# Python爬虫实战之爬取百度贴吧帖子

大家好，上次我们实验了爬取了糗事百科的段子，那么这次我们来尝试一下爬取百度贴吧的帖子。与上一篇不同的是，这次我们需要用到文件的相关操作。

## 本篇目标

1. 对百度贴吧的任意帖子进行抓取
2. 指定是否只抓取楼主发帖内容
3. 将抓取到的内容分析并保存到文件

## 1. URL格式的确定

首先，我们先观察一下百度贴吧的任意一个帖子。

比如：http://tieba.baidu.com/p/3138733512?see_lz=1&pn=1，这是一个关于NBA50大的盘点，分析一下这个地址。

> `http://` 代表资源传输使用http协议
> `tieba.baidu.com` 是百度的二级域名，指向百度贴吧的服务器。
> `/p/3138733512` 是服务器某个资源，即这个帖子的地址定位符
> `see_lz`和`pn`是该`URL`的两个参数，分别代表了只看楼主和帖子页码，等于1表示该条件为真

所以我们可以把URL分为两部分，一部分为基础部分，一部分为参数部分。

例如，上面的URL我们划分基础部分是 http://tieba.baidu.com/p/3138733512，参数部分是 ?see_lz=1&pn=1

## 2. 页面的抓取

熟悉了URL的格式，那就让我们用urllib2库来试着抓取页面内容吧。上一篇糗事百科我们最后改成了面向对象的编码方式，这次我们直接尝试一下，定义一个类名叫BDTB(百度贴吧)，一个初始化方法，一个获取页面的方法。

其中，有些帖子我们想指定给程序是否要只看楼主，所以我们把只看楼主的参数初始化放在类的初始化上，即init方法。另外，获取页面的方法我们需要知道一个参数就是帖子页码，所以这个参数的指定我们放在该方法中。

综上，我们初步构建出基础代码如下：

```python
#-*-coding:utf8-*-
#created by 10412

import urllib
import urllib2
import re

#百度贴吧爬虫类
class BDTB:

    #初始化，传入基地址，是否只看楼主的参数
    def __init__(self, baseUrl, seeLZ):
        self.baseURL = baseUrl
        self.seeLZ = '?see_lz=' + str(seeLZ)

    #传入页码，获取该页帖子的代码
    def getPage(self, pageNum):
        try:
            url = self.baseURL + self.seeLZ + '&pn=' + str(pageNum)
            request = urllib2.Request(url)
            response = urllib2.urlopen(request)
            print response.read()
            return response
        except urllib2.URLError, e:
            if hasattr(e, "reason"):
                print u"连接百度贴吧失败,错误原因",e.reason
                return None

baseURL = 'http://tieba.baidu.com/p/3138733512'
bdtb = BDTB(baseURL, 1)
bdtb.getPage(1)
```

运行代码，我们可以看到屏幕上打印出了这个帖子第一页楼主发言的所有内容，形式为HTML代码。


## 3. 提取相关信息

### 1)提取帖子标题

在浏览器中审查元素，或者按F12，查看页面源代码，我们找到标题所在的代码段如下:

```
<h3 class="core_title_txt pull-left text-overflow  " title="纯原创我心中的NBA2014-2015赛季现役50大" style="width: 416px">纯原创我心中的NBA2014-2015赛季现役50大</h3>
```

所以我们要提取`<h3>`中的内容，因为一开始可以查看整个界面的原代码，查看里面含有`<h3>`标签的不止一个。所以需要写正则表达式来匹配，如下：

```
<h3 class="core_title_txt.*?>(.*?)</h3>
```

然后，我们可以写个获取标题的方法

```
# 获取帖子标题
    def getTitle(self):
        page = self.getPage(1)
        pattern = re.compile('<h3 class="core_title_txt.*?>(.*?)</h3>', re.S)
        result = re.search(pattern, page)
        if result:
            # print result.group(1)  #测试输出
            return result.group(1).strip()
        else:
            return None
```

