# urllib模块的使用

Python3.X 版本下将python2.X版本的urllib和urllib2 合并在一起成一个新的urllib，方法并入error、parse、request、response下，连接请求基本在request中通过urllib.request调用，数据处理的一些方式放入parse中。

## 爬取一个网页

get.py

import urllib.request

file=urllib.request.urlopen('http://www.baidu.com')

data=file.read() #读取全部  
print(data)

fhandle=open("./1.html","wb") #将爬取的网页保存在本地

fhandle.write(data)

fhandle.close()

## 模拟器的浏览

**应用场景**：有些网页为了防止别人恶意采集其信息所以进行了一些反爬虫的设置，而我们又想进行爬取。   
**解决方法**：设置一些Headers信息（User-Agent），模拟成浏览器去访问这些网站。

header.py

import urllib.request

import urllib.parse

url='http://www.baidu.com/s?wd='

key='fengxin的博客'

key\_code=urllib.request.quote(key) #因为URL里含中文，需要进行编码

url\_all=url+key\_code

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'

} #头部信息

request=urllib.request.Request(url\_all,headers=header)

reponse=urllib.request.urlopen(request).read()

fh=open("./baidu.html","wb") #写入文件中

fh.write(reponse)

fh.close()

## 代理服务器的设置

**应用场景**：使用同一个IP去爬取同一个网站上的网页，久了之后会被该网站服务器屏蔽。   
**解决方法**：使用代理服务器。 （使用代理服务器去爬取某个网站的内容的时候，在对方的网站上，显示的不是我们真实的IP地址，而是代理服务器的IP地址）

代理服务器IP地址资源：

<http://www.xicidaili.com/nn/1>

agent.py

**import** urllib.request

import random

proxy\_list = [

{"http": "59.39.128.22:8118"},

{"http": "121.205.254.83:8010"}

]

url = "http://www.wave12.com/"

# 随机选择一个代理

proxy = random.choice(proxy\_list)

# 使用选择的代理构建代理处理器对象

httpproxy\_handler = urllib.request.ProxyHandler(proxy)

opener = urllib.request.build\_opener(httpproxy\_handler)

urllib.request.install\_opener(opener)

data=urllib.request.urlopen(url).read().decode('utf8')

print data

## post请求

**import** urllib.request  
**import** time  
  
url=**'http://www.wave12.com/bigfish/user/loginHandle'**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={**'loginName'**:**'test'**,**'passWord'**:**'111111'**}  
postdata=urllib.parse.urlencode(data).encode(**'utf8'**)  
request=urllib.request.Request(url,postdata,headers=header)  
reponse=urllib.request.urlopen(request).read()  
  
fhandle=open(**"./post.html"**,**"wb"**)  
fhandle.write(reponse)  
fhandle.close()

## cookie的使用

**应用场景**：爬取的网页涉及登录信息。访问每一个互联网页面，都是通过HTTP协议进行的，而HTTP协议是一个**无状态协议**，所谓的无状态协议即**无法维持会话之间的状态**。

基本的urlopen()函数不支持验证、cookie或其他HTTP高级功能。要支持这些功能，必须使用build\_opener()函数来创建自己的自定义[Opener](http://www.nowamagic.net/academy/tag/Opener" \t "_blank)对象。

实例cookie.py：

import http.cookiejar, urllib.request  
  
url = 'http://www.wave12.com/bigfish/user/loginHandle'  
  
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'  
}  
  
cookie = http.cookiejar.CookieJar()  
handler = urllib.request.HTTPCookieProcessor(cookie)  
opener = urllib.request.build\_opener(handler)  
*#将opener安装为全局*urllib.request.install\_opener(opener) *# 去掉这句，url2访问会异常*data = {'loginName': 'test', 'passWord': '111111'}  
postdata = urllib.parse.urlencode(data).encode('utf8')  
request = urllib.request.Request(url, postdata, headers=header)  
reponse=urllib.request.urlopen(request).read()  
  
for item in cookie:  
 print(item.name+"="+item.value)  
  
fhandle = open("./data1.html", "wb")  
fhandle.write(reponse)  
fhandle.close()  
  
*# 后续继续访问登陆后的url，会保持cookie，即客户端的cookie信息会自动发送到服务端，以让服务端用session识别当前的用户*url2='http://www.wave12.com/bigfish/myWeb/manage' *#打开test2.html文件，会发现此时会保持我们的登录信息，为已登录状态。也就是说，对应的登录状态已经通过Cookie保存。*reponse2=urllib.request.urlopen(url2)  
fhandle2=open('./data2.html','wb')  
fhandle2.write(reponse2.read())  
fhandle2.close()

install\_opener(opener) 安装opener作为urlopen()使用的全局URL opener，即意味着以后调用urlopen()时都会使用安装的opener对象。

**保存cookie：**

filename = 'cookie.txt'

cookie = http.cookiejar.MozillaCookieJar(filename)

handler = urllib.request.HTTPCookieProcessor(cookie)

opener = urllib.request.build\_opener(handler)

response = opener.open('http://www.baidu.com')

cookie.save(ignore\_discard=True, ignore\_expires=True)

**使用cookie：**

import http.cookiejar, urllib.request

cookie = http.cookiejar.MozillaCookieJar()

cookie.load('cookie.txt', ignore\_discard=True, ignore\_expires=True)

handler = urllib.request.HTTPCookieProcessor(cookie)

opener = urllib.request.build\_opener(handler)

response = opener.open('http://www.baidu.com')

print(response.read().decode('utf-8'))

## urllib.error的使用

用 try-except来捕捉异常,主要的错误方式就两种 URLError（错误信息）和HTTPError(错误编码).

try:

data=urllib.request.urlopen(url)

print(data.read().decode('utf-8'))

except urllib.error.HTTPError as e:

print(e.code)

except urllib.error.URLError as e:

print(e.reason)