day03

- 复习
- 爬虫库(urllib系类, requests)

复习

爬虫基础知识

> 所谓网页抓取，就是把URL地址中指定的网络资源从网络流中读取出来，保存到本地。 在Python中有很多库可以用来抓取网页，urllib系列(urllib,urllib2,urllib3, reqeusts[工作中使用])。

```
# 爬虫基本流程
1. 发起请求
2. 获取响应
3. 解析响应结果
4. 存储

# 请求Request
1. 请求方式/方法: 常用:get/post        还有更多:delete删除请求, put修改请求
2. 请求url地址组成
   http://IP地址:端口/地址
   协议     IP地址:端口/页面地址          http协议:默认端口80
                                        https协议:默认端口443
3. 请求头(headers):
   浏览器发送请求时, 默认添加一些头信息:比如 user-agent标识,cookie,host IP地址等...
   F12 可查看!
   Fiddler抓包工具
4. Respons响应
   状态码. 响应头,响应体(用户看到的内容)
```

爬虫框架

- 可以爬去网络中的任何信息(文本.多媒体(图片,视频.音频))
- 爬取数据有对应的库! urllib系列,[早期,网上案例多] requests [工作常用]
- 解析数据

```
1. 直接处理
2. Json解析
3. 正则表达式处理
4. BeautifulSoup4解析处理
5. PyQuery解析处理
6. XPath解析处理
```

- python中有很多爬虫库,python默认自带的urllib

```
# urllib 版本比较多，相互不兼容！  最基本爬虫库！    更加方便的：urllib为基础,reqeusts

python3.x：自带urllib（是urllib+urlib2合并后的）
           可扩展:urllib3
pip install urllib3
```

> 在python2.x里面有urllib和urllib2;在python3.x里面就把urllib和urllib2合成一个urllib;urllib3是在python3.x了里面新增的第三方扩展.

urllib

> python3中的urllib = (urllib + urllib2)
>
> urllib3 需要单独安装

- 各版本区别参考扩展阅读

https://zhidao.baidu.com/question/1115891882574120939.html

https://blog.csdn.net/jiduochou963/article/details/87564467

urllib库的核心模块4个

- **urllib.request**---请求模块，用于发起网络请求
  - 发送请求
  - 模拟浏览器,添加头信息 user-agent
  - 设置cookie
  - 设置代理等...

  通过rulllib.request.ProxyHandler()可以设置代理,网站它会检测某一段时间某个IP 的访问次数，如果访问次数过多，它会禁止你的访问,所以这个时候需要通过设置代理来爬取数据

```
当频繁访问某个网站时， 会被当成恶意攻击， IP地址会被封掉!!!!
可以使用代理IP访问该网站!
代理IP可以花钱购买
常用代理服务器
西刺免费代理IP：http://www.xicidaili.com/
快代理：http://www.kuaidaili.com/
代理云：http://www.dailiyun.com/

等...
```

- **urllib.parse**---解析模块，用于解析URL
- **urllib.error**---异常处理模块，用于处理request引起的异常
- **urllib.robotparser robots.txt**---用于解析爬虫协议robots.txt文件

请求模块方法

from urllib import request

方法	说明	参数说明
request.urlopen(url,timeout=10)	发送get请求	地址,数据,超时时间s
request.urlopen(url,data=data)	发送post请求	data类型必须位bytes
req = request.Request(url=url, headers=headers, method='GET')	使用Request方法发送请求,包含请求头	headers是请求头,method是方法
request.ProxyHandler({代理Ip列表})	创建代理对象	参数是代理IP列表
opener=request.build_opener(代理对象)	创建代理打开器对象	参数是代理对象
opener.open(url)	使用代理代开url	参数是url

from urllib import parse

- **urlencode转码和参数拼接**

```
# 转码
dict = {
    'name': 'zhaofan'
}
# post传参数.参数字符串集需要转化为bytes
data = bytes(parse.urlencode(dict), encoding='utf8')
req = request.Request(url=url, data=data, headers=headers, method='POST')
response = request.urlopen(req)
print(response.read().decode('utf-8'))
```

urlencode

这个方法可以将字典转换为url参数，例子如下

```
from urllib.parse import urlencode

params = {
    "name":"zhaofan",
    "age":23,
}
base_url = "http://www.baidu.com?"

url = base_url+urlencode(params)
print(url)
```

拼接为?值1=值1&值2=值2....

