第十章　网络爬虫

第十章 网络爬虫

网络爬虫又称为网络蜘蛛，是一种高效的信息收集工具，能够对海量的信息进行自动抓取和筛选。网络爬虫通过requests库和beautifulsoup4库抓取互联网站上的信息并形成一个本地的备份，再借助其他的Python模块，将数据信息进行提取和可视化，方便用户进行分析。

本章将用三节内容来介绍网络爬虫的相关知识：

10.1 程序包requests
10.2 程序包beautifulsoup4
10.3 网络爬虫实例

10.1 程序包requests

程序包requests库是一个http请求库，该程序包可以模拟用户向网站服务器发出访问请求，得到服务器响应之后，通过服务器返回的requests对象“爬取”网页信息。程序包requests属于Python语言的外部库，需要用户自行下载。在PyCharm软件中，单击菜单“文件 | 设置”，找到“项目: ZYPrj03”下的“Python解释器”，这里的“ZYPrj03”为本书使用的项目名（对于不同的用户项目名会不相同），然后，单击左上部的“+”号弹出“可用软件包”窗口，在其中输入“requests”，然后，下载并安装该软件包。

如果使用Visual Studio集成开发环境，则需要在控制台执行命令“pip install requests”安装程序包requests。

10.1 程序包requests

通常情况下，网站服务器使用的都是HTTP或者HTTPS协议，这两种协议的请求方式均为GET方式和POST方式。在爬取网页信息之前，需要先了解该网站访问的请求方式，之后，才能使用网络爬虫。在下表中列举了程序包requests中的常用方法。

序号	方法名	描述
1	get	向指定的网址发送 GET 方式请求
2	post	向指定的网址发送 POST 方式请求
3	delete	向指定的网址发送 DELETE 方式请求
4	head	向指定的网址发送 HEAD 方式请求
5	put	向指定的网址发送 PUT 方式请求
6	request	向指定的网址发送指定的请求方法

10.1.1 get方法

爬取一个网页信息的步骤为：首先，向该网页发送HTTP请求，网页响应后会返回一个response对象，网页的响应信息就储存在该对象中；然后，调用response对象中的属性，将其中的响应信息输出。下表列举了response对象的响应信息属性。

序号	对象参数	含义
1	content	响应内容（二进制形式）
2	headers	以字典形式返回响应信息的头部
3	status_code	返回响应的状态码，用于检验请求是否成功得到响应（200 表示成功，404 表示失败）
4	reason	描述响应的状态并返回（OK 表示成功，Not Found 表示找不到该网页）
5	text	响应内容（字符串形式）
6	apparent_encoding	响应内容编码方式
7	encoding	设置接收网站编码

10.1.1 get方法

下面的实例将对上表中常用的参数举例说明：

```
1    import requests as req
2    if __name__=='__main__':
3        url1 = 'https://5b0988e595225.cdn
4                '20180720/f49a02305490
5        url2 = 'https://fy.tingclass.n
6        re1=req.get(url1)
7        re2=req.get(url2)
8        photo=open('moon.jpg','wb
9        photo.write(re1.content)
10       photo.close()
11       print(re1.headers)
12       print(re1.status_code)
13       print(re1.reason)
14       print(re2.apparent_encoding)
15       print(re2.text)
```

第1行装载程序包requests，并命名其别名为req。
第3~4行将网址赋给url1，该网页为一个月球的照片。
第5行将网址赋给url2，该网址为“听力课堂”关于月球moon
址url1和url2的信息，这两行
get方法，向指定的网址url1
在得到网页响应信息之后，
回的两个response对象分别
oon.jpg。
re1的响应内容。
行程序后，工程所在目录下将
生成一个moon.jpg文件。
第11~15行输出re1对象（响应信息）的头部、状态码和状态，和re2对象（响应信息）的编码和文本信息。

10.1.2 post方法

调用post方法可以向指定的网址发送POST请求，且该请求将包含的数据一起发送至网址，适用于向指定的网页发送特定的数据内容，例如上传图片文件等。post方法有三种常用的携带数据方式：

