**4.6.1 Cookies池简介**

Cookies是因为HTTP协议是无状态的，即服务器不知道用户上一次做了什么，这严重阻碍了交互式Web应用程序的实现。在典型的网上购物场景中，用户浏览了几个页面，买了一盒饼干和两饮料。最后结帐时，由于HTTP的无状态性，不通过额外的手段，服务器并不知道用户到底买了什么。 所以Cookie就是用来绕开HTTP的无状态性的“额外手段”之一。服务器可以设置或读取Cookies中包含信息，借此维护用户跟服务器会话中的状态。

从cookies的定义可以看出，cookies也是可以作为一个验证用户身份的工具，所以可以通过cookies来区别机器和人，所以有一种反爬的策略，就是通过cookies，微博的反爬虫机制就是基于cookies，所以同一个cookies可以重复请求，而同一个IP不带cookies却是不能重复请求，会封IP，但是微博做的还是不够，就是只需要一个cookies，但是微博的cookies所保存的时间短一点，就需要一个cookies池了，定期加入cookies，这样爬取的难度就会增大很多，再厉害一点，就是IP和cookies一起识别。

微博是这AJAX一类的网站，这一类网站的特点和上一种有点类似，但是为什么要分开来说呢？因为上一种情形一般会和源码一起结合出现，而Ajax网站，需要用户构造表单，向服务器要求返回想要的内容，而且这一类网站，一般是几乎所有信息（除了导航栏之类的通用信息）外，都是通过json, xml返回的，所以只需要从这个json包中便能获取所有数据。

**4.6.2构建Cookies池**

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关于重写中间件的详细

情况看官方文档：

http://scrapychs.readthedocs.io/zh\_CN/latest/topics/downloadermiddleware.html#scrapy.contrib.downloadermiddleware.DownloaderMiddleware

在项目中新建一个middlewares.py的文件（如果你使用的新版本的Scrapy，在新建的时候会有这么一个文件，直接用就好了）首先导User AgentMiddleware 毕竟我们要重写它啊第一行：定义了一个类UserAgentmiddleware继承自UserAgentMiddleware

第二行：定义了函数process\_request(request, spider)为什么定义这个函数，因为Scrapy每一个request通过中间件都会调用这个方法。下面就需要登陆了。这次我们不用上一篇博文的FromRequest来实现登陆了。我们来使用Cookie登陆。这样的话我们需要重写Cookie中间件！分布式爬虫啊！你不能手动的给每个Spider写一个Cookie吧。而且你还不会知道这个Cookie到底有没有失效。所以我们需要维护一个Cookie池(这个cookie池用redis)。

好！来理一理思路，维护一个Cookie池最基本需要具备些什么功能呢？

* 获取Cookie
* 更新Cookie
* 删除Cookie
* 判断Cookie是否可用进行相对应的操作（比如重试）

好，我们先做前三个对Cookie进行操作。首先我们在项目中新建一个cookies.py的文件用来写我们需要对Cookie进行的操作

首先我们把登陆用的账号密码 以Key:value的形式存入redis数据库。不推荐使用db0（这是Scrapy-redis默认使用的，账号密码单独使用一个db进行存储。）解决第一个问题：获取Cookie：使用requests模块提交表单登陆获得Cookie，返回一个通过Json序列化后的Cookie（如果不序列化，存入Redis后会变成Plain Text格式的，后面取出来Cookie就没法用啦。）第二个问题：将Cookie写入Redis数据库（分布式呀，当然得要其它Spider也能使用这个Cookie了）使用我们上面建立的redis链接获取redis db2中的所有Key(我们设置为账号的哦！)，再从redis中获取所有的Value(我设成了密码哦！)判断这个spider和账号的Cookie是否存在，不存在 则调用get\_cookie函数传入从redis中获取到的账号密码的cookie；保存进redis，Key为spider名字和账号密码，value为cookie。

以下是middlewares.py代码部分：

# encoding=utf-8

import random

from cookies import cookies

from user\_agents import agents

# from getproxies import proxies

import json

class UserAgentMiddleware(object):

""" 换User-Agent """

def process\_request(self, request, spider):

agent = random.choice(agents)

request.headers.setdefault("User-Agent", agent)