### 2）提取帖子页数

同样地，帖子总页数我们也可以通过分析页面中的共?页来获取。

```
<li class="l_reply_num" style="margin-left:8px"><span class="red" style="margin-right:3px">4784</span>回复贴，共<span class="red">36</span>页</li>
```

所以我们的获取总页数的方法如下

```
#获取帖子一共有多少页
def getPageNum(self):
    page = self.getPage(1)
    pattern = re.compile('<li class="l_reply_num.*?</span>.*?<span.*?>(.*?)</span>',re.S)
    result = re.search(pattern,page)
    if result:
        #print result.group(1)  #测试输出
        return result.group(1).strip()
    else:
        return None
```

### 3）提取正文内容

审查元素，可以看到百度贴吧每一层楼的主要内容都在

标签里面，所以我们可以写如下的正则表达式

```
<div id="post_content_.*?>(.*?)</div>
```

所以提取正文内容的方法：

```
#获取每一层楼的内容,传入页面内容
def getContent(self,page):
    pattern = re.compile('<div id="post_content_.*?>(.*?)</div>',re.S)
    items = re.findall(pattern,page)
    for item in items:
        print item
```

运行截图如下：

很多媒体都在每赛季之前给球员排个名，我也有这个癖好……，我会尽量理性的分析球队地位，<a href="http://www.baidu.com/s?wd=个人能力&ie=gbk&tn=SE_hldp00990_u6vqbx10" cl
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/w%3D580/sign=cb6ab1f8708b4710ce2ffdc4f3ccc3b2/06381f30e924b899d8ca30e16c061d950b7bf671.jpg" pic_ext="jpeg" p
纯手打，更新就慢，大家慢慢看
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/ae0a19d8bc3eb135cfe70162a41ea8d3fc1f44c3.jpg" pic_ext="jpeg" pic_type="0" width="560" height="315">
今天再更新一个就回家啦
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/614a20a4462309f7136012cb700e0cf3d6cad6d8.jpg" pic_ext="jpeg" pic_type="0" width="450" height="293">
搞定今天更新完毕 准备下班了
晚上偷偷顶一个<img pic_type="1" src="http://tb2.bdstatic.com/tb/editor/images/face/i_f26.png?t=20140602" class="BDE_Smiley" height="30" width="30">
开更今天的了！
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/20f41bd5ad6eddc4c86a8e213bdbb6fd5366335e.jpg" pic_ext="jpeg" pic_type="0" width="424" height="381">
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/b282d158ccbf6c81e8290a32be3eb13532fa40c9.jpg" pic_ext="jpeg" pic_type="0" width="450" height="299">
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/03f5e0fe9925bc31088723425cdf8db1ca137015.jpg" pic_ext="jpeg" pic_type="0" width="459" height="331">
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/e7177f3e6709c93d04942d2d9d3df8dcd0005481.jpg" pic_ext="jpeg" pic_type="0" width="550" height="366">
顶一顶，难道写着写着成单机了？
吃饭去了 下午来更吧
下午回来继续更新了
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/012eb9389b504fc2c3ddc5dfe7dde71191ef6d6d.jpg" pic_ext="jpeg" pic_type="0" width="560" height="315">
上个厕所再回来更新下一个
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/e7177f3e6709c93d396a7d2d9d3df8dcd000544b.jpg" pic_ext="jpeg" pic_type="0" width="450" height="449">
再顶顶
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/33950a7b02087bf4a0652833f0d3572c10dfcf5b.jpg" pic_ext="jpeg" pic_type="0" width="327" height="313">
再顶一顶，写得好辛苦，别又变成单机了
话说现在除了一个人都没有人来说高了，低了呢，好奇怪
<img class="BDE_Image" src="http://imgsrc.baidu.com/forum/pic/item/15c79f3df8dcd100fb179efb708b4710b8122f64.jpg" pic_ext="jpeg" pic_type="0" width="399" height="600">

