链滴

记录公司项目 --- 模拟用户登录

作者：kidcao
原文链接：https://ld246.com/article/1626073859289
来源网站：链滴

模拟登录的流程

准备工作：

1. 尽可能的找开发或者文档了解项目的登录流程。这里一步非常重要能减少很多工作。

这里有几个常见的点，在模拟的时候需要注意：

1. 请求时需要带上的是“session”还是“token”
2. 登录时，token/session返回的位置是哪里
3. 需要重定向登录时，请求需要带上哪些参数
4. 如果有多套环境时，不同环境是否有对登录作出限制，例如环境A强制校验验证码，环境B没有验
5. 确定账号在所在环境能够正常登录

2. 了解流程后，最好在浏览器先登录一遍，看看整个流程是否了解。
3. 开始写脚本模拟登录

本文章这次记录的是在无文档，通过浏览器查看登录流程，熟悉流程再编写脚本的过程。

通过浏览器查看登录流程

首先先找到登录rul，然后使用账号密码登录一次，确保网址和账号密码可用。

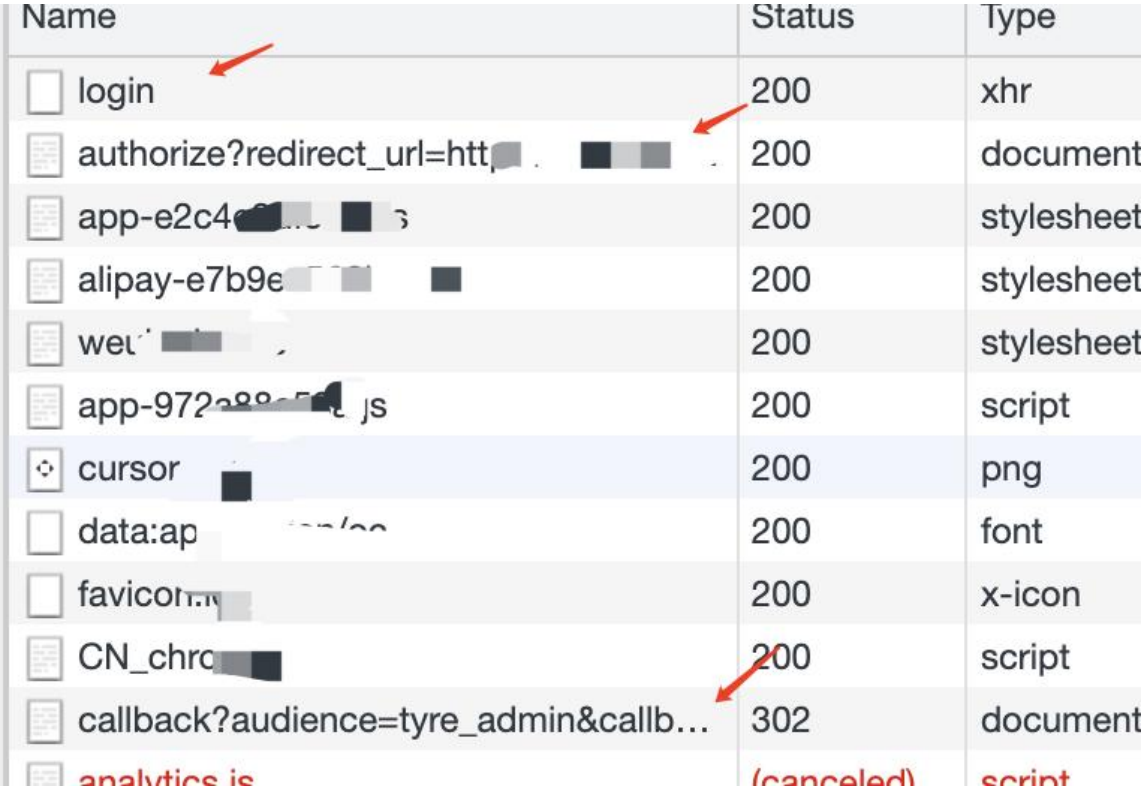
当登录成功后，可以打开浏览器的控制台查看登录过程中，到底请求了哪些url（chrome 按F12，其浏览器可以邮件鼠标点击“检查”）

