基于 pyquery 的 GitHub 爬虫及基于 selenium 的自动关注 GitHub 项目实现

许靖桐

一、 项目需求

1. 获得 Github trending 上的十个最热门项目，输出项目 stars 数，项目 fork 数，以及主要贡献者 id 到一个 csv 文件中。其中检索要求为 Spoken Language=English, Language=python, Date Range=This month。
2. 利用 Web Drivers/Restful API/GraphQL API，使用测试账号关注（watch）爬取到的十个项目，并选择“Releases only”选项。

二、 项目实现

1. 爬虫部分

spider.py 中包含了网络爬虫函数 spider()，该函数通过使用 pyquery 对网页进行解析，获得十个最热门项目的 star 数，fork 数，主要贡献者并将这些数据输出到一个 csv 文件中。该函数返回一个包含十个最热门项目 URL 的列表，以便下一步使用 webdriver 关注这些项目。

首先对于 GitHub 网站热门项目页面进行观察，发现网站可以根据项目语言，编写语言，以及日期范围进行检索，在进行检索时，网页 URL 会根据检索项的变化而改变，因此根据要求的选项进行检索，获得 URL，通过该 URL 可以直接检索到要求的热门项目。

通过对 GitHub 网页进行解析，可以发现每一个热门项目的内容都以一个<article>标签的形式出现，因此首先获得所有带有<article>标签的元素，然后对这些元素进行处理，分别根据元素 class 的名称获得项目名称的字段，URL，星星数，fork 数的字段，以及主要贡献者的 id。然后将这些数据保存在一个列表中，并创建 csv 文件，在文件名前加上保存日期，然后保存。

2. 自动关注部分

自动关注部分利用 Selenium 中的 webdriver 编写。该部分的代码在 webdriver.py 中。该部分包含三个函数，分别负责登录 GitHub，关注项目以及取消关注项目。在使用 Selenium 前，我对我使用的 Chrome 浏览器下载了相应的驱动并进行了配置。

首先，使用 webdriver 可以自动打开浏览器，并通过 URL 访问指定的网页，但这样访问 GitHub 时需要先登录，因此通过 github_login()函数，传入 GitHub 登录页 URL，然后输入测试账号的账号及密码，并控制 webdriver 找到登录按钮点击登录，这样就可以登录 GitHub 并完成后续操作。

然后，watch_project()函数可以通过传入的项目 URL 访问项目主页，并查找到主页上的“Watch”按钮，选择“Releases only”选项并点击，以实现关注项目。同

理，unwatch_project()函数可以实现取关项目的功能。

在 webdriver.py 的主函数中，首先通过 webdriver 打开 Chrome 浏览器，并爬取需求的数据，然后登录 GitHub 并关注这些项目。最后关闭浏览器。

三、 项目总结

在项目中，我使用了 pyquery 进行爬虫的编写，使用 Selenium 进行自动化测试，由于之前并没有使用这两种工具的经验，因此在项目编写中也遇到了一些困难。比如，在利用 pyquery 访问包含星星数结点时，由于星星数节点与 fork 数节点的标签除了超链接都相同，因此只能搜索所有类名为 muted-link d-inline-block mr-3 的节点并用 text()输出节点的内容，这样得到的内容是一个含有星星数和 fork 数的字符串，然后对这个字符串进行处理分别获得两项数据。同样在处理主要贡献者时，也遇到了无法准确定位节点的问题，只能通过逐步定位，然后利用查找的方式获得所有主要贡献者数据。在 webdriver 方面，我直接使用了输入账号密码的方式进行登录页面，这样虽然容易但是丧失了安全性，当然也可以使用控制台输入账号与密码，但这样的话又需要在控制台和打开的浏览器之间相互切换，比较麻烦。最理想的状态应该可以手动输入账号密码，然后通过 Selenium 判断是否登录，然后再进行下一步，但我并未实现这些功能。

我并没有为代码编写单元测试，但在测试方面，我能想到的测试包括对于网络状况的测试，对于爬虫爬取数据准确性的测试，对于写入数据准确性的测试，对于 Selenium 登录及关注是否成功的测试等。其中最重要的就是对于爬取数据准确性的测试。但是我对于如何测试这些数据有一些疑问，如果编写自动化测试，那么是否也需要利用爬虫获取相应的数据，这样的话，测试代码的准确性是否也需要再进行测试？这样就变成了无限循环下去的问题，目前我还没有想到如何对这些数据的准确性进行验证的方法。

在这个项目中，我也学到了很多之前不了解的知识，比如 pyquery 与 Selenium 的原理，其中 Selenium 的使用非常方便，令我对其产生了极大的兴趣。总的来说，我收获了许多。