可以看到有很多的换行符和图片符，既然出现这样的情况，那肯定不是我们想要的结果。那我们就必须要将文本进行处理，将各种复杂的标签给剔除，还原帖子的原来面貌。可以使用一个方法或者类将这个处理文本的实现，不过为了更好的代码重用和架构，还是建议使用一个类。

我们将这个类命名为`Too`（工具类），里面定义一个`replace`方法，替换各种标签。然后在类中定义几个正则表达式，利用`re.sub`方法对文本进行匹配后然后替换。

```python
import re

#处理页面标签类
class Tool:
    #去除img标签,7位长空格
    removeImg = re.compile('<img.*?>| {7}|')
    #删除超链接标签
    removeAddr = re.compile('<a.*?>|</a>')
    #把换行的标签换为\n
    replaceLine = re.compile('<tr>|<div>|</div>|</p>')
    #将表格制表<td>替换为\t
    replaceTD= re.compile('<td>')
    #把段落开头换为\n加空两格
    replacePara = re.compile('<p.*?>')
    #将换行符或双换行符替换为\n
    replaceBR = re.compile('<br><br>|<br>')
    #将其余标签剔除
    removeExtraTag = re.compile('<.*?>')
    def replace(self,x):
        x = re.sub(self.removeImg,"",x)
        x = re.sub(self.removeAddr,"",x)
        x = re.sub(self.replaceLine,"\n",x)
        x = re.sub(self.replaceTD,"\t",x)
        x = re.sub(self.replacePara,"\n    ",x)
        x = re.sub(self.replaceBR,"\n",x)
        x = re.sub(self.removeExtraTag,"",x)
        #strip()将前后多余内容删除
        return x.strip()
```

在使用时，我们只需要初始化一下这个类，然后调用replace方法即可。

现在整体代码是如下这样子的，现在我的代码是写到这样子的:

```python
#-*-coding:utf8-*-
#created by 10412

import urllib
import urllib2
import re


# 处理页面标签类
class Tool:
    # 去除img标签,7位长空格
    removeImg = re.compile('<img.*?>| {7}|')
    # 删除超链接标签
    removeAddr = re.compile('<a.*?>|</a>')
    # 把换行的标签换为\n
    replaceLine = re.compile('<tr>|<div>|</div>|</p>')
    # 将表格制表<td>替换为\t
    replaceTD = re.compile('<td>')
    # 把段落开头换为\n加空两格
    replacePara = re.compile('<p.*?>')
    # 将换行符或双换行符替换为\n
    replaceBR = re.compile('<br><br>|<br>')
    # 将其余标签剔除
    removeExtraTag = re.compile('<.*?>')

    def replace(self, x):
        x = re.sub(self.removeImg, "", x)
        x = re.sub(self.removeAddr, "", x)
        x = re.sub(self.replaceLine, "\n", x)
        x = re.sub(self.replaceTD, "\t", x)
        x = re.sub(self.replacePara, "\n    ", x)
        x = re.sub(self.replaceBR, "\n", x)
        x = re.sub(self.removeExtraTag, "", x)
        # strip()将前后多余内容删除
        return x.strip()


# 百度贴吧爬虫类
class BDTB:
    # 初始化，传入基地址，是否只看楼主的参数
    def __init__(self, baseUrl, seeLZ):
        self.baseURL = baseUrl
        self.seeLZ = '?see_lz=' + str(seeLZ)
        self.tool = Tool()

    # 传入页码，获取该页帖子的代码
    def getPage(self, pageNum):
        try:
            url = self.baseURL + self.seeLZ + '&pn=' + str(pageNum)
            request = urllib2.Request(url)
            response = urllib2.urlopen(request)
            return response.read().decode('utf-8')

        except urllib2.URLError, e:
```

```
            if hasattr(e, "reason"):
                print u"连接百度贴吧失败,错误原因", e.reason
                return None

    # 获取帖子标题
    def getTitle(self):
        page = self.getPage(1)
        pattern = re.compile('<h1 class="core_title_txt.*?>(.*?)</h1>', re.S)
        result = re.search(pattern, page)
        if result:
            # print result.group(1)  #测试输出
            return result.group(1).strip()
        else:
            return None

    # 获取帖子一共有多少页
    def getPageNum(self):
        page = self.getPage(1)
        pattern = re.compile('<li class="l_reply_num.*?</span>.*?<span.*?>(.*?)</span>',
re.S)
        result = re.search(pattern, page)
        if result:
            # print result.group(1)  #测试输出
            return result.group(1).strip()
        else:
            return None

    # 获取每一层楼的内容,传入页面内容
    def getContent(self, page):
        pattern = re.compile('<div id="post_content_.*?>(.*?)</div>', re.S)
        items = re.findall(pattern, page)
        # for item in items:
        #  print item
        print self.tool.replace(items[1])


baseURL = 'http://tieba.baidu.com/p/3138733512'
bdtb = BDTB(baseURL, 1)
bdtb.getContent(bdtb.getPage(1))
```

