大家好哈，上一节我们研究了一下爬虫的异常处理问题，那么接下来我们一起来看一下Cookie的使用。

为什么要使用Cookie呢？

Cookie，指某些网站为了辨别用户身份、进行session跟踪而储存在用户本地终端上的数据（通常经过加密）

比如说有些网站需要登录后才能访问某个页面，在登录之前，你想抓取某个页面内容是不允许的。那么我们可以利用Urllib2库保存我们登录的Cookie，然后再抓取其他页面就达到目的了。

在此之前呢，我们必须先介绍一个opener的概念。

**1.Opener**

当你获取一个URL你使用一个opener(一个urllib2.OpenerDirector的实例)。在前面，我们都是使用的默认的opener，也就是urlopen。它是一个特殊的opener，可以理解成opener的一个特殊实例，传入的参数仅仅是url，[data](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=19492f893907b29d&k=data&k0=data&kdi0=0&luki=8&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=9db20739892f4919&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F968%2Ehtml&urlid=0" \t "_blank)，timeout。

如果我们需要用到Cookie，只用这个opener是不能达到目的的，所以我们需要创建更一般的opener来实现对Cookie的设置。

**2.Cookielib**

cookielib模块的主要作用是提供可存储cookie的对象，以便于与urllib2模块配合使用来访问Internet资源。 Cookielib模块非常强大，我们可以利用本模块的CookieJar类的对象来捕获cookie并在后续连接请求时重新发送，比如可以实现模拟登录 功能。该模块主要的对象有CookieJar、FileCookieJar、MozillaCookieJar、LWPCookieJar。

它们的关系：CookieJar —-派生—->FileCookieJar —-派生—–>MozillaCookieJar和LWPCookieJar

**1）获取Cookie保存到变量**

首先，我们先利用CookieJar对象实现获取cookie的功能，[存储](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=19492f893907b29d&k=%B4%E6%B4%A2&k0=%B4%E6%B4%A2&kdi0=0&luki=7&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=9db20739892f4919&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F968%2Ehtml&urlid=0)到变量中，先来感受一下

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | import urllib2  import cookielib  #声明一个CookieJar对象实例来保存cookie  cookie = cookielib.CookieJar()  #利用urllib2库的HTTPCookieProcessor对象来创建cookie处理器  handler=urllib2.HTTPCookieProcessor(cookie)  #通过handler来构建opener  opener = urllib2.build\_opener(handler)  #此处的open方法同urllib2的urlopen方法，也可以传入request  response = opener.open('http://www.baidu.com')  for item in cookie:      print 'Name = '+item.name      print 'Value = '+item.value |

我们使用以上方法将cookie保存到变量中，然后打印出了cookie中的值，运行结果如下

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Name = BAIDUID  Value = B07B663B645729F11F659C02AAE65B4C:FG=1  Name = BAIDUPSID  Value = B07B663B645729F11F659C02AAE65B4C  Name = H\_PS\_PSSID  Value = 12527\_11076\_1438\_10633  Name = BDSVRTM  Value = 0  Name = BD\_HOME  Value = 0 |

**2）保存Cookie到文件**

在上面的方法中，我们将cookie保存到了cookie这个变量中，如果我们想将cookie保存到文件中该怎么做呢？这时，我们就要用到

FileCookieJar这个对象了，在这里我们使用它的子类MozillaCookieJar来实现Cookie的保存

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | import cookielib  import urllib2    #设置保存cookie的文件，同级目录下的cookie.txt  filename = 'cookie.txt'  #声明一个MozillaCookieJar对象实例来保存cookie，之后写入文件  cookie = cookielib.MozillaCookieJar(filename)  #利用urllib2库的HTTPCookieProcessor对象来创建cookie处理器  handler = urllib2.HTTPCookieProcessor(cookie)  #通过handler来构建opener  opener = urllib2.build\_opener(handler)  #创建一个请求，原理同urllib2的urlopen  response = opener.open("http://www.baidu.com")  #保存cookie到文件  cookie.save(ignore\_discard=True, ignore\_expires=True) |

