Python Selenium 爬取裁判文书网：从登录到批量下载全流程自动化

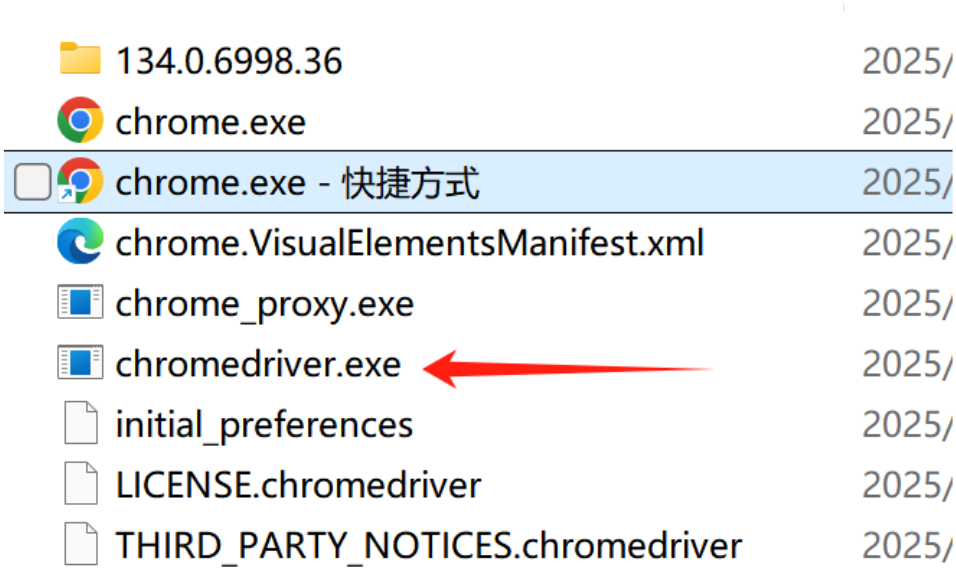
爬虫俱乐部@Stata and Python数据分析 2025年06月27日 10:03

