使用 Python 在 SNS 获取和发布信息

施舒阳

（上海交通大学电子信息与电气工程学院，上海 200240）

摘　要：随着互联网的发展，社交网络的用户数量成指数级增长。以人人网为例，列举了使用 Python 登录人人网、发布状态、踩好友以及获取所有人状态的方法。用类似的方法可以推广到其他的网站信息获取和信息发布，从而方便实际使用和科研数据的获取。

关键词：Python 语言；社交网络；信息获取；网络爬虫

1　引言

Python 是一种面向对象、解释型计算机程序设计语言。Python 语法简洁而清晰，具有丰富和强大的类库。自从 20 世纪 90 年代初 Python 语言诞生至今，它逐渐被广泛应用于处理系统管理任务和 Web 编程。使用 Python 的库可以方便地登录网站，爬取网络上的信息。人人网是一个真实的社交网络，可以联络你和你周围的朋友。列举使用 Python 登录人人网、发布状态、踩好友、获取所有人状态的方法，通过此方法还可以写出各种有趣的应用，比如人人大笨钟（与新浪微博上古城钟楼相同），小黄鸡；也可以将获得的信息数据作为分析人际网络和用户行为等的数据来源，进行社会学、心理学的研究。该方法使用的 Python 版本为 2.7。

2　cookie

cookie 是一种服务器留在用户电脑中的小文件。常用来对用户进行识别每当同一台电脑通过浏览器请求页面时，这台电脑也会发送 cookie。为了让服务器认识用户，需要在程序运行过程中保留用户的 cookie。由于人人网每天对于同一个 IP 的登录次数有限制，超过一定数之后需要输入验证码，所以还需要将登录成功后的 cookie 保存在本地，以便在下一次跑程序的时候使用。Python 中使用 cookielib 库完成此操作：

```
import cookielib
cj = cookielib.MozillaCookieJar()
  opener = urllib2.build_opener(urllib2.HTTPCookieProcessor
(cj))
urllib2.install_opener(opener) # 安装 cookie
```

安装完的 cookie 可以认为是全局量，在整个程序中通用。在登录之后，使用

```
cj.add_cookie_header(req) # 将新的 cookie 加入 cj
urllib2.urlopen('http://www.renren.com')
cj.save (email, ignore_discard=True, ignore_expires=True) #
储存 cookie
```

能够储存成功登录的 cookie，其中 email 是登录人人的账号，这里也作为保存的 cookie 的文件名。

在第一次登录成功并存储后，接下来就可以读取本地存储的 cookie 了。在安装完 cookie 之后执行：

```
cj.load(email, True, True)
```

就进入到了登录的状态。

3　登录

网页间传递消息有 POST 和 GET 两种方式。使用 GET 方法从表单传送的信息对所有的用户都是可见的（出现在浏览器的地址栏），并且对所发送信息的量也有限制。使用 POST 方法从表单传送的信息对用户是不可见的，并且对所发送信息的量也没有限制。一般网站登录均使用的是 POST 方法。使用 chrome 或者火狐浏览器的开发者工具（F12 打开 developer tools）可以看到这些信息，如图 1 所示。从中可以看出用户名和密码被送到 http://www.renren.com/ajaxLogin/login。

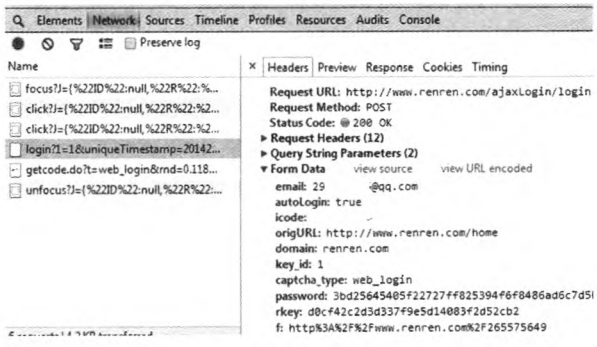

图 1　使用开发者工具看到的登录请求信息

经过多次测试发现其中发送的消息中有用的是

```
information = {'email':email, 'password':pw, 'origURL':'http://
www.renren.com/Home.do', 'domain':'renren.com', 'key_id':
'1', 'autoLogin':'True'}
```