运行截图如下：

```
50 惊喜新人王 迈卡威
上赛季数据
篮板 6.2  助攻 6.3  抢断 1.9 盖帽  0.6 失误 3.5 犯规  3  得分 16.7

新赛季第50位，我给上赛季的新人王迈卡威。 上赛季迈卡威在彻底重建的76人中迅速掌握了球队，一开始就三双搞定了热火赢得了万千眼球。后来也屡屡有经验的表现，新秀赛季就拿过三双的球员不多，
作为上赛季弱队的老大，迈卡威刷出了不错的数据，但我们静下心来看一看他，还是发现他有很多问题。首先，投篮偏弱刚刚40%的命中率和惨淡的26%的三分命中率肯定是不合格的！加之身体瘦弱，个字高
说完缺点，来说说优点，作为后卫篮板球非常突出，高大的身形能较好的影响对方的出手，也能发现己方的空位球员。突破虽然速度一般，但节奏感不错，大局观也在平均水准之上。提醒瘦而高大，不会投
就球队地位而言，迈卡威现在是绝对的老大，球你想怎么玩就怎么玩，数据你想怎么刷就怎么刷！去年的潜力新人诺尔是蓝领，其他人都可以清退，恩比德还受伤不能打，76人队的战绩怎么样，就看你了！

Process finished with exit code 0
```

