基于 Python 的豆瓣音乐数据爬虫的设计与实现

王英杰 毛红霞

四川大学锦城学院 计算机与软件学院 四川 成都 611731

【摘 要】：本文是基于 Python 来对某豆瓣音乐网站进行定向爬取网页数据的爬虫程序，现在是大数据的时代了，大家平时上网都会有种很明显的体验，你刚在一个网页搜索了某个东西，下一秒打开淘宝天猫就会发现主页在给推送相关的东西。这就是数据的力量，而网络爬虫就是我们对数据抓取很有力并且高效的一个工具了，所以如何使用网络爬虫也就变的十分重要了。今天就通过对豆瓣音乐排行榜的数据抓取来简要介绍网络爬虫的基本知识。之后如果想要统计最近最火的音乐榜单就可以通过网络爬虫去实现了。了解 xpath 语法删选数据的用法，最后详细介绍 Beautiful Soup 的用法。其中用到的核心库有 requests 网页请求库和 BeautifulSoup 网页数据爬取库。

【关键词】：图片爬取；xpath 语法；requests 网页请求库；BeautifulSoup 网页数据爬取库

引言

当今这个数据化的社会，网络上所存储的数据是我们远远超出我们预期的，如何高效的从这么多的网页资源中快速地找到自己所需要地资源是非常重要的。随着网络的发展。人们一般都是通过使用搜索引擎来查找自己需要的内容，但搜索引擎一般只能搜索到很多的网页，并且其中还会夹杂很多的广告推销等等，效率比较低下。而网络爬虫则是一种可以自动采集定向网络信息的程序，人们只有手中有一台电脑就可以手动自主编写爬虫程序来实现对自己所需要的某网站或是某网页的信息的定向收集，并且还可以将收集到的数据进行整理存储以达到更好的直观对比或者商业效果。本文是通过 Python 实现的一套定向爬取网页数据的爬虫程序，并将爬取结果整理写入数据库中。Python 中的 request 库和 BeautifulSoup 库在编写爬虫程序时十分方便快捷。

1 基本爬虫流程，如图 1 所示

①准备工作：首先准备工作就是看一下目标网页，怎么样去分析它，怎么样去查看它里面哪一些是我们想要的内容并且找到它。以及如何保证我们的程序的输出是一个能够有非常好的框架，并且能把所有的问题给提前预防和解决。②获取数据：通过 HTTP 库发起请求，模拟我们的浏览器去获取我们网页中的信息，最后得到一个网页的内容。③解析内容：获取到信息之后就要解析它，得到的内容可能是 HTML，json 等格式就可以用到页面解析库和正则表达式等④保存数据：可以保存为 txt 文本也可以保存到数据库，或者保存特定的格式文件。

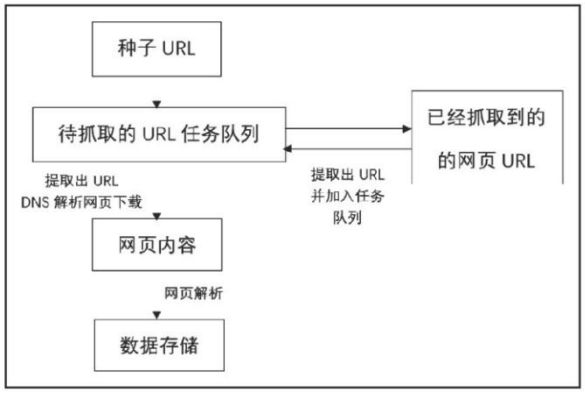

图 1 爬虫的基本流程

2 爬取案例实现

2.1.项目描述

该项目是为了编写一个网络爬虫程序，将豆瓣音乐网站上的歌曲排行榜信息爬取下来。爬取的信息字段包括：排名顺序，歌曲链接，歌曲名，歌手名，播放次数以及上榜时间。

2.2 爬取网站过程分析

我们打开豆瓣音乐首页 https://music.douban.com/，在豆瓣音乐首页的左上侧点击排行榜标签，就会跳转到豆瓣音乐排行榜网页 https://music.douban.com/chart/。如图 2 所示，其中可以看到每一首歌的排名顺序，歌曲名，歌手名以及播放次数和上榜时间。