其中 email 是用户名，pw 是账号。打开网页需要用到

作者简介：施舒阳（1992-），男，本科，研究方向：电子信息与计算机技术。

urllib，urllib2 两个库。使用postdata=urllib.urlencode(information)将之前的信息变成字符串类型，这个类型能够发送给服务器端。

```
req = urllib2.Request(url = 'http://www.renren.com/ajaxLogin/login', data = postdata)
```

部分网站需要获取Request Headers 中的 User-Agent 信息。以此可以判断当前使用的浏览器类型，来提供适合用户浏览器的界面（比如移动设备和电脑显示的东西是不同的）。以下就是模拟的这一信息：

```
req.add_header('User-Agent', 'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.802.30 Safari/535.1 SE 2.X MetaSr 1.0')
```

接下来就可以发送登录消息了：

```
urllib2.urlopen(req)。
```

如果登录成功，使用：

```
content = urllib2.urlopen('http://www.renren.com').read()
print content
```

能够看到好友的新鲜事等各种消息的源 HTML 代码（content 的数据类型为 string）！这个时候就可以保存 cookie 了。在源代码中有一部分是：

```
<script>XN={get_check:'235515844',get_check_x:'d9caae1e',env: {domain:'renren.com',shortsteName:' 人人 ',siteName:'人人网 '}};
```

其中的 get_check_x 后的一段字符串是人人的验证码，下面称为 rtk，使用 string 的一些方法可以简单地获取。这个 rtk 每次登录可能不同，需要每次记录下来，之后发状态，看好友状态等操作均需要此验证码。

4 发状态

登录之后最常用的应该就是发状态了，人人网发状态同样使用的是 POST。方法和登录类似，这里只表结果：

```
postdata=urllib.urlencode({'content':状态内容, 'hostid':你的ID, '_rtk':rtk, 'channel':'renren'})
req = urllib2.Request (url = "http://shell.renren.com/你的 ID/status", data = postdata)
content = urllib2.urlopen(req)
```

如果状态内容中存在中文，要使用的是 UTF-8 码，其中 rtk 就是上面所述的验证码。无论发布成功与否，都会有返回结果告诉你。

5 踩好友

用开发者工具能够找到人人网的好友名单，发现全部好友名单以 JSON 形式保存在 http://friend.renren.com/groupsdata 中。JSON 是 JavaScript 对象表示法（JavaScript Object Notation）。JSON 是存储和交换文本信息的语法。类似 XML，但是它比 XML 更小、更快，更易解析。Python 中的 JSON 库可以将 JSON 转换为 Python 的数据类型，它的对应关系如表 1。

表 1 Ptyhon 与 JSON 的对应关系

JSON	Python
object	dict
array	list
string	unicode
number (int)	int, long
number (real)	float
true	True
false	False
null	None

用 content =urllib2.urlopen ('http://friend.renren.com/groupsdata') .read () 读取其中的内容，并且截取返回字符串中 JSON 的部分，然后使用：

```
jsonList = json.loads(content[起始位置: 终点位置])
renrenids = [] # 储存好友 ID 的列表
renrennames = [] # 储存好友名字的列表
for i in jsonList:
    renrenids.append(i["fid"])
    renrennames.append(i["fname"])
```

将好友的 ID 和名字储存在列表中，接下来只要循环使用：

```
urllib2.urlopen('http://www.renren.com/' + renrenid[i])
```

就能够踩好友了。由于人人网最近加入了踩 100 个人需要输入验证码的功能，所以这个方法还是需要手动输入几下验证码的。

6 读取好友的状态

同样用开发者工具能够找到状态的 URL：

```
url = "http://status.renren.com/GetSomeomeDoingList.do?userId="+ id + "&curpage=" + str(i)
```

