用 Python 抓取公号文章保存成 HTML

原创 極光 Python技术 3天前

上次为大家介绍了如果用 Python 抓取公号文章并保存成 PDF 文件存储到本地。但用这种方式下载的 PDF 只有文字没有图片，所以只适用于没有图片或图片不重要的公众号，那如果我想要图片和文字下载下来怎么办？今天就给大家介绍另一种方案——HTML。

需解决的问题

其实我们要解决的有两个问题：

1. 公众号里的图片没有保存到 PDF 文件里。
2. 公众号里的一些代码片段，尤其那些单行代码比较长的，保存成 PDF 会出现代码不全的问题。
3. PDF 会自动分页，如果是代码或图片就会出现一些问题。

综上问题，我觉得还是把公众号下载成网页 HTML 格式最好看，下面就介绍下如何实现。

功能实现

获取文章链接的方式，和上一篇下载成 PDF 的文章一样，依然是通过公众号平台的图文素材里超链接查询实现，在这里我们直接拿来上一期的代码，进行修改即可。首先将原来文件 `gzh_download.py` 复制成 `gzh_download_html.py`，然后在此基础进行代码改造：

```
# gzh_download_html.py
# 引入模块
import requests
import json
import re
import time
from bs4 import BeautifulSoup
import os

# 打开 cookie.txt
with open("cookie.txt", "r") as file:
    cookie = file.read()
cookies = json.loads(cookie)
url = "https://mp.weixin.qq.com"
#请求公号平台
response = requests.get(url, cookies=cookies)
# 从url中获取token
token = re.findall(r'token=(\d+)', str(response.url))[0]
# 设置请求访问头信息
headers = {
    "Referer": "https://mp.weixin.qq.com/cgi-bin/appmsg?t=media/appmsg_edit_v2&action=edit&isNew=1&t
```

```
    "Host": "mp.weixin.qq.com",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like G
}

# 循环遍历前10页的文章
for j in range(1, 10, 1):
    begin = (j-1)*5
    # 请求当前页获取文章列表
    requestUrl = "https://mp.weixin.qq.com/cgi-bin/appmsg?action=list_ex&begin="+str(begin)+"&count=
    search_response = requests.get(requestUrl, cookies=cookies, headers=headers)
    # 获取到返回列表 Json 信息
    re_text = search_response.json()
    list = re_text.get("app_msg_list")
    # 遍历当前页的文章列表
    for i in list:
        # 目录名为标题名，目录下存放 html 和图片
        dir_name = i["title"].replace(' ','')
        print("正在下载文章：" + dir_name)
        # 请求文章的 url ，获取文章内容
        response = requests.get(i["link"], cookies=cookies, headers=headers)
        # 保存文章到本地
        save(response, dir_name, i["aid"])
        print(dir_name + "下载完成!")
    # 过快请求可能会被微信问候，这里进行10秒等待
    time.sleep(10)


```

好了，从上面代码可以看出，主要就是将原来的方法 `pdfkit.from_url(i["link"], i["title"] + ".pdf")` 改成了现在的方式，需要用 `requests` 请求下文章的 URL，然后再调用保存文章页面和图片到本地的方法，这里的 `save()` 方法通过以下代码实现。

调用保存方法

```
#保存下载的 html 页面和图片
def save(search_response,html_dir,file_name):
    # 保存 html 的位置
    htmlDir = os.path.join(os.path.dirname(os.path.abspath(__file__)), html_dir)
    # 保存图片的位置
    targetDir = os.path.join(os.path.dirname(os.path.abspath(__file__)),html_dir + '/images')
    # 不存在创建文件夹
    if not os.path.isdir(targetDir):
        os.makedirs(targetDir)
    domain = 'https://mp.weixin.qq.com/s'
    # 调用保存 html 方法
    save_html(search_response, htmlDir, file_name)
    # 调用保存图片方法
    save_file_to_local(htmlDir, targetDir, search_response, domain)

