

Python爬虫

NSD python

爬虫

- 网络爬虫（又被称为网页蜘蛛，网络机器人，在FOAF社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。另外一些不常使用的名字还有蚂蚁、自动索引、模拟程序或者蠕虫。



通用爬虫/搜索引擎

- Google
- Baidu
- Yahoo



协议/robots.txt

- Robots协议（也称为爬虫协议、机器人协议等）的全称是“网络爬虫排除标准”（Robots Exclusion Protocol），网站通过Robots协议告诉搜索引擎哪些页面可以抓取，哪些页面不能抓取。



Robots.txt

禁止所有搜索引擎访问网站的任何部分

User-agent: *

Disallow: /

实例分析：淘宝网的 Robots.txt文件

User-agent: Baiduspider

Disallow: /

User-agent: baiduspider

Disallow: /



urllib库

urllib get -1

```
import urllib.request
response = urllib.request.urlopen('http://python.org/')
html = response.read()
print(html)
```



urllib get - 模拟浏览器

```
import urllib.request  
import urllib.parse
```

```
url='http://www.baidu.com'  
header={  
'User-Agent':'Mozilla/5.0 (X11; Fedora; Linux x86_64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110  
Safari/537.36'}
```

```
request=urllib.request.Request(url,headers=header)  
reponse=urllib.request.urlopen(request).read()
```

```
fhandle=open("./1.html","wb")  
fhandle.write(reponse)  
fhandle.close()
```



urllib post

```
import urllib.request
import urllib.parse
```

```
url='http://www.example.com/login'
header={
    'User-Agent':'Mozilla/5.0 (X11; Fedora; Linux x86_64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110
    Safari/537.36'
}
```

```
data={'name':'fengxin','pass':'123'}
postdata=urllib.parse.urlencode(data).encode('utf8') #进行编码
request=urllib.request.Request(url,data=postdata)
reponse=urllib.request.urlopen(request).read()
```

```
fhandle=open("./1.html","wb")
fhandle.write(reponse)
fhandle.close()
```



Urllib cookie

```
import http.cookiejar
import urllib.request
import urllib.parse

cj = http.cookiejar.CookieJar()
opener =
urllib.request.build_opener(urllib.request.HTTPCookieProcessor(
cj))
data = {'username': 'admin', 'password': 'admin'}
r = opener.open("http://localhost:8888/login",
data=urllib.parse.urlencode(data).encode('utf8'))
print(r.read())
r = opener.open('http://localhost:8888/')
print(r.read())
```



Urllib 代理

```
from urllib import request
```

```
proxy = request.ProxyHandler({'http': '81.89.71.166:51890'}) # 设置
proxy
opener = request.build_opener(proxy) # 挂载opener
opener.addheaders = [('User-Agent', 'Mozilla/5.0 (X11; Fedora; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
58.0.3029.110 Safari/537.36')]
request.install_opener(opener)
page = opener.open('http://www.baidu.com').read()
page = page.decode('utf-8')
print(page)
```



练习

- 利用urllib发送get请求到任意网页
- 使用代理访问 <http://2017.ip138.com/ic.asp>



requests库

requests

- 安装

`pip install requests`



基本操作

```
import requests
```

```
r = requests.get('https://github.com/timeline.json')
```

```
r = requests.post("http://httpbin.org/post")
```

```
r = requests.put("http://httpbin.org/put")
```

```
r = requests.delete("http://httpbin.org/delete")
```

```
r = requests.head("http://httpbin.org/get")
```

```
r = requests.options("http://httpbin.org/get")
```



Get传参

```
import requests
```

```
payload = {'key1': 'value1', 'key2': ['value2', 'value3']}  
r = requests.get('http://httpbin.org/post', params=payload)
```



POST传参

```
import requests
```

```
payload = {'key1': 'value1', 'key2': 'value2'}
```

```
r = requests.post("http://httpbin.org/post", data=payload)
```

```
print r.text
```



定制请求头

```
url = 'https://api.github.com/some/endpoint'  
headers = {'user-agent': 'my-app/0.0.1'}  
r = requests.get(url, headers=headers)
```



代理

```
import requests
```

```
proxies = { "http": "http://10.10.1.10:3128", "https": "http://  
10.10.1.10:1080", } requests.get("http://example.org", proxies=proxies)
```



Session

```
s = requests.Session() s.get('http://httpbin.org/cookies/set/  
sessioncookie/123456789')  
r = s.get("http://httpbin.org/cookies")  
print r.text  
# '{"cookies": {"sessioncookie": "123456789"}}'
```