图 2 豆瓣音乐排行榜

2.3 页面分析

按下 F12 打开网页源代码然后从源代码中找出我们所需要的标签内容，如图 3 所示；如果想要快捷找到自己想要的标签属性可以点击左下角的小箭头，后再将鼠标移到网页上自己想找寻的内容下方将自动显示其源代码，如图 4 所示。

图 3 所需的标签内容

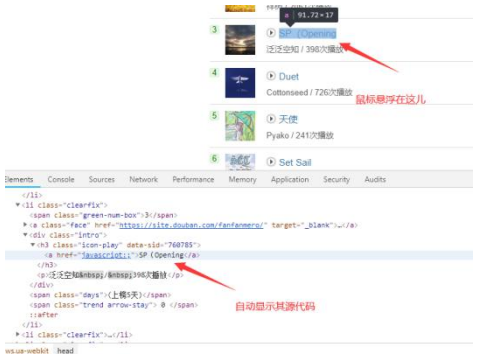

图 4 自动显示源代码

2.4 发起网络请求

Import requests #导入 requests 库

（①.Requests 是用 python 语言基于 urllib 编写的，采用的是 Apache2 Licensed 开源协议的 HTTP 库。与 urllib 相比，Requests 更加方便，可以节约我们大量的工作。②.如果未安装 requests 库可以按住 Windows 加 r 键打开 cmd 命令提示符，输入 pip install requests 然后回车。）

url = “https://music.douban.com/chart”

ua = {

“user-agent” :“Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.25 Safari/537.36 Core/1.70.3776.400 QQBrowser/10.6.4212.400”

}

（创建一个 ua 参数然后赋值给 headers，user-agent 是浏览器的一个标识，可以在网页源代码，点击 Network 选中数据包就可以查看该数据的 user-agent，如图 8 所示；添加了 user-agent 后可以将你的身份从爬虫转变成浏览器，这样网页就不会阻止你访问数据。）

result = requests.get(url , headers = ua).content.decode()

（requests 后面具体使用 get 方法或者 post 方法是由网址的数据包本身决定的，F12 打开网页源代码，点击 Network 然后选中数据包就可以查看该数据的 request method，如图 5 所示。）

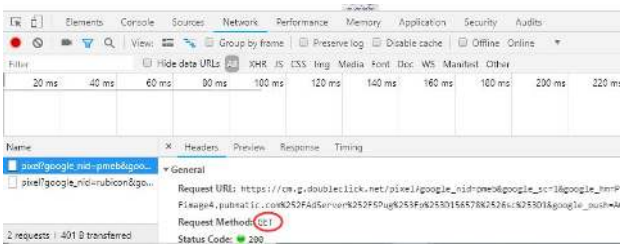

图 5 request method 查询

print (result)

2.5 xpath 删选数据

在得到的数据中找到我们需要的数据，发现基本都是在 div 标签内，只后就需要用 xpath 语言（XPath 使用路径表达式来选取 XML 文档中的节点或节点集。节点是通过沿着路径 (path) 或者步 (steps) 来选取的。）对数据进行删选。

2.5.1 xpath 选取节点的具体步骤

基于当前的网页源代码界面选择 Elements，网页右上方打开插件 Xpath Helper，它的作用是可以基于当前页面去写一个 xpath 语法然后用这个语法去得到我们需要的数据。具体的路径表达式用法。如图 6 所示；例如这个项目需要的歌

曲名，歌手名，播放次数等数据就可以使用//div[@class=”intro”]来得到。

选取节点

XPath 使用路径表达式在 XML 文档中选取节点。节点是通过沿着路径或者 step 来选取的。

下面列出了最有用的路径表达式：

表达式	描述
nodename	选取此节点的所有子节点。
/	从根节点选取。
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。

图 6 路径表达式

2.5.2 将 xpath 导入 python 中

先导入 lxml 库，为什么要导入 lxml 库呢？因为 lxml 库中有方法可以将 HTML 文档转换成 XML 文档，(XML 是用来传输和存储数据的和 json 很相似，HTML 是用来显示和展现数据的。)转化之后我们才可以用 XPath 去解析文档内容。如果你还没有安装 lxml 库，可使用 pip 安装：pip install lxml。然后就可以使用 lxml.etree 来处理 xml 文档，因为 python 它是不支持 xpath 的，但 lxml 库可以用来处理 html 文件，虽然我们结果得到的数据格式很像 HTML，但其实是字符串格式的。所以我们需要 etree.HTML()将得到的数据转化为 HTML 格式。

