软件学院 WEB数据管理 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301127 | 姓名： 王娜 | | 班级： 2018级大数据 |
| 实验题目：实验二设计克服反爬虫障碍的爬虫 | | | |
| 实验学时：6h | | 实验日期：2021-4-15 | |
| 实验目的：  选择具有反爬虫障碍的网站为目标  分析网站的反爬虫障碍  编写爬虫软件，克服障碍获取相应的数据 | | | |
| 硬件环境： | | | |
| 软件环境：  WIN10  Jupyter notebook  Mysql | | | |
| 实验步骤与内容：  1.目标网站分析  1）网站名称 ：boss直聘网  URL：https://www.zhipin.com/job\_detail/?ka=header-job   1. 该网站采用反爬虫技术及展示网站的反爬虫存在  * 用户登录：     首先新用户注册一个账号，记住账号与密码。用户再次登录时需要填入账号与密码。   * 网站监测爬虫     在登入网站后，网站可以知道chrome是人为打开，还是代码启动的。   * Cookies     每个用户都有一个账号，用户登录后会在本地保存一个cookies凭证，再次登录时可以使用该cookies凭证登录。   * AJAX     登入网站后可见每一页列出来的职位只有30个，要想获取更多地职位，需要点击下一页进行翻页，此处使用了Ajax反爬技术。   * 验证码图片识别     在登录时需要进行图片验证，这是该网站反爬的一个重要手段。  2.所使用开发工具   1. 开发语言：python   2）使用的模块，工具：selenium  3.操作过程   1. 工作思路  * 用户登录：   用户首次登录使用selenium的find\_element\_by\_xxx()方法找到账号与密码输入框，再使用send\_keys方法模拟输入账号与密码。   * 网站监测爬虫：   因为网页的window.navigator.webdrive字段设置为true，网站通过这个字段可以知道chrome是人为打开，还是代码启动的。因此可以通过配置  option = webdriver.ChromeOptions()  option.add\_experimental\_option('excludeSwitches',['enable-automation'])  关闭开发者模式，网站就监测不出来爬虫了。   * Cookies：   在登录时，可以在selenium库中调用现成的方法：driver.get\_cookies()获取cookies；而加载网页时，只需要再调用driver.add\_cookies()即可。打开网页后再重新加载cookies并刷新，这样就实现了cookies免账号认证的功能。   * AJAX：     分析网页代码，找到网页中“下一页”的元素，使用selenium的click方法模拟点击该元素，就可以跳转到下一页继续爬取数据。  另外需要注意的是程序有时可能抛出找不到元素的错误，这是因为网页还未将所有元素加载完就进行了信息的抽取导致元素找不到，因此应该适当的调用time.sleep()方法等待。此处我使用了selenium的一个方法，driver.implicitly\_wait()隐式等待。该方法定义程序当未加载完成时就等待，直到超出了设定的最大时间限度，当页面加载完成时就开始爬取数据。   * 验证码图片识别   由于该网站的图片验证码为点选式验证码，需要进行文字识别与文字点触，使用传统的灰度图或者OCR等方法均没有效果。通过网上学习调研，此类验证码目前都是通过使用付费打码平台的API完成的。我需要做的就是将验证码图片提交给平台，然后平台返回识别结果在图片中的坐标位置，接下来我们再解析坐标模拟点击即可。由于平台的识别验证码是由人工完成的，所以比较费时，大约有几秒的等待时间。   1. 源代码分析  * 用户登录：   定位到账号与密码两个输入框，使用selenium的send\_keys()方法模拟用户输入账号与密码。输入后调用time.sleep()方法随机等待几秒。    结果：     * Cookies   定义save\_cookies()方法，通过调用driver的get\_cookies()方法获取cookies并保存到本地文件中，以便后续的登录。    定义add\_cookies（）方法，将本地存储的cookies文件中的cookies信息提取出来，用于用户登录。     * AJAX   定位“下一页”按钮，selenium模拟点击按钮，进行翻页并爬取下一页数据，如此循环。     * 验证码图片识别   获取验证码图片链接，提取该图片的二进制内容，将图片以文件的形式打开以便获得图片尺寸，进一步获取图片与浏览器标签大小的比例。发送图片给打码平台，返回关于关键字位置信息的坐标。再对位置坐标进行解析，并使用actionchains模拟点击坐标位置，进行点触验证。    结果：    定义save\_to\_mysql()方法，将获取的数据导入到mysql数据库对应的表里。    定义get\_data方法，通过selenium的find\_element\_by\_xxx()方法获取招聘信息的各种数据，如职位名称、地址、发布日期、薪酬、要求学历、公司名称、公司信息、职位标签等。同时，使用click()方法点击当前招聘信息，调用selenium的switch\_to\_window()方法转入该招聘信息的详情页，获取职位要求等详细信息；获取结束后再调用switch\_to\_window()方法返回列表页面。  设置webdriver，关闭开发者选项，做成selenium伪装以避免反爬虫。    设置最长隐形等待时间，加载cookies，并打开URL链接。    调用get\_data()方法获取每一页的数据，全部获取结束后调用click方法模拟点击“下一页”按钮，转入下一页后继续爬取数据，如此循环。当每个类型的职位都爬取了10页数据后跳出循环。    4.爬取数据   1. 保存格式  |  |  | | --- | --- | | Name | 职位名称 | | Address | 工作地址 | | Update | 招聘信息发布日期 | | Wage | 薪酬 | | Degree | 学历 | | Company | 公司名称 | | Company\_info | 公司信息 | | tag | 标签 | | detail | 职位具体要求 |     2）数据量大小  12039条关于软件工程、大数据、人工智能、网安的一线城市的职位招聘信息。每条属性有9个属性，总共11.5MB | | | |
| 结论分析与体会：  1.  2.  3. | | | |