# request.headers["User-Agent"] = agent

class CookiesMiddleware(object):

""" 换Cookie """

def process\_request(self, request, spider):

cookie = json.loads(random.choice(cookies))

# print cookie

request.cookies = cookie

以下是cookies.py代码部分：

*import* base64  
*import* requests  
*import* sys  
*import* time  
*from* selenium *import* webdriver  
*from* selenium.webdriver.common.desired\_capabilities *import* DesiredCapabilities  
*from* selenium.webdriver *import* ActionChains  
*from* selenium.webdriver.common.by *import* By  
*from* selenium.webdriver.support.ui *import* WebDriverWait  
*from* selenium.webdriver.support *import* expected\_conditions *as* EC  
*import* logging  
*from* yumdama *import* identify  
*import* json  
  
reload(sys)  
sys.setdefaultencoding('utf-8')  
IDENTIFY = 1 # 验证码输入方式: 1:看截图aa.png，手动输入 2:云打码  
COOKIE\_GETWAY =2 # 0 从https://login.sina.com.cn/sso/login.php?client=ssologin.js(v1.4.18) 获取cookie # 1 从https://weibo.cn/login/获取Cookie # 2 使用chromedriver获取  
dcap = dict(DesiredCapabilities.PHANTOMJS) # PhantomJS需要使用老版手机的user-agent，不然验证码会无法通过  
dcap["phantomjs.page.settings.userAgent"] = (  
 "Mozilla/5.0 (Linux; U; Android 2.3.6; en-us; Nexus S Build/GRK39F) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1"  
)  
logger = logging.getLogger(\_\_name\_\_)  
logging.getLogger("selenium").setLevel(logging.WARNING) # 将selenium的日志级别设成WARNING，太烦人  
  
myWeiBo = [  
{'no':'15874173914','psw':'bbx8514'},  
{'no':'15774109579','psw':'bbx4768'},  
{'no':'17877727541','psw':'bbx3464'},  
{'no':'15898562769','psw':'bbx8712'},  
{'no':'18407320608','psw':'bbx2145'},  
{'no':'15973424313','psw':'bbx8431'},  
{'no':'13762898341','psw':'bbx3186'},  
{'no':'18374112533','psw':'bbx9829'},  
{'no':'15274883774','psw':'bbx8748'},  
{'no':'13873384591','psw':'bbx7247'},  
{'no':'13974708834','psw':'bbx2579'},  
{'no':'18474777738','psw':'bbx3957'},  
{'no':'18397779843','psw':'bbx4491'},  
{'no':'15197752390','psw':'bbx1831'},  
{'no':'15273563186','psw':'bbx9756'},

*def* getCookie(*account*, *password*):  
 *if* COOKIE\_GETWAY == 0:  
 *return* get\_cookie\_from\_login\_sina\_com\_cn(*account*, *password*)  
 *elif* COOKIE\_GETWAY == 1:  
 *return* get\_cookie\_from\_weibo\_cn(*account*, *password*)  
 *elif* COOKIE\_GETWAY == 2:  
 *return* get\_cookie\_from\_weibo(*account*, *password*)  
 *else*:  
 logger.error("COOKIE\_GETWAY Error!")  
  
  
*def* get\_cookie\_from\_login\_sina\_com\_cn(*account*, *password*):  
 """ 获取一个账号的Cookie """  
 loginURL = "https://login.sina.com.cn/sso/login.php?client=ssologin.js(v1.4.18)"  
 username = base64.b64encode(*account*.encode("utf-8")).decode("utf-8")  
 postData = {  
 "entry": "sso",  
 "gateway": "1",  
 "from": "null",  
 "savestate": "30",  
 "useticket": "0",  
 "pagerefer": "",  
 "vsnf": "1",  
 "su": username,  
 "service": "sso",  
 "sp": *password*,  
 "sr": "1440\*900",  
 "encoding": "UTF-8",  
 "cdult": "3",  
 "domain": "sina.com.cn",  
 "prelt": "0",  
 "returntype": "TEXT",  
 }  
 session = requests.Session()  
 r = session.post(loginURL, data=postData)  
 jsonStr = r.content.decode("gbk")  
 info = json.loads(jsonStr)  
 *if* info["retcode"] == "0":  
 logger.warning("Get Cookie Success!( Account:%s )" % *account*)  
 cookie = session.cookies.get\_dict()  
 *return* json.dumps(cookie)  
 *else*:  
 logger.warning("Failed!( Reason:%s )" % info["reason"])  
 *return* ""  
  
