
DecryptLogin

发布 0.3.0

Zhenchao Jin

2022 年 08 月 29 日

1	项目声明	1
2	目前支持的平台列表	3
3	安装 DecryptLogin	5
3.1	测试环境	5
3.2	依赖包	5
3.3	PIP 安装	6
3.4	源代码安装	6
4	快速开始	7
4.1	三行代码实现模拟登录	7
4.2	实例化网站 Client 登录	7
4.3	验证码处理	8
4.4	添加代理	8
4.5	保存 cookies	8
4.6	导入 cookies	9
5	模拟登录	11
5.1	利用 Login 类实现模拟登录	11
5.2	利用 Client 类实现模拟登录	12
5.3	各平台模拟登录简介	12
5.4	一些工具函数	29
6	实战案例	31
6.1	项目声明	31
6.2	安装方式	31
6.3	项目汇总	31
6.4	使用简介	31
7	项目推荐	45
8	关于作者	47

CHAPTER 1

项目声明

本项目仅供 python 爱好者学习使用, 若作者发现该项目以任何不正当方式被使用, 将立即删除该项目。
希望大家合理利用该项目 ☺

CHAPTER 2

目前支持的平台列表

3.1 测试环境

本人测试过的使用环境如下:

- 操作系统: Win10 / Mac OS / Linux
- Python 版本: 3.6~3.8

3.2 依赖包

DecryptLogin 相关依赖包需求如下:

- rsa >= 4.0
- qrcode >= 6.1
- pillow >= 6.0.0
- requests >= 2.22.0
- pycryptodome >= 3.8.1
- requests_toolbelt >= 0.9.1
- gmssl >= 3.2.1
- PyExecJS >= 1.5.1 (测试时使用的 Node.js 版本为 v10.15.3)

3.3 PIP 安装

在终端运行如下命令即可 (请保证 python 在环境变量中):

```
pip install DecryptLogin --upgrade
```

3.4 源代码安装

1. 在线安装

运行如下命令即可在线安装:

```
pip install git+https://github.com/CharlesPikachu/DecryptLogin.git@master
```

2. 离线安装

利用如下命令下载 DecryptLogin 源代码到本地:

```
git clone https://github.com/CharlesPikachu/DecryptLogin.git
```

接着, 切到 DecryptLogin 目录下:

```
cd DecryptLogin
```

最后运行如下命令进行安装:

```
python setup.py install
```

4.1 三行代码实现模拟登录

你可以利用如下三行代码简单地实现支持列表中的任意一个网站的模拟登录操作。以模拟登录百度网盘为例:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.baidupan(username='Your Username', password='Your Password'
→')
```

其中 `infos_return` 为一个字典对象, 包含了模拟登录后网站返回的用户登录信息 (例如 `userid` 等), `session` 为网站模拟登录后的会话。运行效果类似如下:

4.2 实例化网站 Client 登录

你也可以通过实例化的网站 `Client` 进行登录, 例如:

```
from DecryptLogin import login

client = login.Client()
weibo = client.weibo(reload_history=True)
infos_return, session = weibo.login('me', 'pass', mode='scanqr')
```

设置“`reload_history=True`”可以实现历史会话的自动导入功能了, 效果类似这样 (第一次登录的时候是扫码登录, 第二次登录就显示直接导入历史会话了):

4.3 验证码处理

DecryptLogin 默认由用户手动输入验证码, 若您想加入验证码自动识别接口, 则需要传入 `crack_captcha_func` 函数。还是以百度网盘为例:

```
from PIL import Image
from DecryptLogin import login

'''定义验证码识别函数'''
def cracker(imagepath):
    # 打开验证码图片
    img = Image.open(imagepath)
    # 识别验证码图片
    result = IdentifyAPI(img)
    # 返回识别结果(百度网盘为数字验证码)
    return result

lg = login.Login()
infos_return, session = lg.baidupan(username='Your Username', password='Your Password
↪', crack_captcha_func=cracker)
```

4.4 添加代理

若您想利用代理服务器来实现网站模拟登录, 则需传入 `proxies` 参数。以模拟登录 B 站为例:

```
from DecryptLogin import login

lg = login.Login()
proxies = {'https': '127.0.0.1:1080'}
infos_return, session = lg.bilibili(username='Your Username', password='Your Password
↪', proxies=proxies)
```

`proxies` 支持的对象格式请参见: [requests 设置代理](#)

4.5 保存 cookies

对于想要保存 `session` 对象中的 `cookies` 的用户, 您可以利用如下方法进行保存:

```
from DecryptLogin.modules.utils.cookies import saveSessionCookies

session = requests.Session()
session.get(url)
saveSessionCookies(session=session, cookiespath='PATH to SAVE COOKIES (e.g., cookies.
↪pkl)')
```

4.6 导入 cookies

将之前保存的 cookies 重新导入到一个 requests.Session 对象, 您可以利用如下方法:

```
from DecryptLogin.modules.utils.cookies import loadSessionCookies

session = requests.Session()
infos_return, session = loadSessionCookies(session=session, cookiespath='COOKIES_PATH_
↳to be LOADED (e.g., cookies.pkl)')
```


5.1 利用 Login 类实现模拟登录

对于仅想实现网站模拟登录功能的用户, 推荐使用 DecryptLogin 库提供的 Login 类对象。具体而言, 代码实现如下:

```
from DecryptLogin import login

# 实例化Login类对象
lg = login.Login()
# 调用对应的接口实现模拟登录(以B站为例)
infos_return, session = lg.bilibili(username='Your Username', password='Your Password'
    ↪')
```

所有网站接口均支持以下几个参数:

- username: 登录用户名;
- password: 登录密码;
- mode: 选择使用移动端登录 (mode='mobile')/PC 端登录 (mode='pc')/扫码登录 (mode='scanqr'), 一般使用默认的接口即可;
- crack_captcha_func: 支持用户自定义一个验证码识别函数, 该函数传入验证码图片路径, 并返回识别结果;
- proxies: 模拟登录的过程中使用指定的代理服务器, 代理支持的格式同[Requests](#)。

5.2 利用 Client 类实现模拟登录

Client 类为目标网站的客户端类，集成了目标网站的一些常用加密算法，并支持历史登录状态检测的功能。换句话说，如果你用 DecryptLogin 在某个电脑上登录过，则该电脑会把你 session 保留下来，下次再在该电脑上进行登录操作时，我们会优先导入之前保留的 session，并自动检测该 session 是否已经过期，若过期，则再发起新的登录操作。具体而言，代码实现如下：

```
from DecryptLogin import login

# 实例化 Client 对象
client = login.Client()
# 实例化微博客户端
weibo = client.weibo(reload_history=True)
# 调用 login 函数进行模拟登录
infos_return, session = weibo.login('me', 'pass', 'scanqr')
```

所有网站的 Client 实例化对象均支持以下几个参数：

- reload_history: 是否进行历史登录状态检测，设置为 False 则直接发起新的模拟登录请求。

所有网站的 Login 函数均支持以下几个参数：

- username: 登录用户名；
- password: 登录密码；
- mode: 选择使用移动端登录 (mode='mobile')/PC 端登录 (mode='pc')/扫码登录 (mode='scanqr')，一般使用默认的接口即可；
- crack_captcha_func: 支持用户自定义一个验证码识别函数，该函数传入验证码图片路径，并返回识别结果；
- proxies: 模拟登录的过程中使用指定的代理服务器，代理支持的格式同Requests。

5.3 各平台模拟登录简介

5.3.1 新浪微博

1. PC 端登录

示例代码：

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.weibo(username, password, 'pc')
```

暂不支持 crack_captcha_func，因为目前微博的 PC 端登录都需要短信验证码验证，运行之后根据提示输入收到的 SMS 码即可。

2. 移动端登录

示例代码：

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.weibo(username, password, 'mobile')
```


暂不支持 crack_captcha_func，因为目前微博的移动端登录都需要短信验证码验证，运行之后根据提示输入收到的 SMS 码即可。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.weibo('', '', 'scanqr')
```

利用微博 APP 扫码登录即可。

5.3.2 豆瓣

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.douban(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.douban('', '', 'scanqr')
```

利用豆瓣 APP 扫码登录即可。

5.3.3 Github

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.github(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.4 网易云音乐

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.music163(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.music163('', '', 'scanqr')
```

利用网易云音乐的 APP 扫码登录即可。

5.3.5 中国铁路 12306

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.zt12306(username, password, 'pc')
```

暂不支持 crack_captcha_func，因为目前中国铁路 12306 的 PC 端登录都需要短信验证码验证，运行之后根据提示输入收到的 SMS 码即可。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.zt12306('', '', 'scanqr')
```

利用中国铁路 12306 的 APP 扫码登录即可。

5.3.6 QQ 空间

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.QQZone('', '', 'scanqr')
```

利用 TIM 或者 QQ 的 APP 扫码登录即可。

5.3.7 QQ 群

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.QQun('', '', 'scanqr')
```