练习

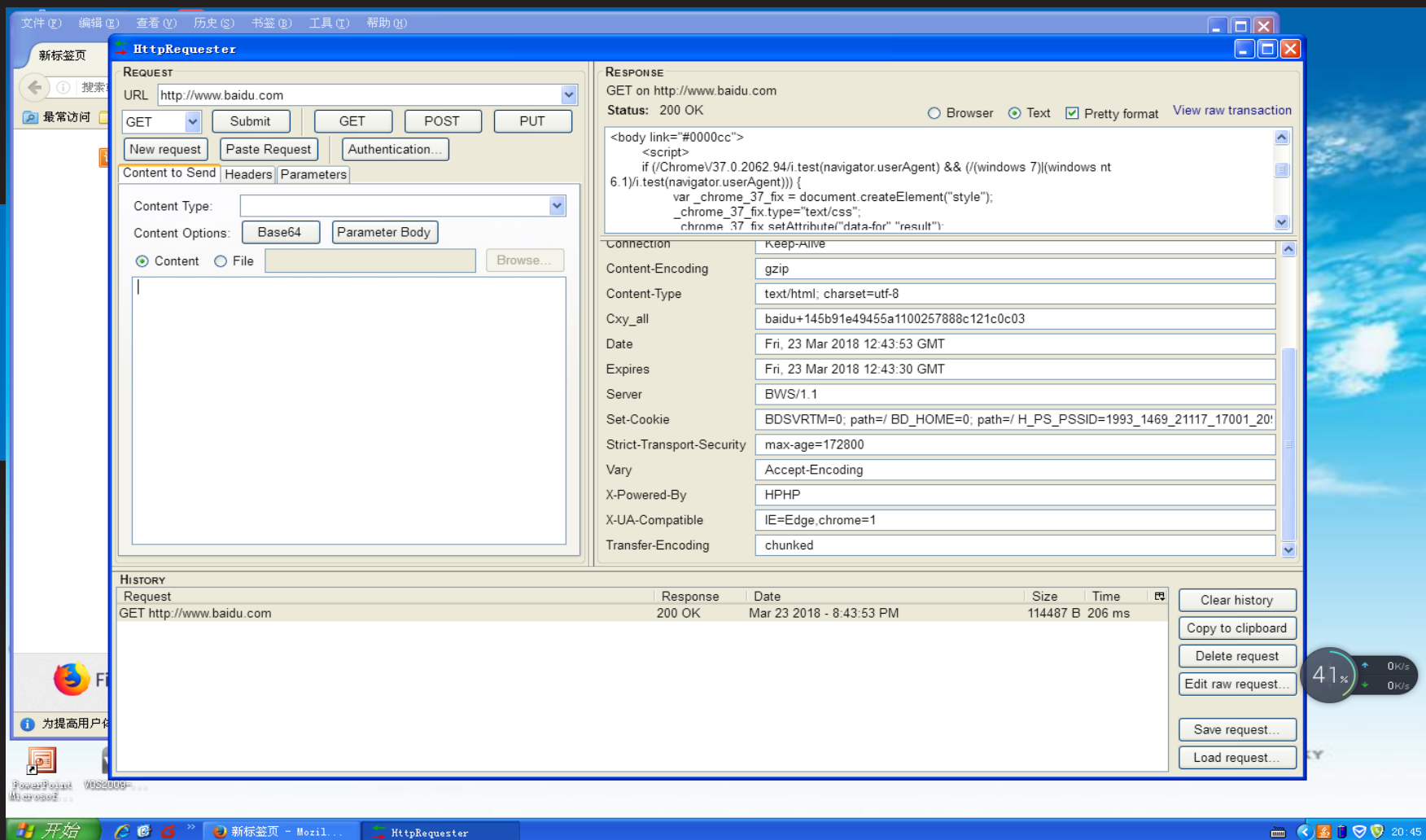
- 使用requests库登陆一个网页



工具



Firefox扩展httprequester



41%
0K/s
0K/s

Firefox扩展httpfox

文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)

登录豆瓣

https://accounts.douban.com/login?alias=hipeace86&redir=https%3A%2F%2Fwww.douban.com%2F%3Fsource=index_nav