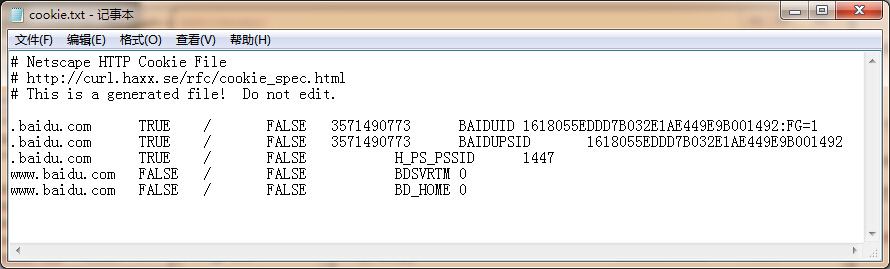
关于最后save方法的两个参数在此说明一下：

官方解释如下：

*ignore\_discard: save even cookies set to be discarded.*

*ignore\_expires: save even cookies that have expiredThe file is overwritten if it already exists*

由此可见，ignore\_discard的意思是即使cookies将被丢弃也将它保存下来，ignore\_expires的意思是如果在该文件中 cookies已经存在，则覆盖原文件写入，在这里，我们将这两个全部设置为True。运行之后，cookies将被保存到cookie.txt文件中， 我们查看一下内容，附图如下

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2015/02/QQ%E6%88%AA%E5%9B%BE20150215215136.jpg)

**3）从文件中获取Cookie并访问**

那么我们已经做到把Cookie保存到文件中了，如果以后想使用，可以利用下面的方法来读取cookie并访问网站，感受一下

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | import cookielib  import urllib2    #创建MozillaCookieJar实例对象  cookie = cookielib.MozillaCookieJar()  #从文件中读取cookie内容到变量  cookie.load('cookie.txt', ignore\_discard=True, ignore\_expires=True)  #创建请求的request  req = urllib2.Request("http://www.baidu.com")  #利用urllib2的build\_opener方法创建一个opener  opener = urllib2.build\_opener(urllib2.HTTPCookieProcessor(cookie))  response = opener.open(req)  print response.read() |

设想，如果我们的 cookie.txt 文件中保存的是某个人登录百度的cookie，那么我们提取出这个cookie文件内容，就可以用以上方法模拟这个人的账号登录百度。

**4）利用cookie模拟网站登录**

下面我们以我们学校的[教育系统](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=19492f893907b29d&k=%BD%CC%D3%FD%CF%B5%CD%B3&k0=%BD%CC%D3%FD%CF%B5%CD%B3&kdi0=0&luki=2&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=9db20739892f4919&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F968%2Ehtml&urlid=0)为例，利用cookie实现模拟登录，并将cookie信息保存到文本文件中，来感受一下cookie大法吧！

注意：密码我改了啊，别偷偷登录本宫的选课系统 o(╯□╰)o

Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | import urllib  import urllib2  import cookielib    filename = 'cookie.txt'  #声明一个MozillaCookieJar对象实例来保存cookie，之后写入文件  cookie = cookielib.MozillaCookieJar(filename)  opener = urllib2.build\_opener(urllib2.HTTPCookieProcessor(cookie))  postdata = urllib.urlencode({  'stuid':'201200131012',  'pwd':'23342321'  })  #登录教务系统的URL  loginUrl = 'http://jwxt.sdu.edu.cn:7890/pls/wwwbks/bks\_login2.login'  #模拟登录，并把cookie保存到变量  result = opener.open(loginUrl,postdata)  #保存cookie到cookie.txt中  cookie.save(ignore\_discard=True, ignore\_expires=True)  #利用cookie请求访问另一个网址，此网址是成绩查询网址  gradeUrl = 'http://jwxt.sdu.edu.cn:7890/pls/wwwbks/bkscjcx.curscopre'  #请求访问成绩查询网址  result = opener.open(gradeUrl)  print result.read() |

以上程序的原理如下

创建一个带有cookie的opener，在访问登录的URL时，将登录后的cookie保存下来，然后利用这个cookie来访问其他网址。

如登录之后才能查看的[成绩](http://cpro.baidu.com/cpro/ui/uijs.php?c=news&cf=1001&ch=0&di=8&fv=17&jk=19492f893907b29d&k=%B3%C9%BC%A8&k0=%B3%C9%BC%A8&kdi0=0&luki=6&n=10&p=baidu&q=35091180_cpr&rb=0&rs=1&seller_id=1&sid=9db20739892f4919&ssp2=1&stid=0&t=tpclicked3_hc&tu=u1959022&u=http%3A%2F%2Fcuiqingcai%2Ecom%2F968%2Ehtml&urlid=0)查询呀，本学期课表呀等等网址，模拟登录就这么实现啦，是不是很酷炫？

好，小伙伴们要加油哦！我们现在可以顺利获取网站信息了，接下来就是把网站里面有效内容提取出来，下一节我们去会会正则表达式！