# 《腾讯文档模板数据采集》结题报告

## 执行目标

* + 1. 腾讯文档模板数据采集，采集腾讯文档首页上所有模板；
    2. 模板文件下载下来是word格式，统一存储在同一个文件夹中；

③ 模板文件的名称需要与模板标题一致。

## 项目的主要内容及步骤

### 设计思路

**（1）项目分析**

该项目的目的是采集腾讯文档主页所有的模板，最初的思路是使用requests库爬取网页，使用BeautifulSoup4解析网页获取下载链接后执行下载操作。后分析腾讯文档模板页的网页代码，得到以下信息：

* + 1. 该网页为动态网页，内容随用户的操作而显现的；
    2. 该网页所有的文档均没有直接的下载链接，都需要进行一系列手动操作，才能实现文档的导出和下载；
    3. 文档的下载需要进行登录，未登陆的用户无法进行文档的下载；
    4. 文档分为普通文档和VIP文档，非VIP用户不能够下载VIP文档；
    5. 页面的文档分为多个种类。

**（2）解决方案**

1. 使用selenium控制浏览器模拟用户操作并执行文档的下载。
2. 设置判定条件，当用户为VIP用户时才进行VIP文档的下载。
3. 使用cookie进行用户登录，简化了操作步骤，节省了时间。
4. 对文件进行分类下载。以文件所属类名建立文件夹，将下载的文件分别存储在相应的文件夹中，所有以类名命名的文件夹存储在同一文件夹中。

**（3）具体实现步骤**

1. 编写获取用户登录cookie的函数，功能是获取cookie，并将获取到的cookie存储到相应的json文件中。
2. 编写获取腾讯文档首页所有模板文档访问链接的函数，功能是获取文档访问链接，并将链接分类存储在相应的json文件中。
3. 分类读取保存文档下载链接的json文件，根据链接访问文档，读取保存用户cookie的文件并进行登录，并模拟用户来执行文档的下载，并将下载后的文档进行分类存储。

**(4) 优化**

1. 设置项目在后台运行，既不影响用户正常的操作，又降低了打开浏览器对系统内存的占用。
2. 采用模块化设计的思想，将相关功能代码封装成函数，通过调用函数实现所有的功能，便于后期代码的维护和修改。
3. 设计了将所有文档统一存储在同一个文件夹内的功能，方便不同用户实现不同的需求。

### 开发工具及其依赖

* + 1. 开发语言:python3
    2. 开发工具:Pycharm、Chrome浏览器
    3. 使用到的python内置库和第三方库: selenium ,os,json,time等

### 运行结果

该项目是通过控制浏览器来模拟人的操作，从而实现腾讯文档模板的采集，因此速度上远比不上爬虫直接爬取。因暂无VIP账号，故只采集了普通文档。采集的部分文档如下图所示，采集到的文档以模板标题命名，有分类存储和统一存储两种方式供用户选择:

|  |  |
| --- | --- |
| 图形用户界面  中度可信度描述已自动生成 | 图形用户界面  描述已自动生成 |
| 表格  描述已自动生成 | 表格  描述已自动生成 |

## 项目的结论

该项目以采集腾讯文档模板为目的，总体使用python语言进行开发，使用selenium库模拟用户操作，从而实现模板的采集，整体思路较为简单，涉及了前端、Python和web自动化的一些知识，实现起来也先相对容易。

该项目实现了基本的腾讯文档模板采集，满足了项目要求。但仍存在一些问题，如模块化程度不够高，采集速度过慢等，因个人能力有限，故需进行不断学习以不断优化项目，使之不仅能够完成项目需求，更能够带来舒适的使用体验和更强的扩展性、普适性和稳定性。

在做项目的过程中，我遇见了很多的问题，这些问题一时的确阻挡了我前行的脚步，但是通过查阅资料去解决这些问题，最终这些问题反而加强了我对项目的理解。项目的设计和开发不是一步完成的，而是经过多次的更新和迭代才逐步完善的。做完之后再回看自己最初做项目时不知所措的行为，才发现自己通过实践确实成长了很多。

在如今的大数据时代，数据成为了非常重要的资源，爬虫是一种很好的数据采集方式，但爬虫总是会涉及到一些隐私和安全性的一些问题，各大网站也对爬虫采取了一定的防范措施，因此使用web自动化技术模拟用户操作来代替爬虫实现数据的采集将显得非常的重要。通过这个项目，我了解并实现了数据的采集，对我之后参与人工智能与大数据项目提供了一些宝贵的经验。

通过做此次项目，增加了我的开发经验，也拓展了我的知识领域。之前的我拥有python和web基础，但是对web自动化涉及较少，因项目需求，故学习了web自动化的相关知识。通过学习，并及时将学习的成果转化为项目经验，不失为一种正确的学习方式，在此非常感谢老师和工程师们给予我项目实战的机会。