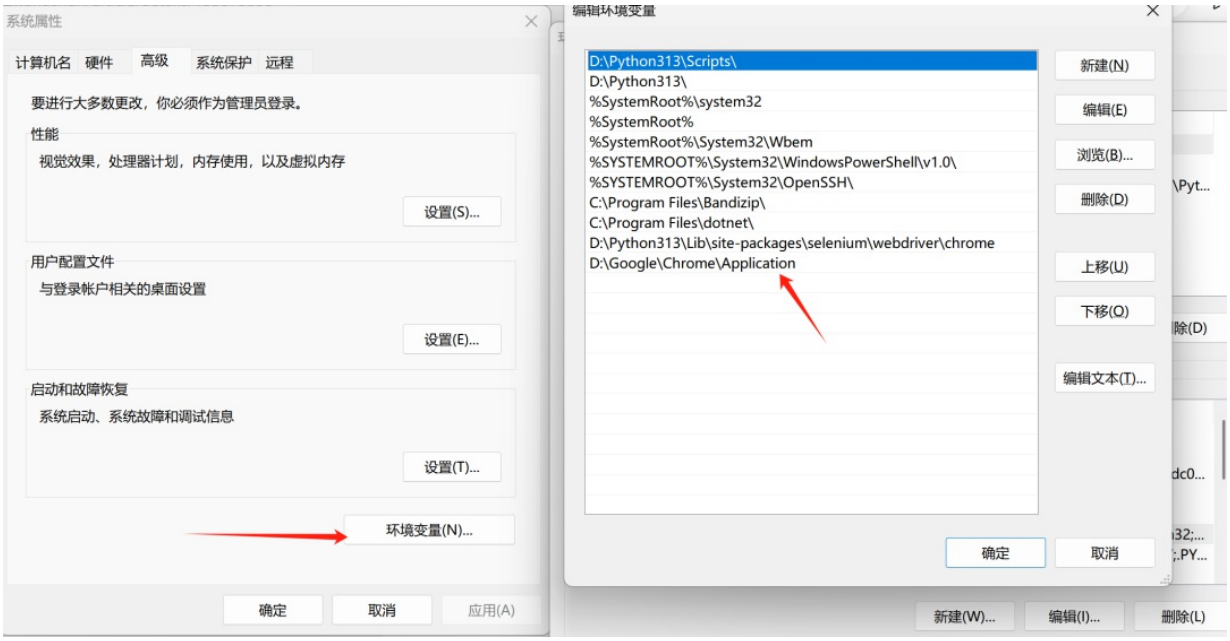
本文作者：蒋成阳，河南大学高级金融学院
本文编辑：张珍珍
技术总编：兰博文

在进行实证研究的过程中，数据的重要性毋庸置疑，优质的数据更是实现好文章的前提。然而，数据的来源多种多样，除了从二手数据、数据库中获得数据，还可以利用Python获取想要的数据。

一、环境准备：关于ChromeDriver

ChromeDriver 是一个由 Google 官方维护的工具，用于通过 WebDriver 协议控制 Chrome 或 Chromium 浏览器。从互联网下载和chrome版本相对应的ChromeDriver 驱动，下载完毕后，把驱动和chreome放在同一个文件夹，然后，在系统-高级设置-环境变量-系统变量/用户变量中-Path中添加ChromeDriver 驱动的路径，以便后续调用做准备，如下图操作。

