

目录

1.自动登录微博	2
1.1 爬虫工具介绍.....	2
1.2 如何定位	2
1.3 验证码问题.....	3
1.4 判别登录成功	4
2.自动发布微博	4
2.1 如何定位	4
3.自动加关注.....	4
3.1 URL 分析	4
3.2 如何定位.....	4
4.自动评论	5
4.1 如何定位.....	5
4.1 流程图	6

一、自动登录微博

首先分析任务要求为编写微博自动化模块，要求完成自动发微博，自动关注以及自动评论。关键字为爬虫以及自动化，根据导师提示用 python 爬虫比较普遍，于是自行学习了一下如何用 python 进行爬虫；同时通过搜索如何用 python 实现自动化找到了 Selenium webdriver。

1.1 爬虫工具介绍

Selenium 是一个浏览器自动化操作框架。Selenium 主要由三种工具组成。第一个工具 SeleniumIDE，是 Firefox 的扩展插件，支持用户录制和回访测试。录制/回访模式存在局限性，对许多用户来说并不适合，因此第二个工具——Selenium WebDriver 提供了各种语言环境的 API 来支持更多控制权和编写符合标准软件开发实践的应用程序。最后一个工具——SeleniumGrid 帮助工程师使用 Selenium API 控制分布在一系列机器上的浏览器实例，支持并发运行更多测试。在项目内部，它们分别被称为“IDE”、“WebDriver”和“Grid”。

WebDriver 针对各个浏览器而开发，取代了嵌入到被测 Web 应用中的 JavaScript。与浏览器的紧密集成支持创建更高级的测试，避免了 JavaScript 安全模型导致的限制。除了来自浏览器厂商的支持，WebDriver 还利用操作系统级的调用模拟用户输入。WebDriver 支持 Firefox(FirefoxDriver)、IE (InternetExplorerDriver)、Opera (OperaDriver) 和 Chrome (ChromeDriver)。本次任务使用的是 Chromedriver。

确定了工具后，所需要调用的最重要的 API 为登陆界面，用户主页的 URL。

1.2 如何定位

登录的主要思路是驱动启动浏览器后，通过微博主页的 URL 进入登录页面，用 driver 的各种 find_element 方法找到账号密码输入框，键入账号密码进行登录。所以最为关键的一步是定位到账号密码以及登录按钮的位置。

```
<input id="loginname" type="text" class="W_input" maxlength="128" autocomplete="off" action-data="text=邮箱/会员帐号/手机号" action-type="text_copy" name="username" node-type="username" tabindex="1"> == $0
```

图 1.输入账号框的 HTML 源代码

根据源代码我们通过 id 定位到账号输入框，即 find_element_by_id(“loginname”);

```
<input type="password" class="W_input" maxlength="24" autocomplete="off" value action-type="text_copy" name="password" node-type="password" tabindex="2"> == $0
```

图 2.输入密码框的 HTML 源代码

根据源代码我们通过 name 定位到密码输入框，即 find_element_by_name(“password”);

```

▼ <div class="info_list login_btn"> == $0
  ▶ <a href="javascript:void(0)" class="W_btn_a
    btn_32px" action-type="btn_submit" node-type=
      "submitBtn" suda-data=
        "key=tblog_weibologin3&value=click_sign" tabindex=
          "6">...</a>

```

图 3. 登录键的 HTML 源代码

最后需要定位的是登录键，当账号密码通过 send_keys 方法输入完毕后，我们还需要 click 方法自动点击登录键才能完成登录。此处通过 classname 定位到登录键，即 find_element_by_classname(“login_btn”)。实际运行代码时，总会出现无法找到元素的 bug，通过搜索发现是网络状态不好导致页面没有加载完毕导致，于是引入了 time 库，在每一步操作后都等待 1s，成功解决了 bug。

1.3 验证码问题

在多次运行代码过程中发现一个问题：不经常登录的账号在点击登录键后会出现验证码。

图 4. 出现验证码

因为不是所有账号都会出现这个情况，所以首先需要判别是否出现验证码。判别方法依旧是通过其 HTML 源代码，若是能定位到验证码则说明出现验证码问题，没有定位到验证码说明无需考虑验证码问题。

```

<input type="text" class="W_input" maxlength=
  autocomplete="off" value="验证码" action-data=
    码" action-type="text_copy" name="verifycode"
      "verifycode" tabindex="3"> == $0

