

# 自动填写疫情通脚本

作者: [Linkaster](#)

原文链接: <https://ld246.com/article/1612777663909>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 自动填写疫情通脚本

## 分析填报疫情通过程

首先在微信链接中选择用默认浏览器打开(这里使用的是edge浏览器),会发现页面重定向到一个登录页

尝试输入账号密码后点开F12查看请求头,发现会将表单post到"<https://xxcapp.xidian.edu.cn/uc/wap/login/check>"这个URL进行处理. 然后就会跳转到填写疫情通的页面,最后将请求头复制下来,将数据表post到对应的url即可完成

## 代码编写

整个过程分为两次Post请求,第一次请求拿到cookies,第二次请求将表单上传,完成疫情通的填写

```
class YqtUpload():
    def __init__(self):
        pass

    # 通过账号密码获取cookies
    def getCookie(self):
        pass

    # 将疫情通数据上传
    def dataUpload(self):
        pass
```

然后编写方法

```
def getCookie(self):
    url_login="https://xxcapp.xidian.edu.cn/uc/wap/login/check"
    data_login={
        # 用户信息修改
        "username": [你的学号],
        "password": [你的密码]
    }

    headers_login={
        "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/88.0.4324.96 Safari/537.36 Edg/88.0.705.50"
    }
    while True:
        num = 1
        try:
            r = requests.post(url_login,data=data_login,headers=headers_login)
            r.raise_for_status()
            break
        except:
            time.sleep(1)
            num += 1
    finally:
        if num>5:
```

```
        print("ALERT: TimeOutError...")
        break
cookies_dict = requests.utils.dict_from_cookiejar(r.cookies)
# 使用utils.dict_from_cookiejar 将cookies数据类型转化为字典
self.cookies = parse.urlencode(cookies_dict).replace("&",";")
```

## 上传数据

```
def dataUpload(self):
    url_upload = "https://xxcapp.xidian.edu.cn/ncov/wap/default/save"
    data_upload={
        "zgfxdq": "0",
        "mjry": "0",
        "csmjry": "0",
        "tw": "2",
        "sfcxtz": "0",
        "sfjcbh": "0",
        "sfcxzysx": "0 ",
        "sfyyjc": "0",
        "jcjgqr": "0",
        "remark": "",
        # 地址修改
        # 这里可以看一下请求头中的地址参数然后复制过来
        "address": "xxxx",
        "geo_api_info": 'xxxxxx',
        "area": "xxxxxx",
        "province": "xxxxx",
        "city": "xxxxx",

        "sfzx": "0",
        "sfjcw hry": "0",
        "sfjchbry": "0",
        "sfcyglq": "0",
        "ismoved": "0",
        "sftjhb": "0",
        "sftjwh": "0",
        "sfjcwry": "0",
    }
    headers_upload={
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 Safari/537.36 Edg/88.0.705.50",
        "cookie":self.cookies
    }
    try:
        r = requests.post(url_upload,headers=headers_upload,data=data_upload)
        r.raise_for_status()
    except:
        print("upload failed!")
        exit(0)
    response = json.loads(r.text)
    print(response['m']+" " + time.asctime( time.localtime(time.time()) ))
```

## 部署到服务器

使用xftp将代码文件上传到服务器,然后使用crontab命令创建自动任务.

直接在crontab命令中调用python脚本可能会出错,所以可以通过调用bash脚本,来使用bash脚本调用python脚本.

yqt.sh脚本内容如下

```
#!/bin/bash
```

```
cd /home/admin/script  
/usr/bin/python3 yqt2021.py >> /home/admin/script/yqt2021.log 2>&1
```

使用crontab创建自动任务

```
crontab -e //使用vim打开crontab配置文件  
crontab -l //查看当前用户定时任务
```

输入crontab -e后,输入

```
30 8 * * * /home/admin/script/yqt.sh >/dev/null2>&1,在每日上午8:30的时候自动执行脚本
```

以上,就完成了脚本编写到部署的全部内容

最后查看一下日志:

```
操作成功 Sun Jan 24 08:30:03 2021  
操作成功 Mon Jan 25 08:30:02 2021  
操作成功 Tue Jan 26 08:30:03 2021  
操作成功 Wed Jan 27 08:30:02 2021  
操作成功 Thu Jan 28 08:30:02 2021
```

可以正常运行