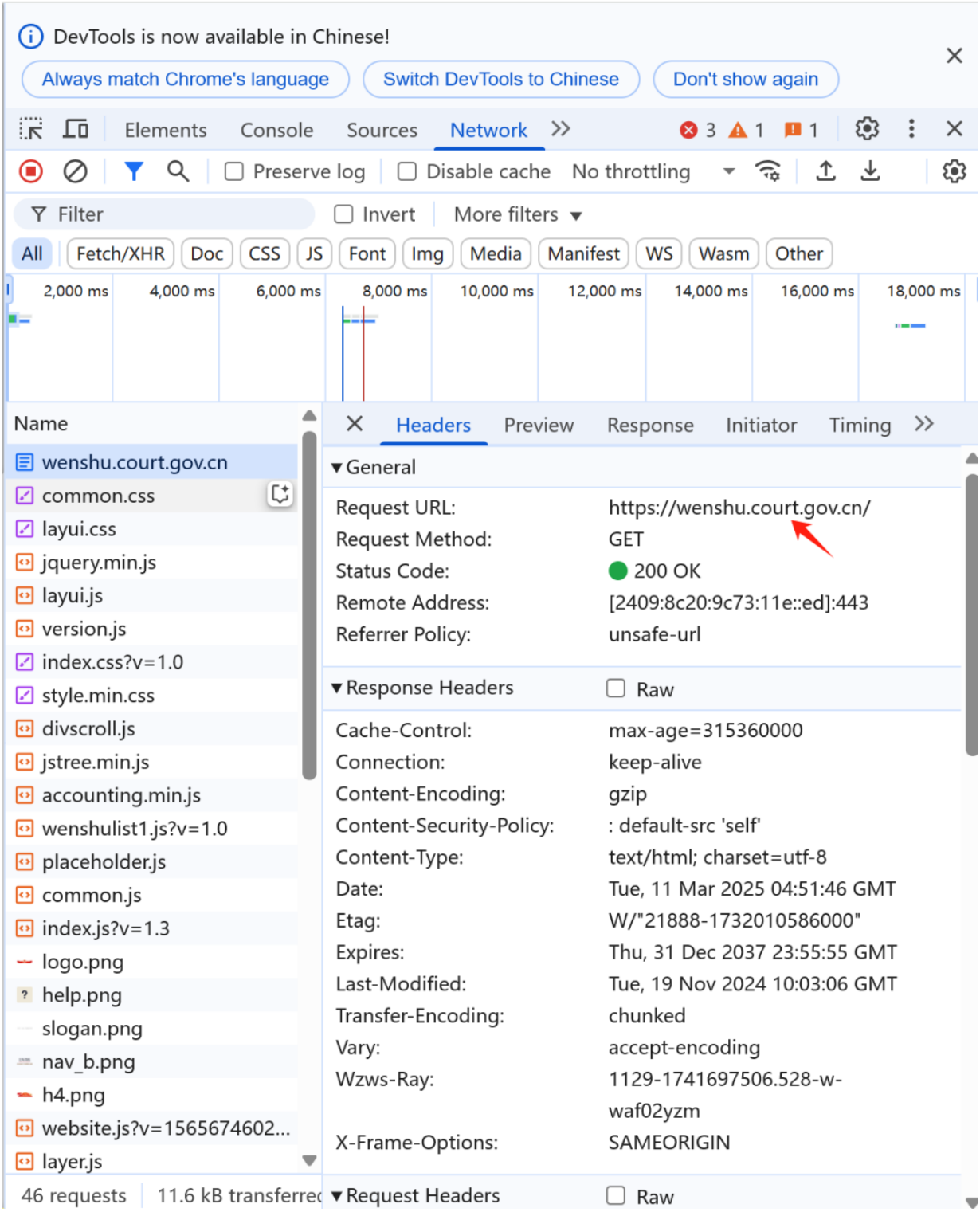




二、关于裁判文书网

中国裁判文书网是由最高人民法院建立的全国性司法公开平台，是全球最大的裁判文书数据库，集结国内各级法院的审判案件。因此，本文以裁判文书网为例，爬取相应的内容。

根据上图，可以观察到裁判文书网的构成。界面中包含刑事、民事、行政等各类案件，该网站的特点是需要登录注册，才能搜索想要的文书信息，第一步是获取网页的url，之后就可以操作爬取，如下图所示。



三、导入必要的库

代码如下：

```
from selenium import webdriver  # 导入Selenium WebDriver库from selenium.webdriver.common.by import By  # 导入By类用于元素定位from selenium.webdriver.support.select import Select  # 导入Select类用于处理下拉框import time  # 导入time模块用于时间控制
```

