Scrapy 框架实战应用 ，爬取 20000 张妹子图 。

27315

文章来源 ： 公号【从零开始学爬虫】，欢迎读者们关注~

之前小编曾发过一篇爬取优美网图片的文章，好多小伙伴私聊反馈说图片质量不好

，说让找点好看的妹子。好吧，我承认是小编自己的锅。这次选取了一个小伙儿伴推荐的网站，网站名称为妹子图，地址：http://www.meizitu.com/。这次的网站图片质量很高哦~我们先来看看网站截图。

怎么样，质量是不是很高？那么这么高质量的网站图片，想不想要？别急，看完本文，你也可以随意爬取类似的图片网站。

之前的文章我们采用的requests库，（请见拙作这一定是学爬虫的你脑中闪现过的第一道邪念！），这次我们不能使用同样的方法再来一次，那样就没有意义了，学习的意义在于不断进步。这次我们主要使用scrapy框架。

关于scrapy，这里不做过多介绍，贴一下百度百科对scrapy的简介：“Scrapy，Python开发的一个快速、高层次的屏幕抓取和web抓取框架，用于抓取web站点并从页面中提取结构化的数据。Scrapy用途广泛，可以用于数据挖掘、监测和自动化测试。” 总之就是一个非常好用的框架，可以帮助我们快速的做爬虫开发。

【网页分析】

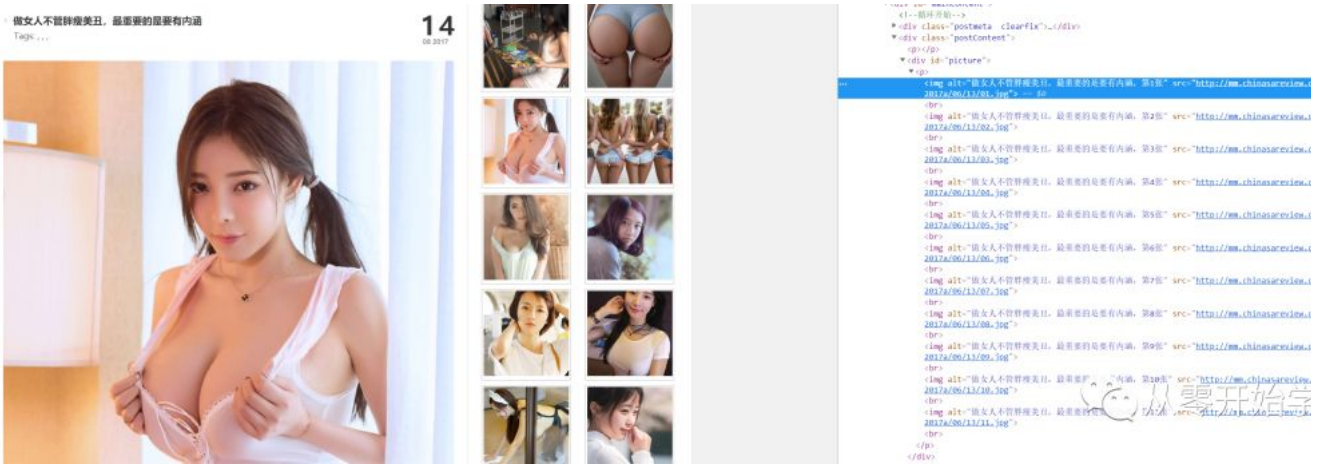
首先，我们打开目标网站，使用Chrom自带的调试功能，查询图片列表页面的信息。网页列出了很多图集，我们就将该网页暂时称为图集列表页面。在图集列表页面选择某一图集，分析其网页构成（见下图）。可见网页图集的地址是存储在一个class = ‘pic’的div下方的a标签中。a标签的herf属性即为图集页面的URL。所以我们可以使用CSS选择器 .pic a::attr(href) 来选取detail_url信息（图集地址）。

图集列表URL ，图片可放大查看

在得到图集地址之后，我们还要获取“下一页”的图集列表页面，重复用爬虫文件不断提取每一个页面中的detail_url信息。同样的方法，我们在定位“下一页”的网页构成（见下图）。可见“下一页”的链接是存储在id = "wp_page_numbers" 的div下面的 a 标签中。我们可以根据标签的text属性 = “下一页”来定位该标签，在获取其herf属性。 这里我们选用Xpath选择器，提取下一页url的信息为//*[@id="wp_page_numbers"]/ul/li/a[contains(text(), "下一页")]/@href 。

“下一页”链接 ，图片可放大查看

在得到图集列表和下一页的URL后，我们需要对图片集中的图片进行分析。同样的方法，我们确认图片地址（见下图）。图片的URL存储在class="postContent"的div下面的img标签下，src属性即为url地址。CSS选择器为 .postContent p img::attr(src)。