## 4）替换楼层

至于这个问题，我感觉直接提取楼层没什么必要呀，因为只看楼主的话，有些楼层的编号是间隔的，所以我们得到的楼层序号是不连续的，这样我们保存下来也没什么用。

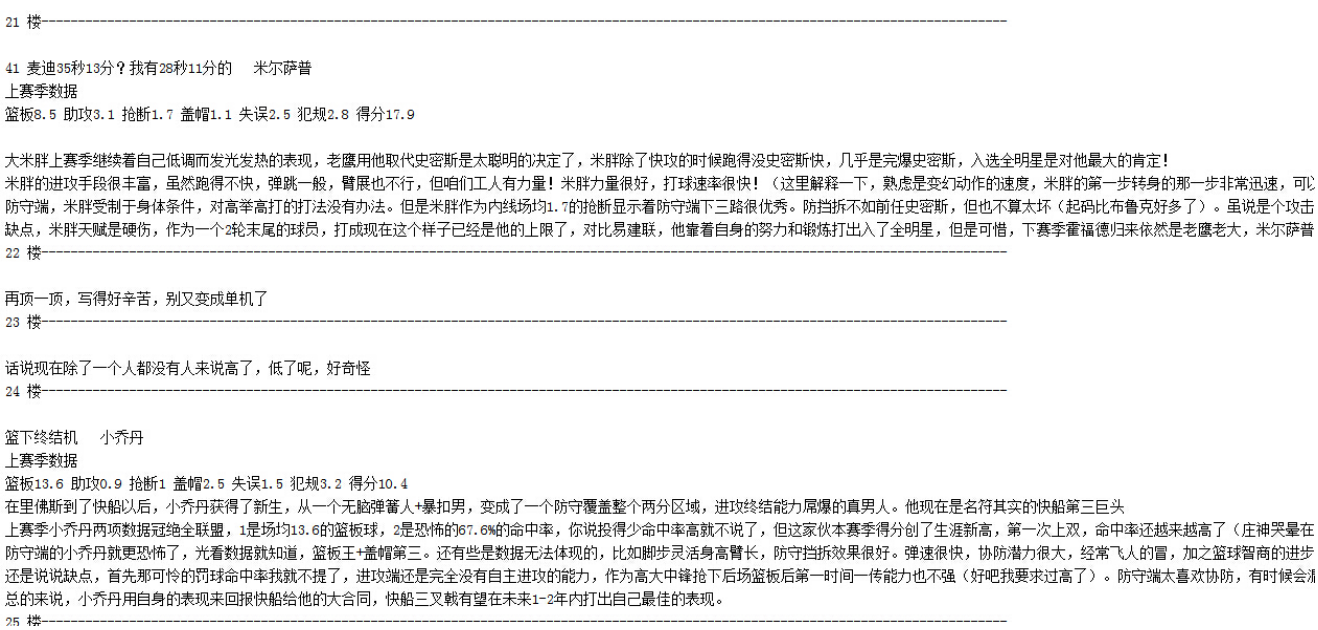
所以可以尝试下面的方法：

> 1.每打印输出一段楼层，写入一行横线来间隔，或者换行符也好。
>
> 2.试着重新编一个楼层，按照顺序，设置一个变量，每打印出一个结果变量加一，打印出这个变量当做楼层。

将`getContent`方法修改如下：

```
#获取每一层楼的内容,传入页面内容
def getContent(self,page):
    pattern = re.compile('<div id="post_content_.*?>(.*?)</div>',re.S)
    items = re.findall(pattern,page)
    floor = 1
    for item in items:
        print floor,u"楼----------------------------------------------------------------------
----------------------------------------------------------------\n"
        print self.tool.replace(item)
        floor += 1
```

