跳过
查看详情
Toggle TesterHome

其他测试框架 使用 Python 将 HTML 转成 PDF

其他测试框架 使用 Python 将 HTML 转成 PDF

狂师 · 2017年04月30日 · 最后由 陈恒捷 回复于 2017年04月30日 · 4264 次阅读

目录

背景

很多人应该经常遇到在网上看到好的学习教程和资料但却没有电子档的，心里顿时痒痒，
下述指导一下大家，如何将网站上的各类教程转换成 PDF 电子书。

关键核心

- 主要使用的是wkhtmltopdf的Python封装—【pdfkit】

环境安装

- python3系列
- pip install requests
- pip install beautifulsoup4
- pip install pdfkit
- 如果是liunx系，则 sudo yum intsall wkhtmltopdf
- 如果是windows系，则下载稳定版的 wkhtmltopdf 进行安装，安装完成之后把该程序的执行路径加入到系统环境 $PATH 变量中

牛刀小试

一个简单的例子:

```
import pdfkit pdfkit.from_url('http://google.com', 'out.pdf')
pdfkit.from_file('test.html', 'out.pdf')
pdfkit.from_string('Hello!', 'out.pdf')
```

你也可以传递一个url或者文件名列表:

```
pdfkit.from_url(['google.com', 'yandex.ru', 'engadget.com'], 'out.pdf')
pdfkit.from_file(['file1.html', 'file2.html'], 'out.pdf')
```

也可以传递一个打开的文件:

```
with open('file.html') as f:
    pdfkit.from_file(f, 'out.pdf')
```

实例代码实现

如将自强学堂中的django教程，生成一个pdf文件

```
#coding=utf-8
from __future__ import unicode_literals
import os,sys,re,time
import requests,codecs
from bs4 import BeautifulSoup
from urllib.parse import urlparse
import pdfkit
import platform
requests.packages.urllib3.disable_warnings()

system=platform.system()
print(sys.getdefaultencoding())

str_encode='gbk' if system is 'Windows' else 'utf-8'
print(str_encode)

html_template = """
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
</head>
<body>
{content}
</body>
</html>

"""


if not os.path.exists(os.path.join(os.path.dirname(__file__),'html')):
    os.mkdir(os.path.join(os.path.dirname(__file__),'html'))
```

```
url_list=[]
start_url='http://www.ziqiangxuetang.com/django/django-tutorial.html'

# s=requests.session()
# html_doc=s.get('{}'.format(start_url),verify=False).content

# soup = BeautifulSoup(html_doc,'html.parser')
# print(soup.prettify())

def get_url_list(url):
    """
    获取所有URL目录列表
    :return:
    """
    last_position = find_last(url, "/") + 1
    tutorial_url_head = url[0:last_position]
    domain = get_domain(url) + "/"
    print(domain)

    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")
    urls = []
    for a in soup.find_all("a"):
        href = str(a.get('href'))
        result = href.find('/')
        if result == -1:
            url = tutorial_url_head + href
        else:
            url = domain + href
        if 'django' in url:
            urls.append(url)
    return urls


def find_last(string, char):
    last_position = -1
    while True:
        position = string.find(char, last_position + 1)
        if position == -1:
            return last_position
        last_position = position


def get_domain(url):
    r = urlparse(url)
    return r.scheme + "://" + r.netloc


def parse_url_to_html(url,name):
    """
    解析URL，返回HTML内容
    :param url:解析的url
    :param name: 保存的html文件名
    :return: html
    """
    try:
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')
        # 正文
        body = soup.find_all(class_="w-col l10 m12")
        h = str(body)
        html = h[1:-1]
        html = html_template.format(content=html)
        html = html.encode("utf-8")
        title=soup.title.get_text()
        print(url)
        with open('{}/{}'.format(os.path.join(os.path.dirname(__file__),'html'),name), 'wb') as f:
```

```
            f.write(html)
        return '{}/{}'.format(os.path.join(os.path.dirname(__file__),'html'),name)
    except Exception as e:
        print(e)


def save_pdf(htmls, file_name):
    """
    把所有html文件保存到pdf文件
    :param htmls:  html文件列表
    :param file_name: pdf文件名
    :return:
    """
    options = {
        'page-size': 'Letter',
        'margin-top': '0.75in',
        'margin-right': '0.75in',
        'margin-bottom': '0.75in',
        'margin-left': '0.75in',
        'encoding': "UTF-8",
        'custom-header': [
            ('Accept-Encoding', 'gzip')
        ],
        'cookie': [
            ('cookie-name1', 'cookie-value1'),
            ('cookie-name2', 'cookie-value2'),
        ],
        'outline-depth': 10,
    }
    pdfkit.from_file(htmls, file_name, options=options)

def main():
    start = time.time()
    urls = get_url_list(start_url)
    htmls = [parse_url_to_html(url, str(index) + ".html") for index, url in enumerate(urls)]
    print(htmls)
    try:
        save_pdf(htmls, 'cralwer_{}.pdf'.format(time.strftime('%Y_%m_%d_%H_%M_%S')))
    except Exception as e:
        print(e)
    for html in htmls:
            os.remove(html)
    total_time = time.time() - start
    print(u"总共耗时：{0:.2f}秒".format(total_time))

main()
```