结果为：

```
urllib的urlencode
    /Library/Frameworks/Python.framework/Versions
    http://www.baidu.com?name=zhaofan&age=23
```

- **各种请求头列表: 手机的PC的**

服务器依靠user-agent区分是手机端或PC端

`https://blog.csdn.net/u012175089/article/details/61199238`

```
# 不同浏览器user-agent不同!
from urllib import request, parse
import  random
url = 'http://www.baidu.com'
user_agent = [
    "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50
(KHTML, like Gecko) Version/5.1 Safari/534.50",
```

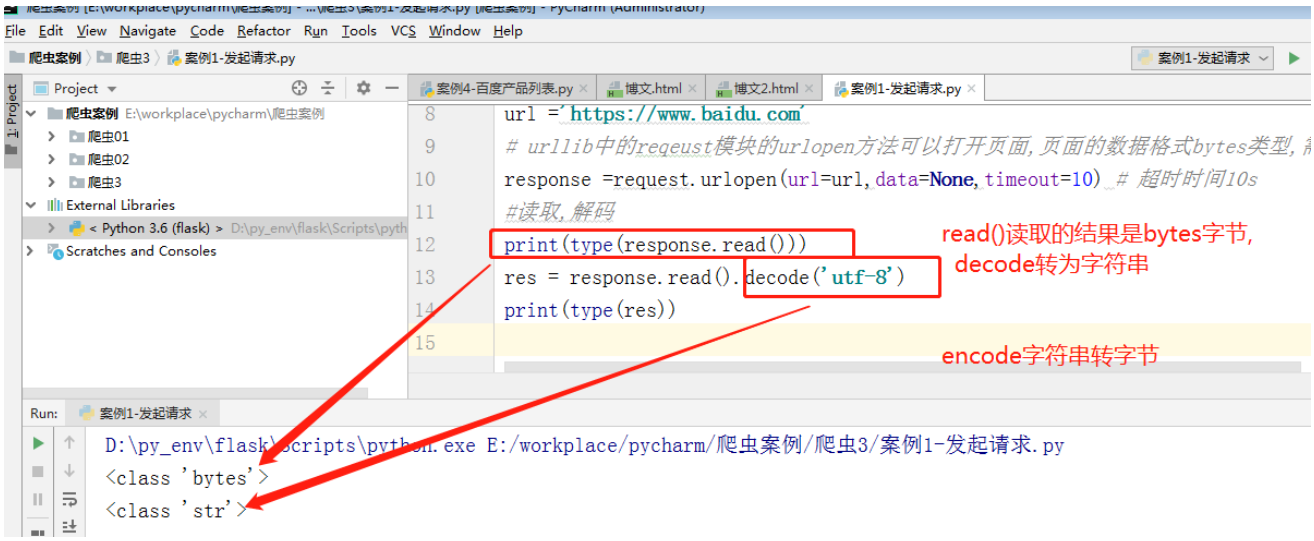
```
    "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML, like
Gecko) Version/5.1 Safari/534.50",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR
2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729; InfoPath.3; rv:11.0) like Gecko",
    "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)",
    "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)",
    "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1) Gecko/20100101
Firefox/4.0.1",
    "Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1",
    "Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.8.131
Version/11.11",
    "Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131 Version/11.11",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML, like
Gecko) Chrome/17.0.963.56 Safari/535.11",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The World)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SE 2.X MetaSr 1.0;
SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr 1.0)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Avant Browser)",
    "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)",
    "Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_3_3 like Mac OS X; en-us)
AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5",
    "Mozilla/5.0 (iPod; U; CPU iPhone OS 4_3_3 like Mac OS X; en-us)
AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5",
    "Mozilla/5.0 (iPad; U; CPU OS 4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9
(KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5",
    "Mozilla/5.0 (Linux; U; Android 2.3.7; en-us; Nexus One Build/FRF91)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1",
    "MQQBrowser/26 Mozilla/5.0 (Linux; U; Android 2.3.7; zh-cn; MB200 Build/GRJ22;
CyanogenMod-7) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1",
    "Opera/9.80 (Android 2.3.4; Linux; Opera Mobi/build-1107180945; U; en-GB)
Presto/2.8.149 Version/11.10",
    "Mozilla/5.0 (Linux; U; Android 3.0; en-us; Xoom Build/HRI39) AppleWebKit/534.13
(KHTML, like Gecko) Version/4.0 Safari/534.13",
    "Mozilla/5.0 (BlackBerry; U; BlackBerry 9800; en) AppleWebKit/534.1+ (KHTML, like
Gecko) Version/6.0.0.337 Mobile Safari/534.1+",
    "Mozilla/5.0 (hp-tablet; Linux; hpwOS/3.0.0; U; en-US) AppleWebKit/534.6 (KHTML,
like Gecko) wOSBrowser/233.70 Safari/534.6 TouchPad/1.0",
    "Mozilla/5.0 (SymbianOS/9.4; Series60/5.0 NokiaN97-1/20.0.019; Profile/MIDP-2.1
Configuration/CLDC-1.1) AppleWebKit/525 (KHTML, like Gecko) BrowserNG/7.1.18124",
    "Mozilla/5.0 (compatible; MSIE 9.0; Windows Phone OS 7.5; Trident/5.0;
IEMobile/9.0; HTC; Titan)",
    "UCWEB7.0.2.37/28/999",
    "NOKIA5700/ UCWEB7.0.2.37/28/999",
    "Openwave/ UCWEB7.0.2.37/28/999",
    "Mozilla/4.0 (compatible; MSIE 6.0; ) Opera/UCWEB7.0.2.37/28/999",
    # iPhone 6：
```

```
    "Mozilla/6.0 (iPhone; CPU iPhone OS 8_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/8.0 Mobile/10A5376e Safari/8536.25",
]
# 从列表中随机获取一个请求头
headers = {'User-Agent': random.choice(user_agent)}
data = bytes(parse.urlencode(dict), encoding='utf8')
req = request.Request(url=url, headers=headers, method='GET')
response = request.urlopen(req)
print(response.read().decode('utf-8'))
```

