grades=[]

        dict={'success':True,'grades':grades}

        grade=''

        for i in course\_list:

            course=i.find\_parent().find\_parent().get\_text("  ")

            print(course)

            grades.append(course)

            grade=grade+str(course)+'\n'

        return grade

    except:

        print('没找到课程,可能被限流了')

exams\_list=[]

@app.get('/api/exams')

def getExam(id:str):

    global exams\_list

    browser=webdriver.Chrome(chrome\_options=chrome\_opts)

    exam\_url = 'http://jxgl.dlut.edu.cn/student/for-std/exam-arrange'

    browser.get(exam\_url)

    add\_cookies(id+'cookies.txt',browser)

    browser.get(exam\_url)

    exams=browser.find\_elements\_by\_xpath('/html/body/div/div[2]/div[1]/table/tbody/tr')  # 考试列表

    time.sleep(10)

    print(exams)

    exams\_list  # 最终返回

    info=[]

    for i in range(1,len(exams)+1):

        exam\_xpath='/html/body/div/div[2]/div[1]/table/tbody/tr[' + str(i) + ']'

        e\_name=browser.find\_element\_by\_xpath(exam\_xpath+'/td[1]').text

        e\_time=browser.find\_element\_by\_xpath(exam\_xpath+'/td[2]').text

        e\_room=browser.find\_element\_by\_xpath(exam\_xpath+'/td[3]').text

        e\_location=browser.find\_element\_by\_xpath(exam\_xpath+'/td[4]').text

        e\_school=browser.find\_element\_by\_xpath(exam\_xpath+'/td[5]').text

        exam={"e\_name":e\_name,

              "e\_time":e\_time,

              "e\_room":e\_room,

              "e\_location":e\_location,

              "e\_school":e\_school

              }

        info.append(exam)

        print(exam)

        exams\_list.append(exam)

    # 获得邮件地址

    mail=''

    with UsingMysql(log\_time=True) as um:

        searchsql1 = 'select mail from user as email where id=%s'

        searchparams = [id]

        um.cursor.execute(searchsql1, searchparams)

        data = um.cursor.fetchone()

        print(data)

    mail=data['mail']

    browser.quit()

    times = []

    email=[]

    exams\_list1=''

    for i in exams\_list:

        e\_time = i['e\_time'].split(' ')[0]

        year = e\_time.split(' ')[0].split('-')[0]

        month = e\_time.split(' ')[0].split('-')[1]

        day = e\_time.split(' ')[0].split('-')[2]

        scheduled\_time = datetime.datetime(int(year), int(month), int(day) - 1, 8, 00, 00)

        times.append(scheduled\_time)

        email.append(mail)

        exams\_list1=exams\_list1+i['e\_name'] + ' ' + i['e\_time'] + ' ' + i['e\_room'] + ' ' + i['e\_location']+'\n'

    # 多线程处理邮件

    with ThreadPoolExecutor(len(email)) as pool:

        results = pool.map(mails, times,info,email)

       # send\_mail(exams\_list)

    return exams\_list1

def mails(scheduled\_time,info,mail):

    while True:

        now = datetime.datetime.now().replace(microsecond=0)

        print(now)

        if now > scheduled\_time:

            print('过期考试')

            # 由于目前没有考试，所以测试不方便，故使用过期的考试进行测试

            info = info['e\_name'] + '  ' + info['e\_time'] + '  ' + info['e\_room'] + '  ' + info['e\_location']

            print(info)

            infos = '明天你有一门考试哦\n' + str(info)

            url = 'https://api.dzzui.com/api/mail?Host=smtp.163.com&Username=a2561952525@163.com&Password=ZRCNUIVFZJIKVBLG&Port=465&SMTPSecure=ssl&addAddress=' + mail + '&title=你明天有一门考试哦&text=' + infos

            print(url)

            req = requests.get(url)

            print(req)

            break  # 不写break的话，也可以增加scheduled\_time的值。如果都不写，会遇到重复运行的问题

        #前一天发送邮件提醒考试！

        if now == scheduled\_time:  # 等于前一天

            print('发送邮件')

            info = info['e\_name'] + '  ' + info['e\_time'] + '  ' + info['e\_room'] + '  ' + info['e\_location']

            print(info)

            infos = '明天你有一门考试哦\n' + str(info)

            url = 'https://api.dzzui.com/api/mail?Host=smtp.163.com&Username=a2561952525@163.com&Password=ZRCNUIVFZJIKVBLG&Port=465&SMTPSecure=ssl&addAddress=' + mail + '&title=你明天有一门考试哦&text=' + infos

            print(url)

            req = requests.get(url)

            print(req)

            break