四、初始化Selenium配置

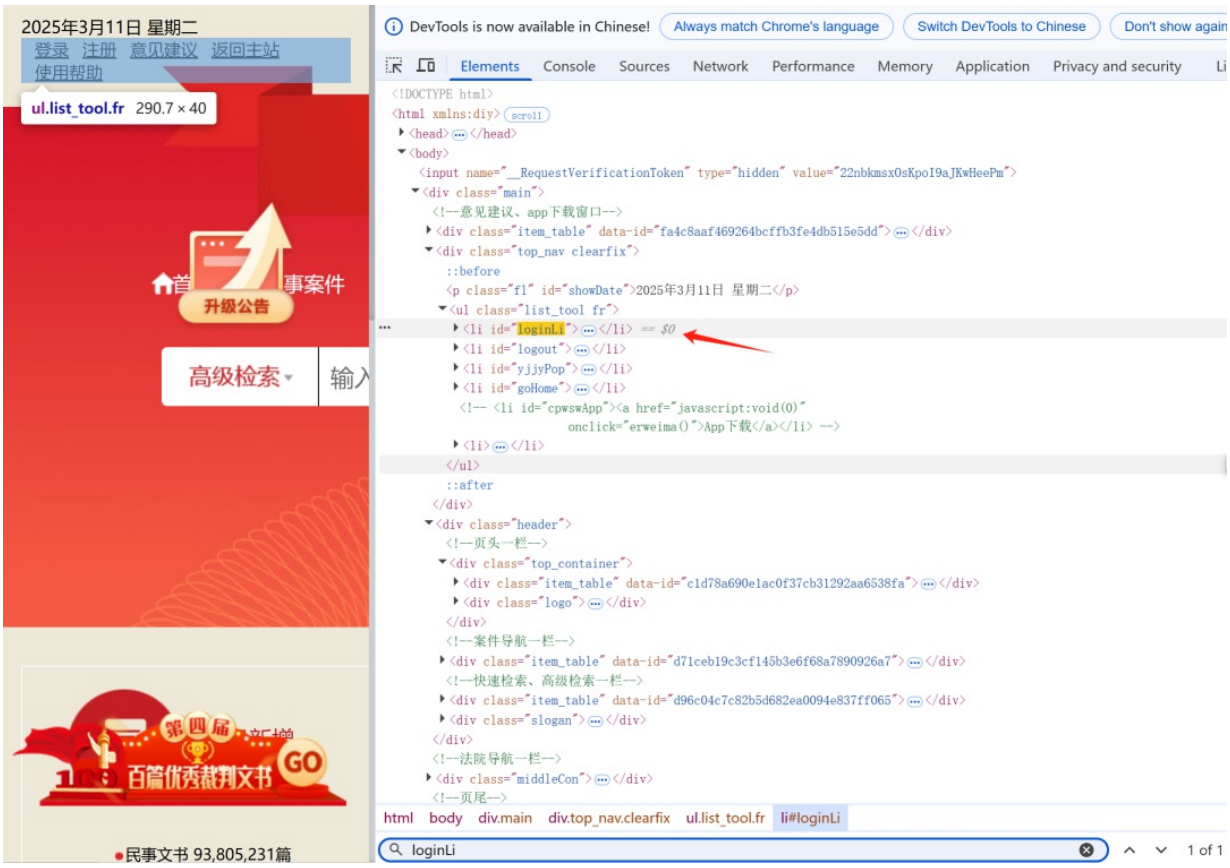
代码如下：

```
url = 'https://wenshu.court.gov.cn'  # 目标网站URLoption = webdriver.ChromeOptions()  # 创建Chrome浏览器配置对象option.add_argument('--start-maximized')  # 浏览器窗口最大化option.add_experimental_option('excludeSwitches', ['enable-automation'])  # 禁用浏览器自动化控制标识，降低反爬检测概率# 设置下载配置 prefs = {      'profile.default_content_settings.popups': 0,  # 禁用下载弹窗      'download.default_directory': 'D:\\裁判文书',  # 设置默认下载路径（注意路径转义）      'profile.default_content_setting_values.automatic_downloads': 1  # 允许自动下载      }option.add_experimental_option('prefs', prefs)  # 将配置应用到浏览器driver = webdriver.Chrome(options=option)  # 创建Chrome浏览器实例driver.maximize_window()  # 确保窗口最大化（与启动参数冗余但双重保险）driver.set_page_load_timeout(30)  # 设置页面加载超时时间为30秒driver.get(url)  # 打开目标网页
```

运行以上命令之后，将自动打开裁判文书网的界面（网速比较卡，页面加载不全），如下图所示。

五、设置登录流程

开启检查，获取登录图标按钮（LoginLi）所对应的XPath路径，并设定点击该按钮。根据XPath具体内容，按照以下代码，可以实现自动登录。



代码如下：

```
driver.find_element(By.XPATH, '//*[@id="loginLi"]/a').click()  # 点击登录按钮进入登录页面time.sleep(10)  # 等待页面加载（可优化为显式等待）# 切换到登录iframe（常见于表单嵌套场景）iframe = driver.find_elements(By.TAG_NAME, 'iframe')[0]driver.switch_to.frame(iframe)# 输入账号密码并提交username = driver.find_element(By.XPATH, '//*[@id="root"]/div/form/div/div[1]/div/div/div/input')username.send_keys('你的手机号')  # 输入自己注册的手机号time.sleep(3)  # 等待输入回显password = driver.find_element(By.XPATH, '//*[@id="root"]/div/form/div/div[2]/div/div/div/input')password.send_keys('你的密码.')  # 输入自己设立的密码time.sleep(2)  # 等待输入回显driver.find_element(By.XPATH, '//*[@id="root"]/div/form/div/div[3]/span').click()  # 点击登录按钮time.sleep(3)  # 等待登录跳转driver.switch_to.default_content()  # 切换回主文档内容
```

