目录

一．下载路径

想要下载海报，第一步要做的就是设置好下载路径。除此之外，根据要求来看，我们是根据演员姓名下载其对应海报，所以还需要以演员姓名设置所对应的图片目录。

1.1 设置文件夹的路径

设置文件夹路径有两个思路：一是使用 os 库的 getwcd 方法获取当前路径并设置为下载路径，二是直接指定想要设置下载路径将其赋值给下载路径。

1.2 设置对应姓名的图片目录

我们需要实现的是搜索演员姓名下载其海报，所以我们在下载的文件夹中还需要设立对应演员姓名的图片目录以进行区分。图片目录只需要在下载路径基础上加上演员姓名，设置完成后我们需要判断对应演员姓名的图片目录是否存在，若不存在则在此路径下创建一个。

二．图片链接

下载海报最关键的一部分就是获取所有海报的 URL。获取海报图片的 URL 首先我们需要找到对应演员姓名的电影海报页 URL，然后再去获取海报图片的 URL。

2.1 搜索界面 URL 分析

图 1.电影海报页第一页 URL

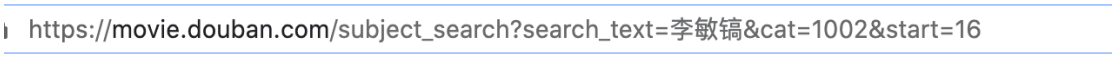

图 2.电影海报页第二页 URL

对比两页的 URL 可知，每页共有 15 张海报，url 区别在于最后”start=”后的数字，并且该数字从 1 开始，每次递增 15；除此之外，”search_text=”后面是演员姓名。综合以上的规律，搜索演员电影海报界面的 URL 为 'https://movie.douban.com/subject_search?search_text=' + name + '&cat=1002' + '&start=' + str(i)，所以我们可以用一个 for 循环遍历所有的电影海报页并使用 webdriver 自动化访问所对应页面。

2.2 工具介绍

自动化访问每一个电影海报页所需要的是 Selenium WebDriver。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，确定了自动化工具后，所需要调用的最重要的 API 为海报图片的 URL。

下面需要做的是找到所有的电影海报的 URL，并通过 request 库的 get 方法获取图片。获取 URL 的方法有很多，本文使用的方法是使用 etree.HTML 的方法将网页源代码由字符串格式转变为 HTML 文档，之后通过找到图片 xpath 根据 html.xpath 方法获取图片的 URL，这里找所有海报图片的 xpath 比较困难，通过查找，发现了一款工具：Xpath Helper。

google 插件 XPath Helper 可以支持在网页点击元素生成 xpath，整个抓取使用了 xpath、正则表达式、消息中间件、多线程调度框架（参考）。xpath 是一种结构化网页元素选择器，支持列表和单节点数据获取，他的好处可以支持规整网页数据抓取。
如果我们要查找某一个、或者某一块元素的 xpath 路径，可以按住 shift，并移动到这一块中，上面的框就会显示这个元素的 xpath 路径，右边则会显示解析出的文本内容，并且我们可以自己改动 xpath 路径，程序也会自动的显示对应的位置，可以很方便的帮助我们判断我们的 xpath 语句是否书写正确。

2.3 获取所有海报的 URL

本文思路首先使用 etree.HTML 方法将搜索页的源代码由字符串格式转变为 HTML 文本，之后需要找到包含海报图片 URL 的源代码。

```
<img src="https://img9.doubanio.com/view/photo/s_ratio_poster/public/p1411040772.webp" alt="虽然我也不懂&lrm; (2008)" class="cover"> == $0
```

图 3.包含海报图片 URL 的源代码

下一步就是要将源代码中的 URL 提取出来，这里用到 etree 中的 xpath()方法，所以我们需要找到这些海报图片的 xpath。使用 xpath helper，分别找到每一个图片对应的 xpath；

图 4.第一张图片的 xpath

图 5.第二张图片的 xpath

通过对比发现每一个图片的 xpath 变化的就是其中的那个数字，于是截取数字后面一段没有发生变化的 xpath 并进行验证，验证后发现输出结果为 15 个图片的 URL；对于海报标题所对应的 xpath 也采取同样方法。将所有的海报图片和海报标题的 URL 分别存储在 urls 和 titles 这两个列表中。

图 6.xpath 验证

三．下载图片

得到 URL 后，我们通过 requests 库请求图片然后以 wb 也就是二进制写模式打开图片，此时还需要设立图片的路径，我们在图片目录的基础上加上图片的 title 以及.jpg 为后缀设立图片路径，打开路径后将请求的图片内容写入。

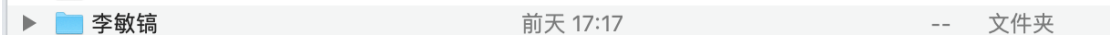

图 7.图片目录

图 8.图片下载成功效果


设置下载路径以及图片目录
分析并获取电影海报搜索页的URL
通过webdriver自动访问演员的每个电影海报页
通过xpath和etree获取每个图片的URL
通过requests库的get方法获取图片
以二进制形式写入设立的图片路径中完成下载

图 9.流程图