这里需要注意两个点：

1. 点击 **network** tab可以用到请求记录，如果需要重定向，那么点击“all”才可以看到所有请求

2. 如果请求速度太慢，还没来得及看请求细节，可以断开网路，或者设置较慢的网路

通过查看请求记录，发现登录需要发起3次请求，第一次是登录用户中心获取用户信息，等二次是登对应项目，第三次是获得项目后台返回的token



那么我们我们拆开每一次请求来了解其中的过程。

登录用户中心

这一步是登录用户中心，获取帐号的信息。

需要在响应头获取seesion，和body中的用户ID

直接上代码

```
def login_user_center(self, username, password, captcha="xxx"):
    """
    登陆用户中心获取session和用户id
    :param username: 用户名
    :param password: 用户密码
    :param captcha: 验证码，默认"xxx"
    :return: 返回登陆成功的session和用户id
    """
    parametrize_data = {
        "username": username,
        "password": password,
        "captcha": captcha
    }

    # 这里在配置类中获取 login url
    url = bc.USERS_CENTER_LOGIN.get("login_url")
    # 这里使用requests库发起请求，由于不需要指定header信息，所以没加上
    response = requests.post(url=url, data=parametrize_data)

    # 获取相应的头
    response_headers = response.headers

    # 获取响应的body，body转换为json格式
    response_json = response.json()

    # 使用 logging 库记录响应信息
    self.logger.info(
        "login_user_center response json:{}, response headers:{}".format(response_json, response
headers))

    # 从头信息获取 session 和 id
    user_info = []
    user_info.append(response_headers.get("Set-Cookie"))
    user_info.append(response_json.get("data").get("id"))
    self.logger.info("login success, user info:{}".format(user_info))
    return user_info
```

登录项目

公司的项目是这样的架构：

- 有一个统一的用户管理后台，不同项目的用户都在这里管理
- 项目A，用户登录时需要现在用户中心登录，登录后带上seesion和id登录项目A，登录成后返回tok
n
- 项目B，用户登录时需要现在用户中心登录，登录后带上seesion和id登录项目B，登录成后返回tok

n

需要注意的是登录项目时，请求的是用户中心专用的重定向接口，接口带上需要登录的项目url，登录功后会返回获取 token url

直接上代码

```
def login_project_A(self, user_info):
        """
        登陆项目A后台获取 token
        :param user_info: 登陆用户中心返回的session和id
        :return: 返回token
        """

        # 通过配置类获取项目A某个页面的url，拼接成重定向url
        parametrize_data = {
            "redirect_url": "{}/v1/ucenter/callback?audience=tyre_admin&callback="
                            "{}}/dist/dashboard".format(self.env.get("project_A_host"), self.env.get("proje
t_A_host"))}

        # header 带上 session
        headers = {
            "Cookie": user_info[0]
        }

        # 这里通过用户中心专用的重定向接口发起请求，需要注意的是cookies重的token一定是str，
能是int
        response = requests.get(url=bc.USERS_CENTER_LOGIN.get("token_url"), params=parame
rize_data, headers=headers,
                                cookies={"token": str(user_info[1])})
        self.logger.info("response data:{}".format(response.text))

        # 响应的body内容是一个html代码，这里直接通过正则表达式匹配，也可以通过bs4的Beautifu
Soup去匹配
        # 从响应中使用正则表达式匹配token url
        token_url = re.findall(r"= '(.+)';", response.text)[0]
        self.logger.info("token_url:{}".format(token_url))

    return token_url
```

获取项目的token

上一步成功登录后，返回的html带有获取token的url，最后一步就是通过url发起请求。请求后会再重定向，重定向后的url带有token，通过正则表达式获取后就可以正常测试了。

```
# 使用token url 发起请求获取token
def get_token(self, token_url):
    token_response = requests.get(url=token_url)

    # 从响应中获得重定向的url，然后截取url中的token
    callback_token_url = token_response.url

    #匹配 token
```

```
    token = re.findall(r"toekn=(.*)", callback_token_url)

    return token
```

总结

到这里，登录就算完成了，后续每个请求都带上token就可以正常请求了。

当时我直接使用浏览器抓请求熟悉登录流程，过程真是曲折。如果可以的话，还是直接问问开发或者文档会比较省时省力！