豆瓣douban 帐号

登录豆瓣

帐号和密码不匹配

手机验证码登录

帐号 邮箱/手机号/用户名

密码

> 还没有豆瓣帐号? 立即注册

> 点击下载豆瓣移动应用

Start Stop Clear Autoscroll

Started	Time	Sent	Received	Method	Result	Type	URL
00:04:59...	0.076	274	243	GET	200	application/json	http://i.g-fox.cn/notification/0.7/rules_sincefx4.json
00:05:01...	0.028	278	0	GET	(Aborted)	NS_BINDING_ABORTED	https://www.google-analytics.com/analytics.js
00:05:01...	0.168	795	158	POST	302	Redirect (cached)	https://www.douban.com/accounts/login
00:05:01...	0.208	676	(0)	GET	(Cache)	text/html	https://accounts.douban.com/login?alias=hipeace86&redir=https%3A%2F%2Fwww.douban.com%2F%3Fsource=index_nav&error=1013
00:05:01...	0.018	447	388	GET	200	application/javascript	https://img3.doubanio.com/f/accounts/c5268df4c1f0bada95cb3d2b80089a50b494b5ee/js/lib/jquery.min.js
00:05:01...	0.108	450	16350	GET	200	image/png	https://img3.doubanio.com/f/accounts/1b6cc3ca91f78cf47f41eafa91fbc4918ae239c/pics/connect_wechat.png
00:05:01...	0.068	454	3824	GET	200	image/png	https://img3.doubanio.com/f/accounts/e2f1d8c0ede93408b46cbbab4e613fb29ba94e35/pics/connect_sina_weibo.png
00:05:01...	0.134	409	(0)	GET	(Cache)	text/html	https://www.douban.com/accounts/misc/ps?ps=true

Headers Cookies Query String POST Data Content

Request Header	Value	Response Header	Value
(Request-Line)	POST /accounts/login HTTP/1.1		
Host	www.douban.com		
User-Agent	Mozilla/5.0 (Windows NT 5.1; rv:52.0) Gecko/20100101 Firefox/52.0		
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8		
Accept-Language	zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3		
Accept-Encoding	gzip, deflate, br		
Referer	https://www.douban.com/		
Cookie	ll="108288"; bid=7e27gmePj7A; _pk_id.100001.8cb4=09eb2d1298bc76c3.1521809299.1.152180929...		
Content-Type	application/x-www-form-urlencoded		
Content-Length	61		

39% 0K/s 0K/s

为提高用户体验, Firefox 将发送部分功能的使用情况给我们, 用于进一步优化火狐浏览器的易用性, 您可以自由选择是否向我们分享数据。

选择您要分享的数据(C)

开始 登录豆瓣 - Mozil...

20:49



pyquery库

pyquery

- pyquery 可让你用 jQuery 的语法来对 xml 进行操作。这和 jQuery 十分类似。如果利用 lxml，pyquery 对 xml 和 html 的处理将更快。
- 这个库不是（至少还不是）一个可以和 JavaScript交互的代码库，它只是非常像 jQuery API 而已。
- 安装
`pip install pyquery`



初始化（1）直接字符串

```
from pyquery import PyQuery as pq  
doc = pq("<html></html>")
```



初始化 (2) lxml.etree

```
from lxml import etree  
doc = pq(etree.fromstring("<html></html>"))
```



初始化（3）直接URL

```
from pyquery import PyQuery as pq  
doc = pq('http://www.baidu.com')
```



初始化（4）传文件

```
from pyquery import PyQuery as pq  
doc = pq(filename='hello.html')
```



样例

```
<div>
  <ul>
    <li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></li>
    <li class="item-0 active"><a href="link3.html"><span
class="bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
  </ul>
</div>
```

```
from pyquery import PyQuery as pq
doc = pq(filename='hello.html')
print doc.html()
print type(doc)
li = doc('li')
print type(li)
print li.text()
```



遍历

```
from pyquery import PyQuery as pq
doc = pq(filename='hello.html')
lis = doc('li')
for li in lis.items():
    print li.html()

print lis.each(lambda e: e)
```



练习

- 使用pyquery解析html



pyspider



pyspider

- A Powerful Spider(Web Crawler) System in Python.
- Write script in Python
- Powerful WebUI with script editor, task monitor, project manager and result viewer
- MySQL, MongoDB, Redis, SQLite, Elasticsearch; PostgreSQL with SQLAlchemy as database backend
- RabbitMQ, Beanstalk, Redis and Kombu as message queue
- Task priority, retry, periodical, recrawl by age, etc...
- Distributed architecture, Crawl Javascript pages, Python 2&3, etc...



安装&启动

```
pip install pyspider
```

```
pyspider
```

<http://localhost:5000/>