运行结果截图如下：




**4.** 写入文件

代码：

```
file = open(“tb.txt”,”w”)
file.writelines(obj)
```

**5.** 完善代码

```python
#-*-coding:utf8-*-
#created by 10412


import urllib
import urllib2
import re

#处理页面标签类
class Tool:
    #去除img标签,7位长空格
    removeImg = re.compile('<img.*?>| {7}|')
    #删除超链接标签
    removeAddr = re.compile('<a.*?>|</a>')
    #把换行的标签换为\n
    replaceLine = re.compile('<tr>|<div>|</div>|</p>')
    #将表格制表<td>替换为\t
    replaceTD= re.compile('<td>')
    #把段落开头换为\n加空两格
    replacePara = re.compile('<p.*?>')
    #将换行符或双换行符替换为\n
    replaceBR = re.compile('<br><br>|<br>')
    #将其余标签剔除
    removeExtraTag = re.compile('<.*?>')
    def replace(self,x):
        x = re.sub(self.removeImg,"",x)
        x = re.sub(self.removeAddr,"",x)
        x = re.sub(self.replaceLine,"\n",x)
        x = re.sub(self.replaceTD,"\t",x)
        x = re.sub(self.replacePara,"\n    ",x)
        x = re.sub(self.replaceBR,"\n",x)
        x = re.sub(self.removeExtraTag,"",x)
        #strip()将前后多余内容删除
        return x.strip()


#百度贴吧爬虫类
class BDTB:

    #初始化，传入基地址，是否只看楼主的参数
    def __init__(self,baseUrl,seeLZ,floorTag):
        #base链接地址
        self.baseURL = baseUrl
        #是否只看楼主
        self.seeLZ = '?see_lz='+str(seeLZ)
        #HTML标签剔除工具类对象
        self.tool = Tool()
        #全局file变量，文件写入操作对象
        self.file = None
        #楼层标号，初始为1
        self.floor = 1
        #默认的标题，如果没有成功获取到标题的话则会用这个标题
        self.defaultTitle = u"百度贴吧"
```

```
        #是否写入楼分隔符的标记
        self.floorTag = floorTag

    #传入页码，获取该页帖子的代码
    def getPage(self,pageNum):
        try:
            #构建URL
            url = self.baseURL+ self.seeLZ + '&pn=' + str(pageNum)
            request = urllib2.Request(url)
            response = urllib2.urlopen(request)
            #返回UTF-8格式编码内容
            return response.read().decode('utf-8')
        #无法连接，报错
        except urllib2.URLError, e:
            if hasattr(e,"reason"):
                print u"连接百度贴吧失败,错误原因",e.reason
                return None

    #获取帖子标题
    def getTitle(self,page):
        #得到标题的正则表达式
        pattern = re.compile('<h1 class="core_title_txt.*?>(.*?)</h1>',re.S)
        result = re.search(pattern,page)
        if result:
            #如果存在，则返回标题
            return result.group(1).strip()
        else:
            return None

    #获取帖子一共有多少页
    def getPageNum(self,page):
        #获取帖子页数的正则表达式
        pattern = re.compile('<li class="l_reply_num.*?</span>.*?<span.*?>(.*?)
</span>',re.S)
        result = re.search(pattern,page)
        if result:
            return result.group(1).strip()
        else:
            return None

    #获取每一层楼的内容,传入页面内容
    def getContent(self,page):
        #匹配所有楼层的内容
        pattern = re.compile('<div id="post_content_.*?>(.*?)</div>',re.S)
        items = re.findall(pattern,page)
        contents = []
        for item in items:
            #将文本进行去除标签处理，同时在前后加入换行符
            content = "\n"+self.tool.replace(item)+"\n"
            contents.append(content.encode('utf-8'))
        return contents


    def setFileTitle(self,title):
```

```
        #如果标题不是为None，即成功获取到标题
        if title is not None:
            self.file = open(title + ".txt","w+")
        else:
            self.file = open(self.defaultTitle + ".txt","w+")

    def writeData(self,contents):
        #向文件写入每一楼的信息
        for item in contents:
            if self.floorTag == '1':
                #楼之间的分隔符
                floorLine = "\n" + str(self.floor) + u"-----------------------------------
-------------------------------------------------------\n"
                self.file.write(floorLine)
            self.file.write(item)
            self.floor += 1

    def start(self):
        indexPage = self.getPage(1)
        pageNum = self.getPageNum(indexPage)
        title = self.getTitle(indexPage)
        self.setFileTitle(title)
        if pageNum == None:
            print "URL已失效，请重试"
            return
        try:
            print "该帖子共有" + str(pageNum) + "页"
            for i in range(1,int(pageNum)+1):
                print "正在写入第" + str(i) + "页数据"
                page = self.getPage(i)
                contents = self.getContent(page)
                self.writeData(contents)
        #出现写入异常
        except IOError,e:
            print "写入异常，原因" + e.message
        finally:
            print "写入任务完成"



print u"请输入帖子代号"
baseURL = 'http://tieba.baidu.com/p/' + str(raw_input(u'http://tieba.baidu.com/p/'))
seeLZ = raw_input("是否只获取楼主发言，是输入1，否输入0\n")
floorTag = raw_input("是否写入楼层信息，是输入1，否输入0\n")
bdtb = BDTB(baseURL,seeLZ,floorTag)
bdtb.start()
```

运行后截图如下：

请输入帖子代号
http://tieba.baidu.com/p/3513281888
是否只获取楼主发言，是输入1，否输入0
1
是否写入楼层信息，是输入1，否输入0
1
该帖子共有7页
正在写入第1页数据
正在写入第2页数据
正在写入第3页数据
正在写入第4页数据
正在写入第5页数据
正在写入第6页数据
正在写入第7页数据
写入任务完成