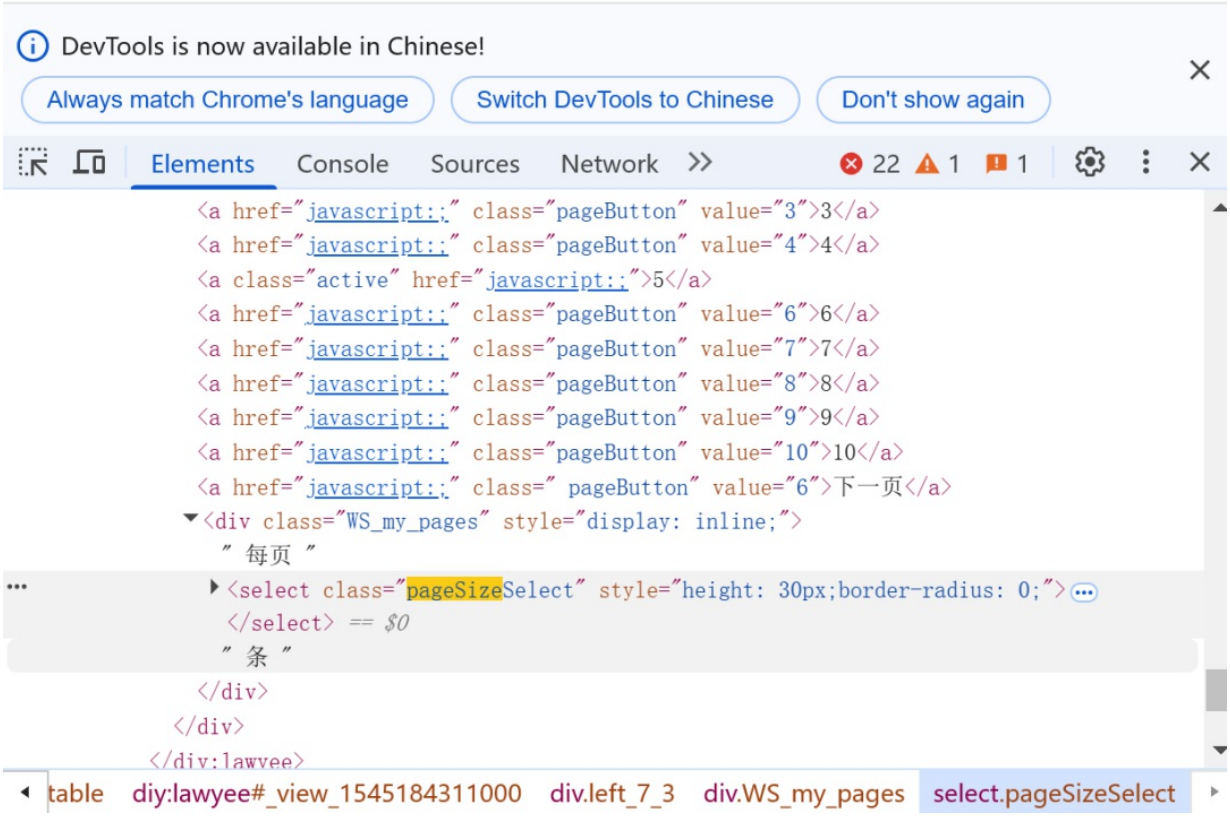
六、案件类型选择与检索设置

本期主要想爬取裁判文书网，“行政案件”中有关“公平竞争”的案件，于是可以设计下面代码：

```
driver.find_element(By.LINK_TEXT, '行政案件').click()  # 选择案件类型为"行政案件"time.sleep(10)  # 等待新窗口加载_lastWindow = driver.window_handles[-1]  # 获取最新打开的窗口句柄driver.switch_to.window(_lastWindow)  # 切换到新窗口# 高级检索设置keyword = driver.find_element(By.XPATH, '//*[@id="_view_1545034775000"]/div/div[1]/div[2]/input')keyword.send_keys('公平竞争')  # 输入检索关键词time.sleep(3)  # 等待输入完成driver.find_element(By.XPATH, '//*[@id="_view_1545034775000"]/div/div[1]/div[3]').click()  # 点击搜索按钮# 设置每页显示15条结果page_size_box = Select(driver.find_element(By.XPATH, '//*[@id="_view_1545184311000"]/div[8]/div/select'))page_size_box.select_by_visible_text('15')  # 选择下拉框中的"15"
```