  
*def* get\_cookie\_from\_weibo\_cn(*account*, *password*):  
 """ 获取一个账号的Cookie """  
 *try*:  
 browser = webdriver.PhantomJS(executable\_path='E:\\phantomjs\\bin\\phantomjs.exe',desired\_capabilities=dcap)  
 browser.get("https://weibo.cn/login/")  
 time.sleep(1)  
  
 failure = 0  
 *while* "微博" *in* browser.title *and* failure < 5:  
 failure += 1  
 browser.save\_screenshot("aa.png")  
 username = browser.find\_element\_by\_name("mobile")  
 username.clear()  
 username.send\_keys(*account*)  
  
 psd = browser.find\_element\_by\_xpath('//input[@type="password"]')  
 psd.clear()  
 psd.send\_keys(*password*)  
 *try*:  
 code = browser.find\_element\_by\_name("code")  
 code.clear()  
 *if* IDENTIFY == 1:  
 code\_txt = raw\_input("请查看路径下新生成的aa.png，然后输入验证码:") # 手动输入验证码  
 *else*:  
 *from* PIL *import* Image  
 img = browser.find\_element\_by\_xpath('//form[@method="post"]/div/img[@alt="请打开图片显示"]')  
 x = img.location["x"]  
 y = img.location["y"]  
 im = Image.open("aa.png")  
 im.crop((x, y, 100 + x, y + 22)).save("ab.png") # 剪切出验证码  
 code\_txt = identify() # 验证码打码平台识别  
 code.send\_keys(code\_txt)  
 *except* Exception, e:  
 *pass* commit = browser.find\_element\_by\_name("submit")  
 commit.click()  
 time.sleep(3)  
 *if* "我的首页" *not in* browser.title:  
 time.sleep(4)  
 *if* '未激活微博' *in* browser.page\_source:  
 *print* '账号未开通微博'  
 *return* {}  
  
 cookie = {}  
 *if* "我的首页" *in* browser.title:  
 *for* elem *in* browser.get\_cookies():  
 cookie[elem["name"]] = elem["value"]  
 logger.warning("Get Cookie Success!( Account:%s )" % *account*)  
 *return* json.dumps(cookie)  
 *except* Exception, e:  
 logger.warning("Failed %s!" % *account*)  
 logger.warning(e)  
 *return* ""  
 *finally*:  
 *try*:  
 browser.quit()  
 *except* Exception, e:  
 *pass  
  
  
def* get\_cookie\_from\_weibo(*username*, *password*):  
 driver = webdriver.Chrome()  
 driver.get('https://weibo.cn//login/')  
 time.sleep(10)  
 *assert* u"微博" *in* driver.title  
 login\_link = driver.find\_element\_by\_link\_text(u'登录')  
 ActionChains(driver).move\_to\_element(login\_link).click().perform()  
 login\_name = WebDriverWait(driver, 10).until(  
 EC.visibility\_of\_element\_located((By.ID, "loginName"))  
 )  
 login\_password = driver.find\_element\_by\_id("loginPassword")  
 login\_name.send\_keys(*username*)  
 login\_password.send\_keys(*password*)  
 login\_button = driver.find\_element\_by\_id("loginAction")  
 login\_button.click() #自动按下登陆  
 WebDriverWait(driver, 30).until(EC.title\_is(u"我的首页")) #等待过了验证到首页  
 cookie = driver.get\_cookies()  
 driver.close()  
 *return* json.dumps(cookie)  
  
  
*def* getCookies(*weibo*):  
 """ 获取Cookies """  
 cookies = []  
 *for* elem *in weibo*:  
 account = elem['no']  
 password = elem['psw']  
 cookie = getCookie(account, password)  
 # logger.warning(type(cookie))  
 *if* cookie != None:  
 cookies.append(cookie)  
  
 *return* cookies  
  
  
cookies = getCookies(myWeiBo)  
logger.warning("Get Cookies Finish!( Num:%d)" % len(cookies))