```

```
# 保存图片到本地
def save_file_to_local(htmlDir,targetDir,search_response,domain):
    # 使用lxml解析请求返回的页面
    obj = BeautifulSoup(save_html(search_response,htmlDir,file_name).content, 'lxml')
    # 找到有 img 标签的内容
    imgs = obj.find_all('img')
    # 将页面上图片的链接加入list
    urls = []
    for img in imgs:
        if 'data-src' in str(img):
            urls.append(img['data-src'])
        elif 'src=""' in str(img):
            pass
        elif "src" not in str(img):
            pass
        else:
            urls.append(img['src'])

    # 遍历所有图片链接，将图片保存到本地指定文件夹，图片名字用0，1，2...
    i = 0
    for each_url in urls:
        # 跟据文章的图片格式进行处理
        if each_url.startswith('//'):
            new_url = 'https:' + each_url
            r_pic = requests.get(new_url)
        elif each_url.startswith('/') and each_url.endswith('gif'):
            new_url = domain + each_url
            r_pic = requests.get(new_url)
        elif each_url.endswith('png') or each_url.endswith('jpg') or each_url.endswith('gif') or eac
            r_pic = requests.get(each_url)
        # 创建指定目录
        t = os.path.join(targetDir, str(i) + '.jpeg')
        print('该文章共需处理' + str(len(urls)) + '张图片，正在处理第' + str(i + 1) + '张……')
        # 指定绝对路径
        fw = open(t, 'wb')
        # 保存图片到本地指定目录
        fw.write(r_pic.content)
        i += 1
        # 将旧的链接或相对链接修改为直接访问本地图片
        update_file(each_url, t, htmlDir)
        fw.close()

    # 保存 HTML 到本地
    def save_html(url_content,htmlDir,file_name):
        f = open(htmlDir+"/"+file_name+'.html', 'wb')
        # 写入文件
        f.write(url_content.content)
        f.close()
        return url_content

```

```
    # 修改 HTML 文件,将图片的路径改为本地的路径
    def update_file(old, new,htmlDir):
         # 打开两个文件，原始文件用来读，另一个文件将修改的内容写入
        with open(htmlDir+"/"+file_name+'.html', encoding='utf-8') as f, open(htmlDir+"/"+file_name+
            # 遍历每行，用replace()方法替换路径
            for line in f:
                new_line = line.replace(old, new)
                new_line = new_line.replace("data-src", "src")
                 # 写入新文件
                fw.write(new_line)
        # 执行完，删除原始文件
        os.remove(htmlDir+"/"+file_name+'.html')
        time.sleep(5)
        # 修改新文件名为 html
        os.rename(htmlDir+"/"+file_name+'_bak.html', htmlDir+"/"+file_name+'.html')

```

好了，上面就是将文章页面和图片下载到本地的代码，接下来我们运行命令 `python gzh_download_html.py` ，程序开始执行，打印日志如下：

```
$ python gzh_download_html.py
正在下载文章：学习Python看这一篇就够了！
该文章共需处理3张图片，正在处理第1张……
该文章共需处理3张图片，正在处理第2张……
该文章共需处理3张图片，正在处理第3张……
学习Python看这一篇就够了！下载完成!
正在下载文章：PythonFlask数据可视化
该文章共需处理2张图片，正在处理第1张……
该文章共需处理2张图片，正在处理第2张……
PythonFlask数据可视化下载完成!
正在下载文章：教你用Python下载手机小视频
该文章共需处理11张图片，正在处理第1张……
该文章共需处理11张图片，正在处理第2张……
该文章共需处理11张图片，正在处理第3张……
该文章共需处理11张图片，正在处理第4张……
该文章共需处理11张图片，正在处理第5张……
该文章共需处理11张图片，正在处理第6张……
该文章共需处理11张图片，正在处理第7张……
```

现在我们去程序存放的目录，就能看到以下都是以文章名称命名的文件夹：

进入相应文章目录，可以看到一个 `html` 文件和一个名为 `images` 的图片目录，我们双击打开扩展名为 `html` 的文件，就能看到带图片和代码框的文章，和在公众号看到的一样。

总结

本文为大家介绍了如何通过 Python 将公号文章批量下载到本地，并保存为 HTML 和图片，这样就能实现文章的离线浏览了。当然如果你想将 HTML 转成 PDF 也很简单，直接用 `pdfkit.from_file(xx.html,target.pdf)` 方法直接将网页转成 PDF，而且这样转成的 PDF 也是带图片的。

老规矩，兄弟们还记得么，**右下角的“在看”点一下**，如果感觉文章内容不错的话，记得分享朋友圈让更多的人知道！

【代码获取方式】

识别文末二维码，回复：200331