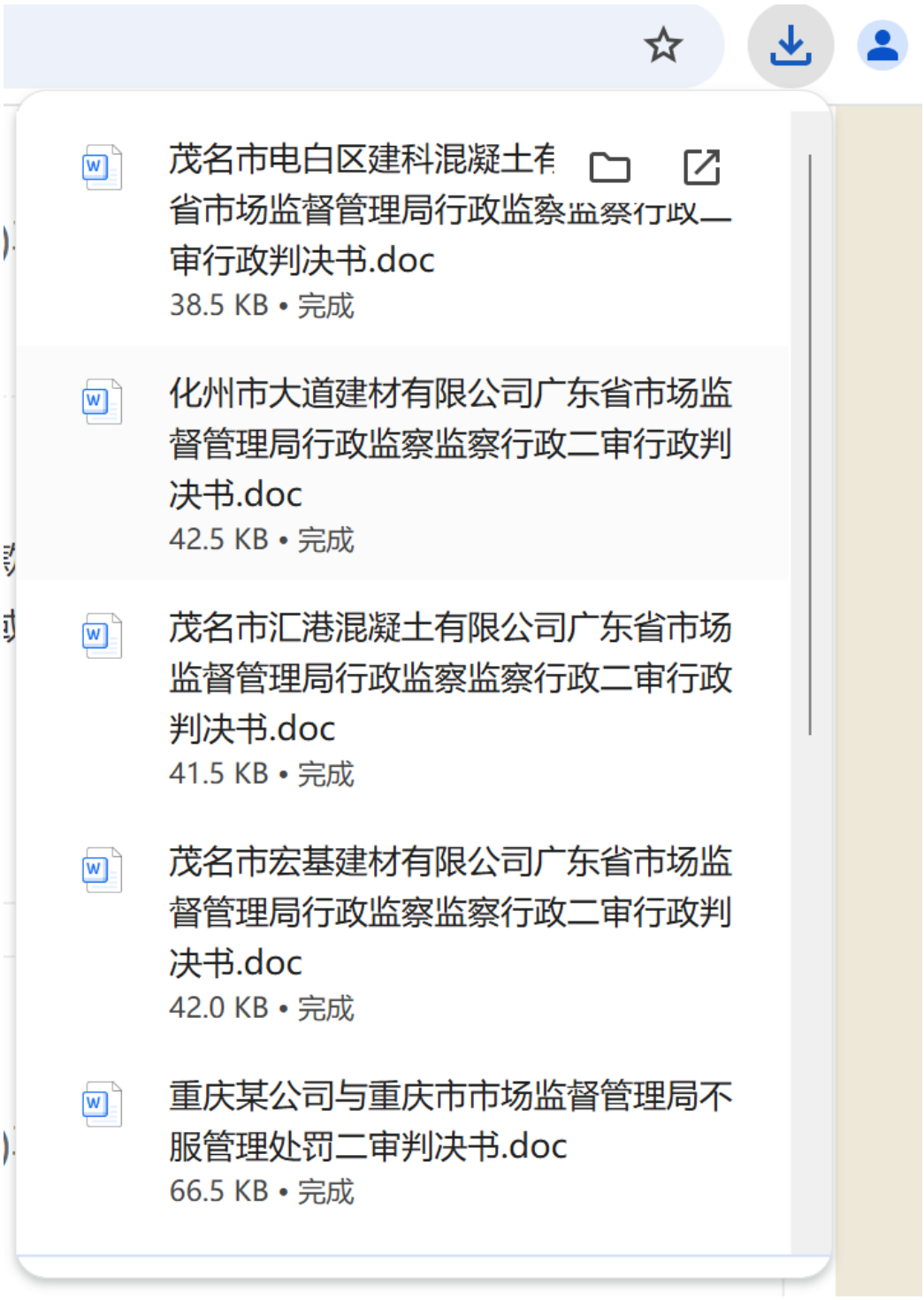
七、开始精确搜索，循环爬取

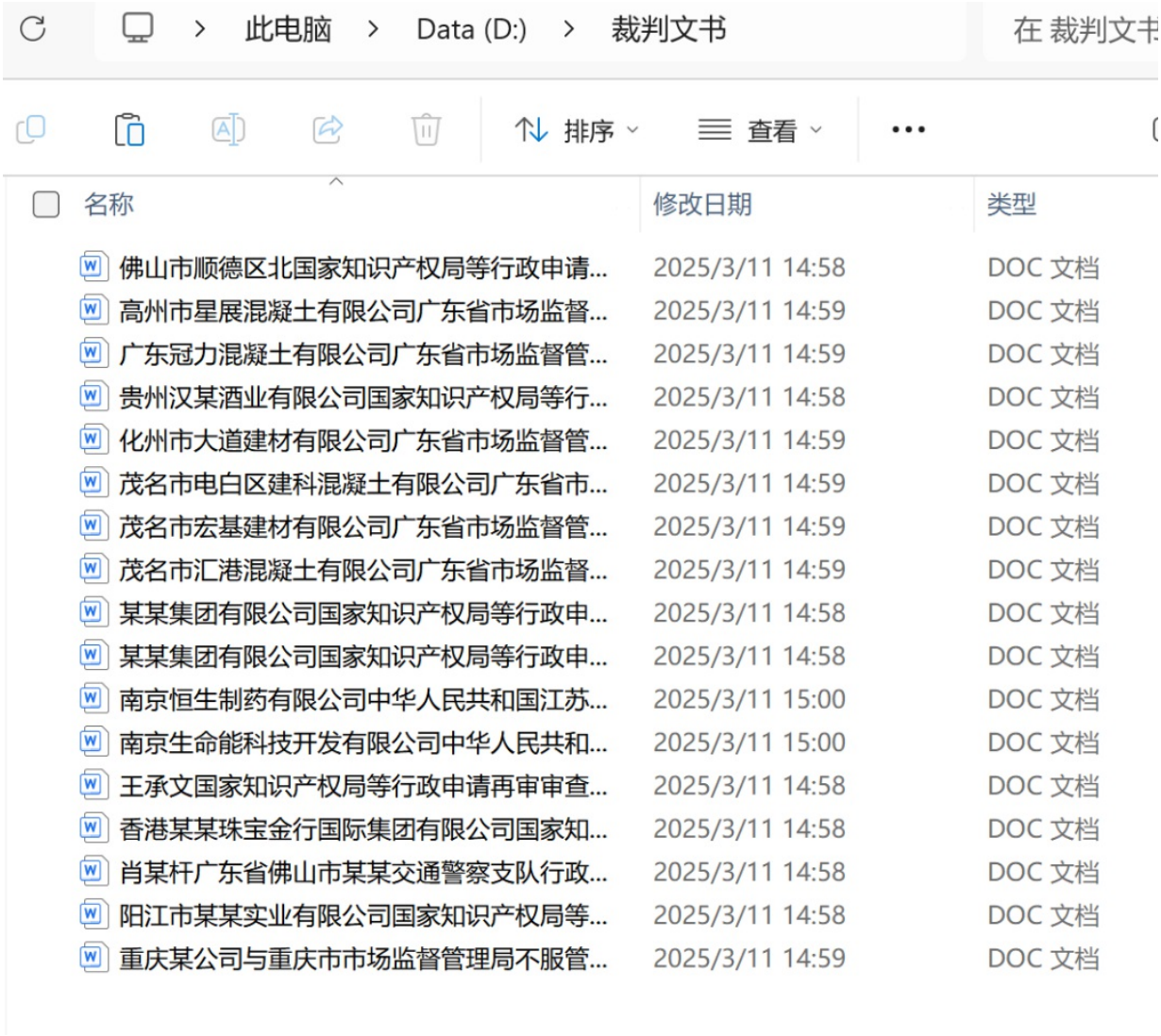
因裁判文书网反爬机制严格，为使得爬取顺利，本期就以两页内容，一页展示15条文书示例，按照上述方法，从检查中，可以找到页码PageSize的标签。




```
def test_exceptions(xpath):    """检测指定XPath是否存在元素"""    try:        driver.find_element(By.XPATH, xpath)        return True    except:        return False # 此处仅仅以两页示例page = 1  # 当前页码计数器max_pages = 2  # 设置最大爬取页数为2页while page <= max_pages:    time.sleep(5 + page/10)  # 动态等待时间（随页码递增）    # 遍历当前页15条结果    for i in range(15):        time.sleep(5 + i/10)  # 动态等待时间（随元素位置递增）                # 尝试两种可能的元素路径（应对页面结构变化）        event_xpath = f'//*[@id="_view_1545184311000"]/div[{i+3}]/div[6]/div/a[2]'        if test_exceptions(event_xpath):            driver.find_element(By.XPATH, event_xpath).click()  # 点击下载按钮        else:            event_xpath = f'//*[@id="_view_1545184311000"]/div[{i+3}]/div[5]/div/a[2]'            if test_exceptions(event_xpath):                driver.find_element(By.XPATH, event_xpath).click()  # 备用路径点击    # 下一页操作    time.sleep(5)  # 等待加载    try:        next_page = driver.find_element(By.LINK_TEXT, '下一页')  # 定位下一页按钮        next_page.click()  # 点击下一页    except:        print("没有更多页码，结束爬取")        break  # 无下一页时终止循环    driver.switch_to.default_content()  # 切换回主内容（避免iframe残留）    page += 1  # 页码计数器+1driver.quit()  # 关闭浏览器并释放资源
```