```

图 5. 验证码的 HTML 源代码

根据源代码通过 name 定位到验证码, 即 find_element_by_name(“verifycode”); 若判定返回 TRUE, 我们需要根据验证码图片手动输入验证码, 通过 send_keys 方法实现自动输入验证码并再次自动点击登录。

1.4 判别登录成功

在登录后我们需要确认是否登录成功, 思路是若登录成功我们会进入到微博主页, 观察主页的 URL 可知, 在主页的 URL 中会出现 home, 所以判别是否登录成功问题转化为判别 URL 中是否出现 home。



图 6.成功登录主页的 URL

此处我们可以通过 python 的 split 函数分割 URL, 若在最后一个 “/” 和第一个问号 “?” 前位置的字符为 home 即判定为登录成功。

二 . 自动发布微博

自动发布微博的思路跟输入账号密码并发送一样, 首先需要定位发布微博输入框以及发布按钮, 将文本内容通过 send_keys 方法自动输入后自动点击发布按钮。

2.1 如何定位

首先需要定位到发布微博的输入框, 我们先找到其 HTML 源代码, 并找到输入框的 xpath 路径, 通过 find_element_by_xpath 方法定位到输入框;

```
<textarea class="W_input" title="微博输入框" name node-type="textEl"
pic_split="1" range="0&0" style="height: 68px; margin: 0px; padding: 0px
border-style: none; border-width: 0px; font-size: 14px; overflow-wrap:
break-word; line-height: 18px; overflow: hidden; outline: none;">
</textarea> == $0
```

图 7.发布微博输入框的 HTML 源代码

接着根据发布键的源代码通过 class 定位到发布键, 并通过 click 方法自动点击发布按钮。

```
<a href="javascript:void(0);" class="W_btn_a btn_30px W_btn_a_disable"
diss-data="module=stissue" node-type="submit" title="发布微博按钮">发布
</a> == $0
```

图 8.发布键的 HTML 源代码

三 . 自动加关注

自动加关注我们首先要找到对方的微博主页 URL, 进入到主页后定位到关注按钮并通过 click 方法自动点击关注键。

3.1 URL 分析

通过观察可知个人主页的 URL 为 <https://www.weibo.com/+u>+个人 id, 所以我们只需要找到对方的个人 id 就可以成功进入其主页。



3.2 如何定位

进入主页后需要根据 HTML 源代码通过其 xpath 定位到关注键, 并通过 click 方法自动

点击关注键完成关注。

```
><a href="javascript:void(0);" suda-uatrac=  
"key=tblog_profile_v6&value=atten" suda-data=  
"key=tblog_attention_click&value=5628619262" action-type="follow"  
class="W_btn_c btn_34px">...</a> == $0
```

图 9.关注键的 HTML 源代码

四 . 自动评论

自动评论第一步同上需要先进入对方主页，然后分别定位评论键，评论输入框以及发布键，最后通过 click 方法自动点击完成自动评论。

4.1 如何定位

进入主页根据评论键 HTML 源代码通过其 xpath 定位，并根据 click 方法自动点击；



图 10.评论键

```
><span class="line S_line1" node-type="comment_btn_text">...  
</span> == $0
```

图 11.评论键的 HTML 源代码

点击评论键后继续根据源码通过 xpath 定位评论输入框和发布键，然后通过 send_keys 方法将评论文本自动输入并通过 click 方法自动点击发布按钮完成自动评论。

```
<textarea class="W_input" action-type="check" cols rows name  
node-type="textEl" style="margin: 0px; padding: 5px 2px 0px  
6px; border-style: solid; border-width: 1px; font-size: 12px;  
overflow-wrap: break-word; line-height: 18px; overflow: hidden  
outline: none; height: 23px;"></textarea> == $0
```

图 12.评论输入框 HTML 源代码

```
<a class="W_btn_a W_btn_a_disable" action-type="post" diss-  
data="module=scommlist&group_source=" href="javascript:  
void(0);" onclick="return false" node-type="btnText">评论</a>
```

图 13.发布评论键 HTML 源代码