利用 TIM 或者 QQ 的 APP 扫码登录即可。

5.3.8 我的 QQ 中心

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.QQId('', '', 'scanqr')
```

利用 TIM 或者 QQ 的 APP 扫码登录即可。

5.3.9 知乎

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.zhihu(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

注意, 如果提示”为了您的账号安全, 请使用短信验证码登录”, 可以尝试绑定邮箱后, 利用邮箱作为用户名登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.zhihu('', '', 'scanqr')
```

利用知乎的 APP 扫码登录即可。

5.3.10 B 站

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.bilibili(username, password, 'pc')
```

暂不支持 crack_captcha_func, 因为目前 B 站的 PC 端登录都需要短信验证码验证, 运行之后根据提示输入收到的 SMS 码即可。

2. 移动端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.bilibili(username, password, 'mobile')
```

暂不支持 crack_captcha_func, 因为目前 B 站的移动端登录都需要短信验证码验证, 运行之后根据提示输入收到的 SMS 码即可。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.bilibili('', '', 'scanqr')
```

利用 Bilibili 的 APP 扫码登录即可。

5.3.11 今日头条

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.toutiao('', '', 'scanqr')
```

利用今日头条的 APP 扫码登录即可。

5.3.12 淘宝

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.taobao('', '', 'scanqr')
```

利用淘宝的 APP 扫码登录即可。

5.3.13 京东

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.jingdong('', '', 'scanqr')
```

利用京东的 APP 扫码登录即可。

5.3.14 凤凰网

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.ifeng(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.15 搜狐

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.sohu(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.sohu(username, password, 'mobile')
```

暂不支持 crack_captcha_func。

3. 扫码登录

暂不支持扫码登录。

5.3.16 中关村在线

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.zgconline(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.17 拉勾网

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.lagou(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.18 推特

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.twitter(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.twitter(username, password, 'mobile')
```

暂不支持 crack_captcha_func。

3. 扫码登录

暂不支持扫码登录。

5.3.19 天翼

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.eSurfing('', '', 'scanqr')
```

利用天翼的 APP 扫码登录即可。

5.3.20 人人网

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.renren(username, password, 'pc')
```


支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):  
    return 'LOVE'
```

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.21 W3Cschoo(编程狮)

1. PC 端登录

示例代码:

```
from DecryptLogin import login  
  
lg = login.Login()  
infos_return, session = lg.w3cschool(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.22 鱼 C 论坛

1. PC 端登录

示例代码:

```
from DecryptLogin import login  
  
lg = login.Login()  
infos_return, session = lg.fishc(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.23 有道

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.youdao(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.24 百度网盘

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.baidupan(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

模拟登录百度网盘一般会触发安全验证机制, 请根据提示输入百度网盘账户绑定的手机/邮箱收到的验证码。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.25 Stackoverflow

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.stackoverflow(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.26 CodaLab

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.codalab(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.27 PyPi

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.pypi(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.28 斗鱼直播

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.douyu('', '', 'scanqr')
```

利用斗鱼的 APP 扫码登录即可, 如果你斗鱼账号是和 QQ 绑定的, 也可以用 QQ 或者 TIM 扫码登录。

5.3.29 咪咕音乐

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.migu(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.30 去哪儿旅行

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.qunar(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.31 小米商城

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.mieshop(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.32 微信公众号

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.mpweixin(username, password, 'pc')
```

暂不支持 crack_captcha_func，密码验证通过后一般需要用微信 APP 扫码进行二次验证。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.33 百度贴吧

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.baidutieba('', '', 'scanqr')
```

使用百度贴吧 APP 扫码登录即可。

5.3.34 大众点评

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.dazhongdianping('', '', 'scanqr')
```

使用大众点评 APP 扫码登录即可。

5.3.35 坚果云

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.jianguoyun(username, password, 'pc')
```

暂不支持 crack_captcha_func。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

暂不支持扫码登录。

5.3.36 天翼云盘

1. PC 端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.cloud189(username, password, 'pc')
```

支持用户自定义 crack_captcha_func 识别 PC 端登录的数字字母验证码, 例如:

```
def cracker(imagepath):
    return 'LOVE'
```

2. 移动端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.cloud189(username, password, 'mobile')
```

暂不支持 crack_captcha_func。

3. 扫码登录

暂不支持扫码登录。

5.3.37 QQ 音乐

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.qqmusic('', '', 'scanqr')
```

使用 QQ 或者 TIM 的 APP 扫码登录即可。

5.3.38 喜马拉雅

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.ximalaya('', '', 'scanqr')
```