表单方式（默认方式）

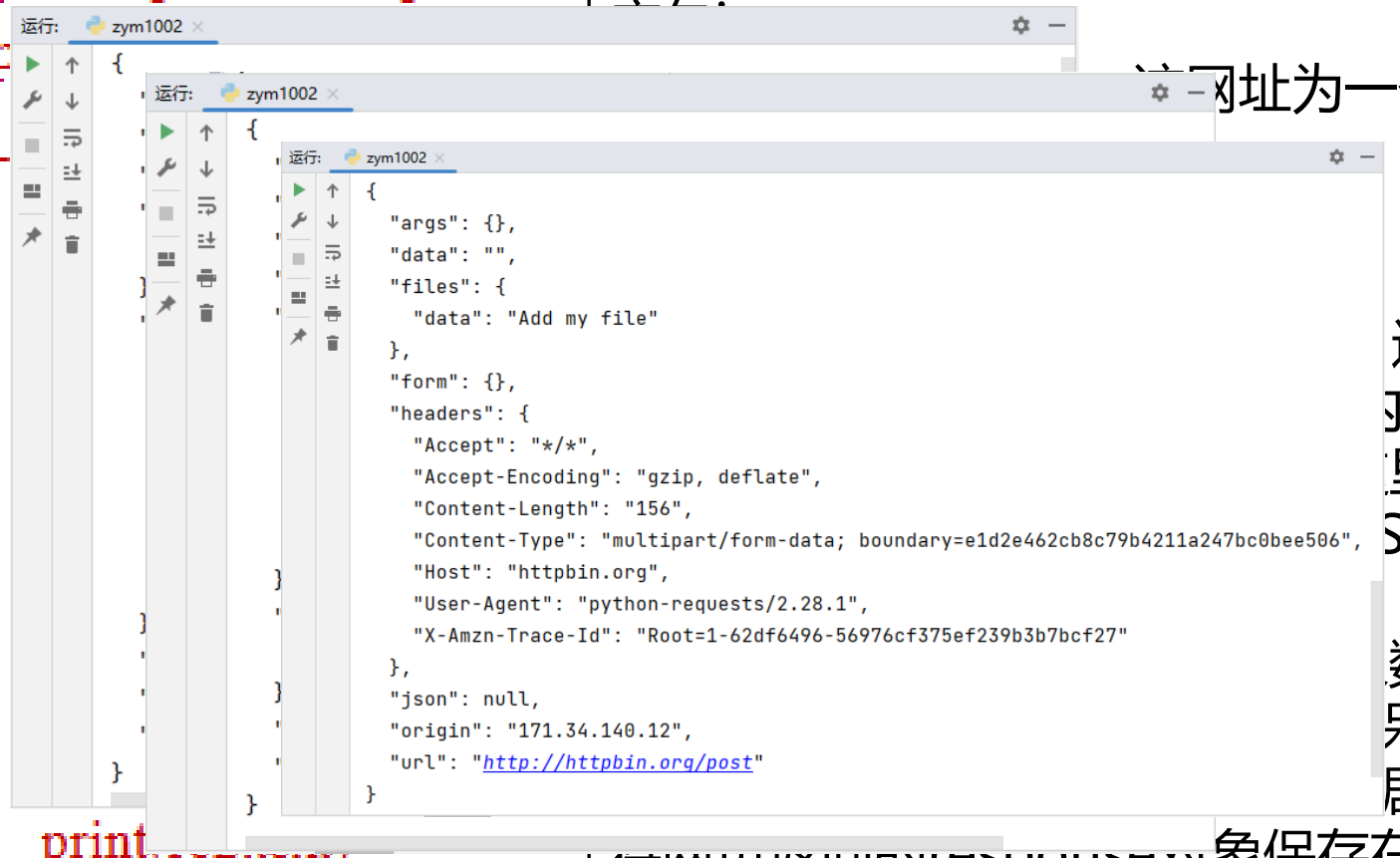
json方式

文件方式

10.1.2 post方法

下面的实例将介绍上述三种方式的具体用法:

```
1   import requests as req
2   imp
3   if _
4
5
6
7
8
9
10
11
12      print(re3.text)
13      print(re3.text)
```



第1行装载requests程序包，并命名为别名req；第2行装载程

网址为一个测试HTTP请求与响应服

这里的文件zydat.txt为当前
内容为“Add my file”。
里是调用post方法携带着指
ST请求，将网页返回的

数据向网页发送POST请求，
保存在对象re2中。
据信息向网页发送POST请求，
将网页返回的response对象保存在对象re3。
第11~13行输出re1对象、re2对象和re3对象中的文本数据。

10.1.3 网页链接异常情况

调用程序包requests中的请求方法链接网页时，会出现多种多样的异常情况，弄清每一种异常情况的提示语句，才能解决和避免异常。下表中列举了几种常见的异常情况。

序号	异常情况	报错类型
1	连接或者读取服务器，在指定时间内没有得到响应	ConnectTimeout
2	指定网址对应的服务器不存在	ConnectionError
3	得到的响应状态码是 404	HTTPError
4	网络出现异常导致连接失败	ConnectionError
5	代理服务器的响应超时	HTTPConnectionPool

