说明文档

概述

ZJULogin 类是一个用于登录浙江大学系统的 Python 库，它包含以下功能：获取RSA公钥、加密密码、执行登录操作、以及在控制台输出日志。

使用前提

使用该库需要先安装 requests和 bs4 这两个库。

使用方式

1. 导入 ZJULogin 类：

from zju\_login import ZJULogin

2. 创建一个 ZJULogin 实例，指定登录页面的URL和HTTP头信息：（下述xx表示自己填写）

url\_login = 'xx'

headers = {'User-Agent': 'xx'}

zju\_login = ZJULogin(url\_login, headers)

3. 使用该实例进行登录，传入用户名（学号）和密码：

user\_id = 'xx'

password = 'xx'

response = zju\_login.login(user\_id, password)

4. 通过检查响应文本中是否包含“统一身份认证”检查登录是否成功，如果包含表示登录失败，否则表示登录成功

主要函数

\_\_init\_\_(self, url\_login, headers)

初始化 ZJULogin 实例

参数：

url\_login:浙江大学登录页面的URL

headers:包含在HTTP请求中的头信息

setup\_logger(self)

创建一个名为 "ZJULogin" 的日志记录器，将日志信息输出到控制台。

返回值：

logger：配置好的日志记录器

login(self, user\_id, password)

登录浙江大学系统。

参数：

user\_id:用户名（学号）

password:密码

返回值：

r3:登录尝试后的HTTP响应

get\_key(self)

获取用于密码加密的RSA公钥

返回值：

包含RSA加密密码和 'execution' 值的列表

transform\_rsa(self, m, N, e)

使用RSA加密算法加密密码

参数：

m: 待加密的密码

N : 用于RSA加密的模数

e : 用于RSA加密的指数

返回值：

pwd:RSA加密后的密码

find\_execution(self, r)

从HTML响应中找到并提取 'execution' 值

参数：

r: get登录界面的HTTP响应

返回值：

execution: execution值

post(self, u, p, execution)

执行POST请求以登录

参数：

u: 用户名（学号）

p: RSA加密后的密码

execution: execution值。

返回值：

r3: 尝试登录后的HTTP响应。

setup\_logger(self)

设置日志记录器。

创建一个名为 "ZJULogin" 的日志记录器，将日志信息输出到控制台。

示例

以下是使用 ZJULogin 类的示例代码：

from zju\_login import ZJULogin

url\_login = 'https://zjuam.zju.edu.cn/cas/login'

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0;WOW64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Safari/537.36 Core/1.94.202.400 QQBrowser/11.9.5355.400' }

user\_id = 'xx'

password = 'xx'

zju\_login = ZJULogin(url\_login, headers)

response = zju\_login.login(user\_id, password)

if '统一身份认证' not in response.text:

print('登录成功')

else:

print('登录失败')

**学习过程**

学习了用vscode而不是原生ide写代码，提高效率

复习python requests库、bs4库的基本操作，也学习了一些较高级操作

Fiddler抓包分析url、json和cookies

在维基百科上学习rsa原理，用某一次登录的数据验证代码的正误

菜鸟教程学习面向对象编程（开始只会面向过程）

看了logging库网络教程实现将日志输出到控制台

代码共四稿，变化如下

第一稿



第二稿

第三稿



第四稿



BONUS

OAuth是一种开放标准的授权框架，用于**授权第三方应用程序或服务访问用户在受保护资源上的数据，而无需将用户的凭据（例如用户名和密码）分享给第三方应用**。OAuth的主要目标是**增加互联网服务的安全性和隐私性**，同时使用户能够控制他们的数据。

OAuth的工作流程如下：

1. 客户端（我）请求授权：客户端向资源所有者请求授权访问其受保护资源。
2. 资源所有者（AppService）同意：资源所有者决定是否授予客户端访问权限。如果资源所有者同意，他们将被重定向到授权服务器。
3. 获取授权：客户端向授权服务器发送授权请求，包括其身份和所需的访问权限。
4. 授权服务器（浙大统一身份认证）验证身份：授权服务器验证客户端的身份，并验证资源所有者是否授权该客户端。
5. 授权授予：如果验证成功，授权服务器向客户端颁发访问令牌(token)。
6. 访问资源：客户端使用访问令牌来访问资源服务器上的受保护资源。

说到OAuth就不得不说SSO，即单点登录。它是一种身份验证机制，允许用户一次登录即可访问多个应用或服务，而不需要为每个应用单独登录。OAuth 2.0 的访问令牌可以用于访问多个应用或服务，但它并不直接提供单点登录功能。要实现SSO，需要使用专门设计用于此目的的协议或系统。OAuth可以与这些SSO协议结合使用，以提供综合的身份验证和授权解决方案。（我感觉浙大身份认证也用了SSO？）

**总结与反思**

1. 你觉得解决这个任务的过程有意思吗？

我觉得很有意思，自高中毕业以来我已经很久没有如此专注地去做一件事情，如此情愿地去学习相关技术。

1. 你在网上找到了哪些资料供你学习？你觉得去哪里/用什么方式搜索可以比较有效的获得自己想要的资料？

主要是各类教程。我觉得去csdn等专业性较强的网站可以比较有效地获得资料。

1. 在过程中，你遇到最大的困难是什么？你是怎么解决的？

我遇到了很多比较大的困难，例如将面向过程的代码改为面向对象，如何高效地定位“execution”值以便将其加入在POST的data中，以及如何将日志输出到控制台，我想我解决的方式就是花时间去琢磨，查阅各种资料，不断地试错，将每一个函数分离出来进行调试直到能够达到其理想功能。

4.完成任务之后，再回去阅读你写下的代码和文档，有没有看不懂的地方？如果再过一年，你觉得那时你还可以看懂你的代码吗？

没有。一年之后如果我在此期间经常使用python编程那估计看得懂，如果不是那可能就有一部分看不懂了，但我觉得我的注释还挺详细的。

1. 其他想说的想法或者建议。

建议在刚开始就说明该题需要一定基础