文档下载小帮手项目报告

作者：张建夫

目录

[一、项目概述 3](#_Toc185961416)

[二、项目整体架构 3](#_Toc185961417)

[三、项目模块详述 4](#_Toc185961418)

[1.文档下载小帮手.py 4](#_Toc185961419)

[2.yuanchuangli.py 4](#_Toc185961420)

[3.renrendoc.py 4](#_Toc185961420)

[4.goldenhoe.py 5](#_Toc185961420)

[5.wenku.py 5](#_Toc185961420)

[6.doc88.py 5](#_Toc185961420)

[7.docin.py 6](#_Toc185961420)

[8.basic\_tools.py 6](#_Toc185961420)

[9.list\_tool\_box.py 6](#_Toc185961420)

[10.selenium\_tools.py 7](#_Toc185961420)

四、项目主要界面贴图 [7](#_Toc185961421)

五、[总结 8](#_Toc185961421)

# 项目概述

本项目为弘扬互联网精神，同时解决各位网友因打开一个文档网站点击下载而被收费弹窗吓了一个措手不及的尴尬：



此处隆重推出个人耗时四周（三周写代码+一周测试和部署）写成的文档下载小帮手，一键解决文档付费下载问题，可以为相应的文档生成一个pdf文件并保存在本地。代码已在github上开源（https://github.com/saydontgo/python-spider-proj） 软件目前一共支持6个不同文档网站的文档爬取：百度文库，道客巴巴，豆丁，原创力，人人文库，金锄头文库。

# 项目整体架构

项目整体由两个部分组成，一个是主程序，主要是负责构建gui界面和与用户交互，另一个部分则是爬虫部分，负责对目标网站的文档进行爬取。

对于主程序，其模块名称为 ‘文档下载小帮手.py’

剩下的模块均为爬虫部分，其实现思路大致为：

1. 为每一个需要爬取的文档网站（例如百度文库），创建一个类，在类中的main方法实现主要的爬虫代码
2. 将爬取不同文档网站之间可以复用的代码编写进不同的模块方便复用
3. 所有文档网站的爬取都主要使用selenium库实现

# 三、项目模块详述

## 1. 文档下载小帮手.py：

该模块是本程序的主模块，其主要任务是生成与用户交互的gui界面，并在爬取的过程中适当提示爬取进度，同时在绕过验证码时弹出一个弹窗提示用户输入收到的验证码（百度文库爬取时需要）。

本模块的实现主要采用了python的tkinter库来生成相应的gui界面，在用户输入了想爬取网站的url并点击‘一键下载’后，该模块的主函数会创建一个新线程来处理用户本次的下载请求，线程函数也位于这个模块之中，该函数会自动识别用户输入url的网站，并创建对应的文档网站类来实现爬取，爬取结束后该线程自动退出。

## 2.yuanchuangli.py:

该模块定义了原创力文档的类，其中的main方法是爬取原创力文档的主方法，其他类内的方法均为main方法服务。

爬取原创力文档的思路如下：

原创力文档的呈现方式为图片，因此，只要得到文档对应的图片url，利用requests库请求得到图片url的内容，再将这些内容（内容实际上是二进制字节的形式）写入到打开的pdf文档，待所有内容写入后保存即可。

由于原创力文档的图片url加载采用的是慢加载的形式（也就是说，等你翻到一个页面时，该页面及其附近页面才会开始加载），在正式爬取原创力文档前要将所有的页面打开，这样才能获取所有图片的url，为实现这一点，我利用的是selenium库的execute\_script方法，该方法传进一个字符串，这个字符串的内容是js的语句，通过js语句实现强制翻页的功能。

## 3.renrendoc.py:

该模块定义了人人文库的类，其中的main方法是爬取人人文库的主方法，其他类内的方法均为main方法服务。

爬取人人文库的思路如下：

人人文库的呈现方式也为图片，主要爬取思路和原创力文档一样，唯一不同的地方是对于图片url的识别，人人文库的图片url相比较原创力文档更为多变，因此需要找到所有图片url的共性并利用re库进行匹配 ，这里测试了十几个人人文库的文档才没有问题。

## 4.goldenhoe.py:

该模块定义了金锄头文库的类，其中的main方法是爬取金锄头文库的主方法，其他类内的方法均为main方法服务。

爬取金锄头的思路如下：

金锄头文库的呈现方式也为图片，爬取思路和原创力文档一样。

金锄头文库是这6个文档网站里最好爬取的网站，而且图片加载的速率远快于原创力。

## 5.wenku.py:

该模块定义了一个百度文库的类，其中的main方法是爬取百度文库的主方法，其他类内的方法均为main方法服务。

爬取百度文库的思路如下：

百度文库使用的页面文档呈现方式为canvas（http5的新元素），使用画布来实现文档的展现（基本原理为页面会调用一个js函数，该js函数将文档的json数据（这个很容易获取）“画”在画布上），因此，爬取的思路有两种，一是找到这个js函数，并使用python复现该js函数的实现，二是直接得到canvas画完的结果（也就是让网页本身帮你做），这两种思路各有好处，前者能一定程度上展现原格式的文档，且无水印，而后者实现相对简单，且能保存完全（但不是原格式，而是图片的形式）。