整理上述所有代码一次性运行，可以观察到，浏览器正在自动下载所需要爬取的裁判文书，并且保存在所设置的下载路径里（两分钟密集爬取17条文书），如下图所示。


茂名市电白区建科混凝土有
省市场监督管理局行政监察监察行政二
审行政判决书.doc
38.5 KB • 完成
化州市大道建材有限公司广东省市场监督管理局行政监察监察行政二审行政判决书.doc
42.5 KB • 完成
茂名市汇港混凝土有限公司广东省市场监督管理局行政监察监察行政二审行政判决书.doc
41.5 KB • 完成
茂名市宏基建材有限公司广东省市场监督管理局行政监察监察行政二审行政判决书.doc
42.0 KB • 完成
重庆某公司与重庆市市场监督管理局不服管理处罚二审判决书.doc
66.5 KB • 完成



八、总结

通过上述代码，基本实现了以下功能：

（1）登录：自动输入账号密码登录裁判文书网。

（2）高级检索：通过关键词“公平竞争”筛选行政案件。

（3）分页下载：每页15条，爬取2页内容并自动点击“下一页”。

（4）反检测配置：禁用自动化标识、设置下载路径等。

往期推文推荐

用正则表达式玩转混乱文本

爬虫俱乐部2025暑期Stata&Python编程训练营开始报名啦！

当Stata遇上周易：数据分析师的Cyber算命指南与玄学新副业

【Python CleverCSV】让CSV文件处理更便捷

Stata矩阵 —— 开启高效数据分析的魔法之门

当川普遇到GPT——TimeGPT对川普币价格的时间序列预测分析

一图解千言：从理性函数到浪漫曲线

爬虫实战：中基协私募基金数据爬取与可视化分析

Stata绘图秘籍：代码打造极简时钟

DeepSeek霸榜微博？用爬虫解读微博用户的情感密码

一图读懂：中国各省金融许可证地理分布
用Stata破解百年诅咒！每年到底有多少个"黑色星期五"?
利用Deepseek结合Stata创意绘图——多元素同心圆标识设计与实现
对中国知网高被引学者数据的评估探讨
Stata中reclink命令全解析，必学干货来袭！
【爬虫实战】文献阅读小助手
爬虫实战——stata抓取教育部文件（二）
除夕烟火映征程，回顾这一年的奋斗足迹
“已升升”送如意！

关于我们

微信公众号“Stata and Python数据分析”分享实用的Stata、Python等软件的数据处理知识，欢迎转载、打赏。我们是由李春涛教授领导下的研究生及本科生组成的大数据处理和分析团队。

武汉字符串数据科技有限公司一直为广大用户提供数据采集和分析的服务工作，如果您有这方面的需求，请发邮件到statatraining@163.com。