响应对象方法

```
# get请求
response =request.urlopen(url=url,data=None,timeout=10) # 超时时间10s
# post请求
data = bytes(parse.urlencode({'uname':'admin','pwd':'123'}),encoding='utf8')
response = request.urlopen('http://httpbin.org/post',data=data)
```

方法	说明	参数
response.read().decode('urf-8')	读取并解码	read方法读取的数据类型是bytes,需要decode解码为字符串



发送post请求

```
'''
urllib  不如 request方便
request 需要依赖urllib

urllib依靠参数data来区分get和post请求
有data参数,表示post, 参数书格式必须为bytes
'''
from urllib import request # 请求模块
from urllib import  parse  # 解析模块

data = bytes(parse.urlencode({'uname':'admin','pwd':'123'}),encoding='utf8')
response = request.urlopen('http://httpbin.org/post',data=data)
print(response.read().decode('utf8'))
```

代理

通过rulllib.request.ProxyHandler()可以设置代理,网站它会检测某一段时间某个IP 的访问次数，如果访问次数过多，它会禁止你的访问,所以这个时候需要通过设置代理来爬取数据

```
'''
urllib.reqeust模块中代理方法
Proxy: 代理
request.ProxyHandler(代理IP列表)

'''
```

```python
from urllib import  request
url='http://httpbin.org/get'

# 创建代理
proxy_handler = request.ProxyHandler({
     'http':'113.128.28.52:9999',
     'http':'113.128.28.52:9999',
})
# 创建代理打开器
opener = request.build_opener(proxy_handler)
# 打开
response = opener.open(url)
print(response.read())
```

urllib3

- 是urllib的升级,功能更强大! 条例更清晰!
- 安装: `pip install urllib3` 如果安装了requests默认urllib3已经按照了
- 新特性

```
1. 线程安全
2. 连接池
3  文件部分编码上传
4. 支持压缩
5. 处理重复请请求
.....
```

urllib和urllib3 是一个库么???

作业

- 整理urllibe 和urllib3 的方法列表!
- 作业:交文档/课堂中案例