这两个思路我都试过，最终选取的是后者。前者无法100%展现原格式，有一些数学符号（例如求和符号等）会变成乱码，且乱码率高，使文档根本没法理解，此外，前者的js复现过程比较麻烦，网页的js源码的函数参数大多为单个字母，且无注释。

上面两个思路看起来显而易见，但实际上在笔者爬取的过程中绕了多个弯子才确定下来，

你现在可以去全网查找（甚至github）爬取百度文库的代码，没有一个可以成功的（除了github上有一个用js写的插件，但是没有说他是怎么爬的），因此没有任何现有资料可以参考，我试了很多github里面项目代码的思路，全都fail，最初我找到一篇文章使用思路二爬取，原以为这样就是正确方法（我花了3天时间才看到这篇文章。。。当时没有思路太崩溃了），后面尝试后，发现也不行，最终我将思路转移到百度文库本身的呈现方式上来，简单了解了canvas在http5的一些应用，最后想出思路1（本思路全网唯一，至少网上找不到代码和我一个思路），在我将思路转移之后，实现爬取只花了一天时间。

在确定了思路后，实际爬取也成功了，但是在后续的过程中我又发现一些文档需要用户登录才能全文预览（这样才能爬得完全），我便加上了另外一个用户登录的功能，由于不同主机上有可能登录要使用验证码（我的电脑测试的时候就不用，但是图书馆内的电脑需要），用户会需要提供验证码以实现全文爬取（具体实现上会通过弹窗提醒用户是否要验证码，如果要，用户在弹窗内输入验证码即可）。

## 6.doc88.py:

该模块定义了道客巴巴的类，其中的main方法是爬取道客巴巴的主方法，其他类内的方法均为main方法服务。

爬取道客巴巴的思路如下：

道客巴巴也是用canvas呈现画面，但是其并不是以画布为载体呈现内容，仅仅提供一个背景，因此不能使用爬取百度文库的方法来爬道客巴巴，此外，道客巴巴有一个和其他文档网站不一样的特性，当用户（不论是不是机器人）直接用url进入道客巴巴的文档网页时，点击继续预览需要通过一个滑动验证码（即移动滑块来补全缺口），因此，道客巴巴也比百度文库多一个问题要处理。

爬取问题：我们使用浏览器自带的devtool实现，由于我们只需要文档的呈现内容，一个简单的思路是将页面其他所有的元素删除，只剩下文档本身的呈现内容，然后截图即可（需要注意的是，这个思路并不是爬取文档内容的通解，例如先前的几个文档网站，如果你这么做会导致文档有可能残缺不全）。

验证码问题：该处使用了opencv库识别验证码的缺口，由于缺口图片的初始位置是固定的，在确定了缺口图片的位置后就可以计算缺口最左侧点到缺口图片的距离，用这个距离来移动滑块（此处不是100%成功，有可能会差几个像素，但是道客巴巴允许无限次尝试，多试几次就能成功）

解决了上面两个问题后，道客巴巴的文档就能正常爬取了。

## 7.docin.py:

该模块定义了豆丁文库的类，其中的main方法是爬取豆丁文库的主方法，其他类内的方法均为main方法服务。

爬取豆丁的思路如下：

豆丁需要用户登录才能预览全文（对用户很不友好，但也防不了机器人，不知道豆丁网怎么想的），因此必须先实现登录功能才能够爬取（豆丁网的注册机制相当宽松，不需要邮箱验证即可注册成功）。豆丁的模拟登录实现较为简单，很大部分情况下不需要验证码，向账号密码栏传入用户提供的账号密码并点击登录即可成功。

豆丁网的文档呈现方式有两种，一种是和百度文库的呈现方式一样，另一种和原创力文档的呈现方式一样，要分别处理。此外，豆丁网和其他文档网站的不同点在于，豆丁网的文档呈现形式是会变的（不知道为什么要这样做，很奇怪？），你需要先翻到第二页，再开始爬第一页才能成功，原因是在于你翻到第二面之前，第一面可能是以图片的二进制形式存在（就是一串16进制串，需要解码才能形成图片），等你翻到第二面，第一面才会变成图片的url或者canvas。