图片存储地址 ，图片可放大查看

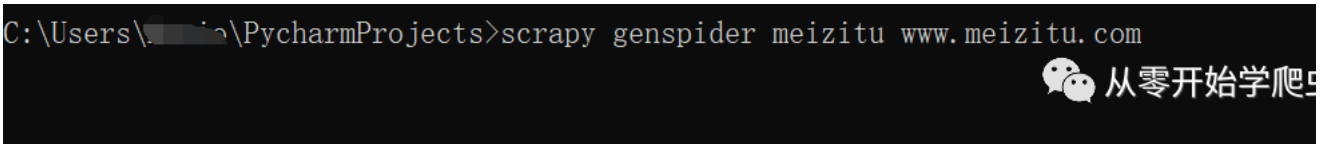
友情提示：在提取图片URL的时候，这里有个坑哦~ 希望大家能动手试一试，不知道有什么坑的，请继续往下看。

【创建项目】

在分析完网页构成之后，我们创建一个爬虫项目。通过CMD进入命令行，在适当的存储位置使用startproject命令创建项目。scrapy会自动创建一些文件。

【创建爬虫】

项目创建完成后，使用genspider命令创建爬虫模板。



【spider编写】

项目和爬虫模板创建完成后，进入项目，选择我们刚才创建的爬虫文件。在这里，我们需要用parse来解析图集列表网页，用parse_meinv来解析图集详情中的图片地址，在获取到图片的地址之后，我们将其写入一个image_url.txt的文件中。也许你会有疑问，为什么我们需要保存在文件中，不直接再写一个方法保存图片呢。这里就涉及到上文所说的“坑”了。我们可以观察一个图片的url。

http://mm.chinasareview.com/wp-content/uploads/2017a/06/13/01.jpg

上面的是一个图片的URL，有没有发现什么？没错，域名变了！！！因为我们创建爬虫的时候指定了allowed_domains = ['www.meizitu.com']，所以这里如果同一爬虫爬取这个图片地址，会报错。（本人知识有限，不知道说的对不对，如有错误希望有大佬帮忙指正，谢谢~）所以我们需要将URL保存到本地文件中，然后在通过另一个爬虫文件去爬取文本中的URL。

具体代码如下：

```
# -*- coding: utf-8 -*-

import scrapy

class MeizituSpider(scrapy.Spider):
```

```
    name = 'meizitu'
    allowed_domains = ['www.meizitu.com']
    start_urls = ['http://www.meizitu.com/a/more_1.html']
    def parse(self, response):
        detail_url = response.css('.pic a::attr(href)').extract()
        for url in detail_url:
            yield scrapy.Request(url,callback=self.parse_meinv)
                next = response.xpath('//*[@id="wp_page_numbers"]/ul/li/a[contains(text(), "下一页")]/@href').extract_first()
        if next:
            next_url = "http://www.meizitu.com/a/" + next
            yield scrapy.Request(next_url,callback=self.parse)
        else:
            pass
    def parse_meinv(self,response):
        with open('./image_url.txt','a') as f :
            image_urls = response.css('.postContent p img::attr(src)').extract()
            for image_url in image_urls:
                f.write(image_url+',')
```

【第二个爬虫文件】

在通过第一个爬虫文件将图片URL保存到本地之后，我们创建第二个文件，将image_url.txt中的URL图片保存到本地。这里还是使用genspider命令来创建，需要注意域名的变化。

```
C:\Users\[redacted]\PycharmProjects>scrapy genspider save_image mm.chinasareview.com
```


文件内容如下：

```
# -*- coding: utf-8 -*-

import scrapy

import os

class SaveImageSpider(scrapy.Spider):

    name = 'save_image'

    allowed_domains = ['mm.chinasareview.com']

    with open('./image_url.txt','r') as f :

        start_urls = f.read().split(',')

    if not os.path.exists('../image'):

        os.mkdir('./image')

    else:

        print('文件夹已经存在')

    def parse(self, response):

        name = ((response.url).split('uploads')[1]).lstrip('/').replace('/','-')

        file_path = './image/'+name

        if os.path.exists(file_path):

            print("%s已经存在"%file_path)

        else:

            with open(file_path,'wb') as f:

                f.write(response.body)
```

【start文件】

最后，为了方便运行爬虫，不用每次都在cmd中手动敲命令，创建了一个start文件。运行爬虫时需要将不用的爬虫注销哦~

```
# -*- coding: utf-8 -*-

from scrapy import cmdline


cmdline.execute(['scrapy','crawl','meizitu'])


# cmdline.execute(['scrapy','crawl','save_image'])
```

【源码】

在小詹公号回复关键词 ：妹子 ，即可获取代码和妹子图 ~ ~

往期推荐：

1. *这一定是学爬虫的你脑中闪现过的第一道邪念！*
2. *卷积神经网络（CNN）学习笔记*
3. *基于tensorflow的手写数字识别 。*

欢迎点赞和转发分享

▲长按关注我们

个人微信：python_jiang

文章转载自公众号

从零开始学爬虫

从零开始学爬虫