使用喜马拉雅 APP 扫码登录即可。

5.3.39 中国大学 MOOC

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.icourse163(username, password, 'mobile')
```

暂不支持 crack_captcha_func。

3. 扫码登录

暂不支持扫码登录。

5.3.40 小米运动

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.xiaomihealth(username, password, 'mobile')
```

暂不支持 crack_captcha_func。

3. 扫码登录

暂不支持扫码登录。

5.3.41 腾讯视频

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.tencentvideo('', '', 'scanqr')
```

使用 QQ 或者 TIM 的 APP 扫码登录即可。

5.3.42 百度

1. PC 端登录

暂不支持 PC 端登录。

2. 移动端登录

暂不支持移动端登录。

3. 扫码登录

示例代码:

```
from DecryptLogin import login

lg = login.Login()
infos_return, session = lg.baidu('', '', 'scanqr')
```

使用百度 APP 扫码登录即可。

5.4 一些工具函数

5.4.1 Cookies

1. Cookies 保存

您可以利用如下方法保存登录后的 session 中的 cookies:

```
from DecryptLogin.modules.utils.cookies import saveSessionCookies

session = requests.Session()
session.get(url)
infos_return = saveSessionCookies(session=session, cookiespath='PATH to SAVE COOKIES_
↪(e.g., cookies.pkl)')
```

函数参数详解:

```
Function:
    保存requests.Session() 的cookies
Input:
    --session: 需要保存cookies的requests.Session() 对象
    --cookiespath: cookies的保存路径
    --encoding: 编码方式
Return:
    --infos_return: 是否保存成功的flag, 以及错误原因
```

2.Cookies 导入

您可以利用如下方法为 requests.Session 对象导入 cookies:

```
from DecryptLogin.modules.utils.cookies import loadSessionCookies

session = requests.Session()
infos_return, session = loadSessionCookies(session=session, cookiespath='COOKIES_PATH',
→to be LOADED')
```

函数参数详解:

```
Function:
    导入cookies到requests.Session()
Input:
    --session: 待导入cookies的requests.Session() 对象
    --cookiespath: cookies的保存路径
    --encoding: 编码方式
Return:
    --infos_return: 是否导入成功的flag, 以及错误原因
    --session: 导入cookies之后requests.Session() 对象
```

6.1 项目声明

本系列文章源代码仅供 python 学习交流，请勿用于商业等违法相关法律法规的用途，如有发现，我会直接删除这个系列的文章和代码。

6.2 安装方式

在终端运行如下命令即可 (请保证 python 在环境变量中):

```
pip install DecryptLoginExamples --upgrade
```

另外部分程序依赖于 ffmpeg 和 aria2c, 请确保这两个工具在你的系统环境变量中可以被调用:

- ffmpeg
- aria2c

6.3 项目汇总

6.4 使用简介

6.4.1 微博监控

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包:

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下:

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
    'time_interval': 查询微博动态的间隔时间,
}

crawler_executor = client.Client()
crawler_executor.executor('weibomonitor', config=config)
```

6.4.2 生成 QQ 个人专属报告

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包:

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下:

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
    'savedir': 生成的报告保存的文件夹,
}

crawler_executor = client.Client()
crawler_executor.executor('qqreports', config=config)
```

6.4.3 下载 B 站指定 UP 主的所有视频

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('bilibiliuservideos', config=config)
```

6.4.4 网易云个人歌单下载器

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('neteasesonglistdownloader', config=config)
```

6.4.5 网易云个人听歌排行榜

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('netease_listenleaderboard', config=config)
```

6.4.6 下载指定微博用户的所有微博数据

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
}
crawler_executor = client.Client()
crawler_executor.executor('userweibospider', config=config)
```

6.4.7 网易云音乐自动签到

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('neteasesignin', config=config)
```

6.4.8 微博表情包爬取

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
}
crawler_executor = client.Client()
crawler_executor.executor('weiboemoji', config=config)
```

6.4.9 大吼一声发微博

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
}
crawler_executor = client.Client()
crawler_executor.executor('weibosender', config=config)
```

6.4.10 淘宝商品数据小爬虫

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('tbgoods', config=config)
```


6.4.11 京东商品数据小爬虫

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('jdgoods', config=config)
```

6.4.12 批量删除微博

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
}
crawler_executor = client.Client()
crawler_executor.executor('delallweibos', config=config)
```

6.4.13 批量删除 QQ 空间说说

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('clearqzone', config=config)
```