## 8.basic\_tools.py:

该模块定义了一些各个爬取文档主模块（上面七个）复用的函数，主要是为了复用。

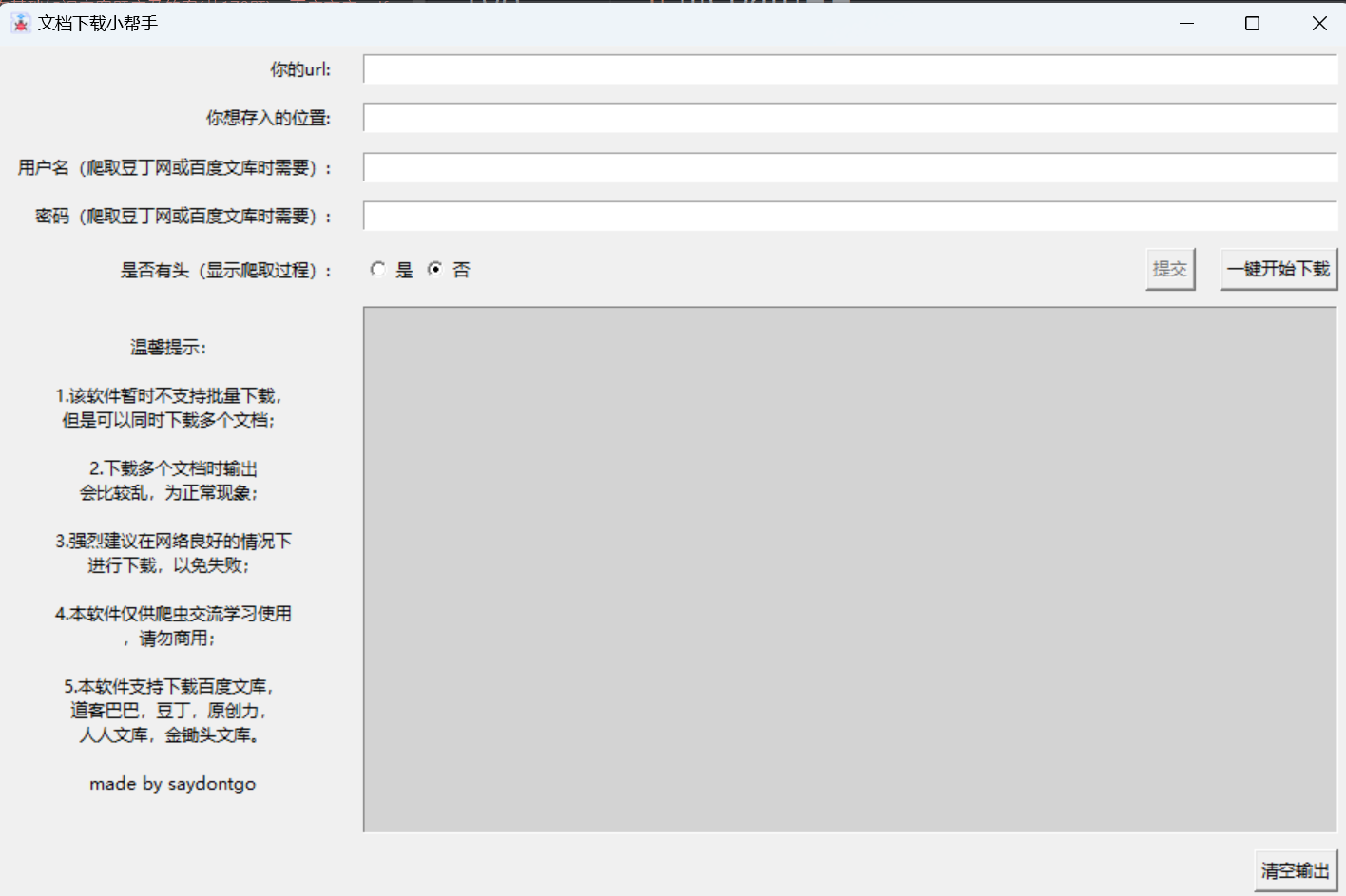
## 9.list\_tool\_box.py:

该模块定义了两个对python数据容器list的操作。

## 10.selenium\_tools.py:

该模块定义了get\_driver函数和close\_browser函数，前者主要用于得到selenium的浏览器头进行爬取，而后者用于软件非正常关闭时将打开的浏览器一起关闭，防止内存泄漏。

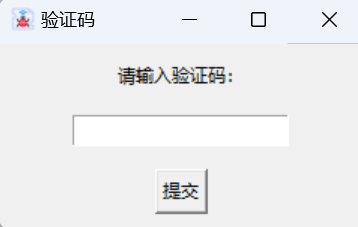
# 四、项目主要界面贴图



在爬取豆丁网和百度文库时用于用户确认其输入的用户名和密码

软件的主界面，各个功能已标识好，灰色框部分为输出部分。

除此之外，在登录百度文库时还会有一个弹窗提示用户输入收到的验证码的消息，弹窗如下：



（验证码输错了也能爬，只是可能爬不完整）

# 总结

本次项目很好地锻炼了我对python语言的使用，以及对爬虫的理解（实际上大部分是对前端的理解，爬虫的成功很大程度上依赖于对前端代码的理解和应用，找到对应元素后写代码其实很快），总而言之，作为无需担心绩点的一门课（我实际上没报课，但是和老师申请也做了项目），很多东西不用有过多的担心，这个学习爬虫和使用爬虫的过程相当充实无负担，算是最愉快的大学生活经历之一？