大概思路

- 先传入一个起始站点的url，本例以自强学堂为例，http://www.ziqiangxuetang.com/django/django-tutorial.html
- 然后，通过爬虫获取所有含django的url地址，存放在一个列表中，然后再依次获取url，解析各个url中的正文body内容，通过人工分析，各个url正文Body对应的class为w-col l10 m12，所以只需要爬取w-col l10 m12的内容即可。
- 将获取到的正文内容存放在html文件中，最终返回一个含所有html文件地址的列表htmls。
- 通过pdfkit.from_file接收一个htmls列表,生成对应pdf文件。

常见问题

- **IOError: 'No wkhtmltopdf executable found'**
  确保 wkhtmltopdf 在你的系统路径中（$PATH），会通过 configuration进行了配置 (详情

看上文描述)。 在Windows系统中使用where wkhtmltopdf命令 或 在 linux系统中使用 which wkhtmltopdf 会返回 wkhtmltopdf二进制可执行文件所在的确切位置.

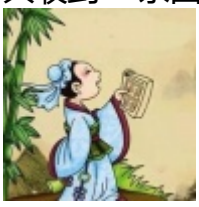
- **IOError: ‘Command Failed’**
  如果出现这个错误意味着 PDFKit不能处理一个输入。你可以尝试直接在错误信息后面直接运行一个命令来查看是什么导致了这个错误 （某些版本的 wkhtmltopdf会因为段错误导致处理失败

- **正常生成，但是出现中文乱码**
  在html中加入

参考

志军的项目: https://github.com/lzjun567/crawler_html2pdf

「原创声明：保留所有权利，禁止转载」
2 个赞 举报
TesterHome 为用户提供「保留所有权利，禁止转载」的选项。 除非获得原作者的单独授权，任何第三方不得转载标注了「原创声明：保留所有权利，禁止转载」的内容，否则均视为侵权。 具体请参见TesterHome 知识产权保护协议。
如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！
打赏支持
共收到 **2** 条回复 时间 点赞



大海 #1 · 2017年04月30日

这个挺实用的，mark

陈恒捷 #2 · 2017年04月30日

很实用的技巧，赞~

需要 登录 后方可回复, 如果你还没有账号请点击这里 注册。

相关话题

- 【我是一本正经的广告】2020 MTSC-北京大会征稿啦！
- 使用 Python 将 HTML 转成 PDF
- Python 与游戏测试 (小工具篇)
- selenium 执行测试用例多线程并发 BeautifulReport 报告中查看日志显示无，大神们知道怎么添加吗？

- Python Web 自动化测试
- python+uiautomator 自动义测试框架

作者

会员
mikezhou (狂师)
第 7574 位会员 / 2016-02-26
金蝶 @ 广州

Appium 中文文档小组

PPmoney

新秀群

健康部落

欢迎关注公众号: 【测试开发技术】
2 个赞

共收到 **2** 条回复
打赏支持

有新回复！ 点击这里立即载入

关于 / 活跃用户 / 中国移动互联网测试技术大会 / 反馈 / Github / API / 帮助推广
TesterHome 移动测试社区，由众多移动测试工作者维护，致力于推进国内测试技术。Inspired by RubyChina

友情链接 WeTest腾讯质量开放平台 / InfoQ / SegmentFault / 测试窝 / 百度测试吧 / IT大咖说
简体中文 / 正體中文 / English

UCLOUD SendCloud