6.4.14 在终端看网易云每日歌曲推荐

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('neteaseeveryday', config=config)
```

6.4.15 网易云音乐刷歌曲播放量

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
}
crawler_executor = client.Client()
crawler_executor.executor('neteasemusicplaylist', config=config)
```

6.4.16 天翼云盘自动签到 + 抽奖

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
}
crawler_executor = client.Client()
crawler_executor.executor('cloud189signin', config=config)
```

6.4.17 中国大学 MOOC 下载器

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+
- ffmpeg: 使用前请确保 ffmpeg 在环境变量中, [下载地址](#)

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'url': 课程链接, 例如: https://www.icourse163.org/course/SJTU-1003381021,
}
crawler_executor = client.Client()
crawler_executor.executor('moocdl', config=config)
```

6.4.18 修改小米运动中的步数

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
    'steps': 想要刷到的目标步数,
}
crawler_executor = client.Client()
crawler_executor.executor('modifyhealthsteps', config=config)
```

6.4.19 淘宝抢购脚本

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
    'trybuy_interval': 抢购商品时查询商品是否可以购买的时间间隔(单位秒),
    'server_key': Server酱的Key,
}
crawler_executor = client.Client()
crawler_executor.executor('taobaosnap', config=config)
```

6.4.20 京东抢购脚本

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
    'trybuy_interval': 抢购商品时查询商品是否可以购买的时间间隔(单位秒),
    'server_key': Server酱的Key,
    'paywd': 支付密码, 部分商品需要支付密码才能提交订单, ↵
    ↵输入密码不会导致你直接购买商品, 请放心使用,
}
crawler_executor = client.Client()
crawler_executor.executor('jingdongsnap', config=config)
```

6.4.21 B 站 UP 主监控

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
    'up_ids': 监控的UP主ID, 例如: ['406756145'],
    'time_interval': 查询UP主的动态的间隔时间,
    'server_key': Server 酱的Key,
}
crawler_executor = client.Client()
crawler_executor.executor('bilibiliupmonitor', config=config)
```

6.4.22 B 站监控关注的 UP 主并自动转发抽奖

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用于存储历史cookies的唯一标识ID,
    'time_interval': 查询UP主的动态的间隔时间,
}
crawler_executor = client.Client()
crawler_executor.executor('bilibililottery', config=config)
```

6.4.23 微博水军

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
    'targetid': 想要流量造假服务的明星微博ID, 例如: '1776448504',
}
crawler_executor = client.Client()
crawler_executor.executor('weibowater', config=config)
```

6.4.24 微博批量拉黑脚本

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
    'blacklist_ids': 想要批量拉黑的用户列表, 例如: ['1776448504', '1792951112',
    ↪ '2656274875'],
}
crawler_executor = client.Client()
crawler_executor.executor('weiboblacklist', config=config)
```

6.4.25 微博自动转发抽奖

1. 相关依赖

在终端运行如下命令即可安装运行该项目所需要的相关依赖包：

```
pip install DecryptLoginExamples
```

2. 环境配置

- 操作系统: Linux or macOS or Windows
- Python 版本: Python3.6+

3. 调用方式

脚本调用方式如下：

```
from DecryptLoginExamples import client

config = {
    'username': 用户名,
    'password': 密码,
    'time_interval': 查询微博动态的间隔时间,
}
crawler_executor = client.Client()
crawler_executor.executor('weibolottery', config=config)
```

项目推荐

- 制作小游戏
- 模拟登录系列
- 音乐下载器
- 视频下载器
- 实用工具
- 玩转微信
- 图像语义分割框架
- 美化图片或视频
- 图像压缩算法
- 免费代理工具
- 美丽的星空图
- 论文下载器
- 终端看中华人民共和国国务院新闻办公室
- 代码自由
- 深度学习小案例
- 数据分析相关的小项目
- 图片下载器
- 从零开始实现一个深度学习框架
- 小说下载器

关于作者

学生党, 主要研究方向是计算机视觉, 顺便对信息安全感兴趣。

我的个人微信公众号是: Charles_pikachu (欢迎搜索关注, 或者搜”Charles 的皮卡丘”)

我的 Github 账号是: <https://github.com/CharlesPikachu> (欢迎搜索关注)

我的知乎账号是: https://www.zhihu.com/people/charles_pikachu (欢迎搜索关注)

我的 B 站账号是: <https://space.bilibili.com/406756145> (欢迎搜索关注)

个人邮箱: charlesblwx@gmail.com