2.5.3 配合 Beautiful Soup 让爬取更容易

Beautiful Soup 和 XPath 的差异：

相同点：Beautiful Soup 和 XPath 一样都是 python 网页解析器，可以用来解析 HTML 和 XML，并从中提取数据。

独有特点：①Beautiful Soup 的它提供给用户的 api 十分简单，类似于 python 中的函数，但是在这简单的 api 背后它有自己强大的功能，比如 Beautiful Soup 和 XPath 一样都可以从文档中提取数据，但是 Beautiful Soup 还可以修改文档中的数据，这是 XPath 做不到的。所以说这就是 Beautiful Soup 强于 XPath 的地方。②Beautiful Soup 它还支持多种解析器，Beautiful Soup 在使用中我们需要去安装解析器，这些解析器一共有三种类型，第一种是 python 标准库中的 HTML 解析器，另外两种是第三方的解析器（之后会详细介绍）。③Beautiful Soup 能帮我们自动的实现编码转换，当我们传入文档时它会给我们转换成 Unicode 类型，但是当它把文档处理完输出的时候，它会将文档输出成 UTF-8 类型，这就大大的减轻了我们的工作量。

Beautiful Soup 解析器：

①python 标准库中的 HTML 解析器，优势：python 内置标准库无需下载，速度适中，容错率高。劣势：会因为版本的不同导致容错率下降（python 2.7.3or3.2.2 前的版本容错率低）

②lxml 解析器，优势：速度快，容错率较高。劣势：需要安装 C 语言库。

③html5lib 解析器，优势：容错率最高，以浏览器的方式解析文档，生成 HTML5 格式的文档。劣势：速度慢，不依赖外部扩展。

推荐使用 lxml 解析器，因为 lxml 解析器能提供 lxml HTML 和 lxml XML 两种不同的解析器。这两种文档的解析都有相同的特点就是非常的快，而且文档的容错能力的话也是非常强的。

Beautiful Soup 选择器：

①节点选择器：作为获取数据的基本方法。②方法选择器：用来查找，定位元素。③CSS 选择器：同方法选择器一样。因此我们在使用这三种选择器时要互相配合灵活运用，才能够从 HTML 文档中提取出我们想要的数据。

3 结果展示

通过爬虫我们成功的抓取了，豆瓣音乐排行榜前十歌曲的歌曲排名，歌曲链接，歌曲名，演唱者以及播放次数和上榜时间，并且用分隔符号将他们分割开来了。

结语

本文对豆瓣音乐排行榜进行爬取并展示，并对一些网站的反爬技术使用对应的反反爬策略,不仅不会增加网站服务

器的压力,还可以保证爬取数据的高效性和稳定性;并且遵守了网站的 Robots 协议。在大数据时代,爬虫行业必将乘风而起，Python 网络爬虫更是其中的翘楚。因为 Python 语言具有跨平台，开发速度快，语言简单等特点在网络爬虫的使用中十分的方便,并且 Python 语言可以通过第三方请求库得到返回值的内容，然后通过 XPath 和 BeautifulSoup 等 Python 过滤技术快速匹配和提取网页中的图像和文本数据，有了这些第三方库我们不仅能精确地找到网页中我们所需的数据，还能自动化快速的将这些数据保存下来，极大的减少了查找数据所需的时间。使用基于 Python 的网络爬虫不仅爬取速度快，其语言的简洁性也大大地缩短了爬取时间，这在当代这个高速运行的社会显得尤为重要。

参考文献：

[1] 张誉曜,陈媛媛. 基于 Python 下的爬虫综述及应用[J]. 中国新通信. 2019(06)

[2] 李琳. 基于 Python 的网络爬虫系统的设计与实现[J]. 信息通信. 2017(09)

[3] 王碧瑶.基于 Python 的网络爬虫技术研究[J].数字技术与应用,2017(5):76.

[4] 魏程程.基于 Python 的数据信息爬虫技术[J]. 电子世界. 2018(11)

[5] 郭丽蓉. 基于 Python 的网络爬虫程序设计[J]. 电子技术与软件工程. 2017(23)