为了避免程序因为得不到服务器响应而陷入无限等待，可以对参数timeout进行赋值，设置程序等待响应的时间。若在指定的时间内除了基本的应答字节，程序没有得到服务器反馈的字节数据，程序将会结束等待，自动抛出一个异常。

10.2 程序包beautifulsoup4

调用程序包requests连接网页，将其HTML页面转换为字符串存储在文档中之后，需要对HTML页面的内容进行处理。程序包beautifulsoup4用于解析Web页面的HTML或者XML，将HTML文档转换为一个树形结构的文档，并将解析结果打包封装，配置了相应的方法对其进行访问。程序包beautifulsoup4还具有一个强大的功能，即可以根据HTML或者XML的语法来创建一个文档树。程序包beautifulsoup4是外部软件包，在使用前需要进行安装，安装方法类似于第10.1节介绍了程序包requests的安装方法。

如果使用Visual Studio集成开发环境，需要在“命令提示符”窗口下使用命令“pip install bs4”安装程序包beautifulsoup4。

对网页中所需的信息进行定位并爬取，需要了解HTML/XML页面的格式结构。程序包beautifulsoup4解析的HTML/XML页面的格式是一个树形结构，其中包含了几种结点对象，常用的四种对象有：Tag、BeautifulSoup、NavigableString和Comment。这里重点介绍Tag对象和BeautifulSoup对象。

10.2.1 Tag对象和BeautifulSoup对象

Tag对象是程序包beautifulsoup4中常用的对象，Tag对象中包含的标签和HTML中的标签相同。下表列举了Tag对象中常用的标签。

序号	标签	描述
1	head	页面的头部（head）信息，其中包含了标题信息
2	title	页面的标题
3	body	页面的主体，其中包含了字符串信息等
4	p	页面 body 标签内的字符串信息标签

在HTML页面中标签是成对使用的，其格式为：“<标签>内容部分</标签>”。可以随意打开一个网页，在页面的空白处单击鼠标右键，在其弹出菜单中选择“查看网页源代码”，或者按下“F12”键，即可查看到该网页的HTML代码。

10.2.1 Tag对象和BeautifulSoup对象

调用beautifulsoup4库中的BeautifulSoup()方法可创建一个BeautifulSoup对象，该对象中包含了解析树的全部信息，实质上也属于一种Tag对象，但是BeautifulSoup对象比Tag对象的性能更强大，不仅可以对文档树进行搜索操作，还可以遍历整个文档树。了解了HTML的语法格式之后，仿照HTML页面的语法格式，可以自定义一个简单的BeautifulSoup对象。

10.2.1 Tag对象和BeautifulSoup对象

自定义的BeautifulSoup对象的HTML格式与网页源代码一致，说明BeautifulSoup对象创建成功。调用主体内容对象（这里为bs）的text属性可获取自定义的页面内容。

在HTML格式中，HTML的每一种标签都有四个基本属性。下表介绍了这四个属性的类型及其含义。

序号	属性	描述
1	name	标签的名字（字符串类型）
2	attrs	标签的所有属性（字典类型）
3	string	标签的文本内容（字符串类型）
4	contents	标签中的所有子标签

10.2.1 Tag对象和BeautifulSoup对象

下面的实例介绍了上表中的标签属性及其用法:

```
1   import bs4
2   if __name__=='__main__'
3       ht = '<html><head><t
4           '<body><div cla
5           '<p class="story
6           '<p class="story
7           '<p class="story
8       soup=bs4.BeautifulSo
9       p1=soup.p
10      print(p1.name)
11      print(p1.attrs)
12      print(p1.string)
13      print(p1.contents)
14      for p2 in soup.find_al
15          print(p2.name)
16          print(p2.attrs)
17          print(p2.string)
18          print(p2.contents)
```

运行: zym1004

```
E:\ZYPythonPrj\ZYPrj03\venv\Scripts\python.exe E
p
{'class': ['story'], 'id': 'name'}
 Zhang Fei
[' Zhang Fei ']
p
{'class': ['story'], 'id': 'name'}
 Zhang Fei
[' Zhang Fei ']
p
{'class': ['story'], 'id': 'sex'}
 M
[' M ']
p
{'class': ['story'], 'id': 'score'}
 95.3
[' 95.3 ']
```

nt。
eautifulSoup对象，
式模拟创建一个HTML页面，
行设置，将创建好的

，将BeautifulSoup对象中

<p>标签的名字，再调用

p>标签的所有属性，再调

<p>标签的文本内容，再调

到<p>标签的所有子标签，

p中的全部<p>标签，依次
子标签信息。