其中 ID 是状态所有者的 ID，i 为状态的页数。这个方法找到的状态不限于自己的好友，整个人人网上的所有人的状态均能被爬取。打开这个网页后发现同样是个 JSON，同样可以使用上面的方法来解析。具体的每条状态的内容这里不再叙述，只解析每个状态的 ID 号，状态的 ID 号用于读取状态下面的回复：

```
jsonList = json.loads(content)
for j in jsonList["likeMap"]:
    statusid = j[7:]
```

读取状态 ID 以后，使用 post 打开可以看到状态下面的回复：

```
postdata=urllib.urlencode({'doingid':statusid, 'source':statusid, 'owner':renrenid, '_rtk':rtk, 't':3})
req = urllib2.Request(url = 'http://status.renren.com/feedcommentretrieve.do', data = postdata)
```

（下转第 89 页）

```
}
```

有了这些代码，日后需要用到 ASPxpivotGrid 控件来统计分析数据的时候，只需要简单地写下两行代码即可：

```
ZiyuWeb.WebFunc.ZiyuDevAspxPivotGridHelper.ConfigPivotGrid("hzsendtotal", true, "sendate", ASPxPivotGrid1, true, "chart");//配置 ASPxPivotGrid 控件;
//下面两行代码是增加计算列的,如不需要,则保持原样,否//则打开注释并根据需要调整代码即可;
// ZiyuWeb.WebFunc.ZiyuDevAspxPivotGridHelper.addCom//puteField("面积", string.Empty, "[gg]*[length]/1000", ASPxPivot//Grid1, DevExpress.Data.PivotGrid.PivotSummaryTy//pe.Sum);
```

完整的代码和界面，请读者朋友们查看本文的示例项目。

在实际使用过程中发现，ASPxpivotGrid 控件在面对大批量数据进行统计分析的时候，页面打开会比较慢，甚至会失去响应，这便有了一个问题：如何提高它的运行效率呢？

4 提高 AspxPivotGrid 运行效率

ASPxPivotGrid 组件是一个基于 ASP.NET 平台下的全面的数据分析、数据挖掘和可视化报表的解决方案。它的出现不仅可以为新的解决方案去除数据分析方面的种种缺陷，也可以从根本上改善已有的大型数据分析软件在最终数据呈现上的不足，从而让最终用户能更好获取和分析相关数据。那么如何进一步提高 ASPxPivotGrid 的性能，对数据进行高效切分，从而为客户提供一个非常直观的终端用户体验呢？这里从两方面来说明问题。

（1）选择一个合适的数据源

并不是所有的数据源都符合目标程序。如果每次访问的数据量在 150,000 以下的话，最好使用 SQLSever 数据库，如果数据量超过 150,000 行的话，将考虑使用 Analysis Services?即分析数据库，否则运行速度会很慢。当然如果选择一个好的数据库服务器，可以大于上面的数据，反之如果使用 xml 或者 MS Access 数据存储格式就可能更少了。

那么如何选择一个合适的数据库呢？首先，需要估计一下数据量，PivotGrid 每次发送请求的一个数据量。如果不知道实际的记录数，可以根据列的成员特性推测出来。例如，有 7 列：产品种类 (50)，产品子类别 (1000)，产品 (50000)，国家 (10)，省 (400)，市 (10000)，顾客 (50000)，这里括号中的数字是对应成员特有的记录数。分析一下，上面“产品种类、产品子类别、国家、省和市”其实根本就没有提供新的度量数据，“产品种类”和“产品子类别”其实是包含在“产品”中，而“国家,省和市”所产生的度量数据则体现在“顾客”信息记录中。因此很快就能通过“产品”和“顾客”的乘积得出记录数 (50 * 50000 = 2 500 000)。

（2）性能优化以及对比

在选择了不同的数据库后，又该怎么提高 PivotGrid 的性能呢？

当使用 SQL Server 作为 PivotGrid 的数据源的时候，总体性能大概由加载数据的时间、计算的时间和数据绘制时间 3 部分组成。为了提高数据加载时间，可以在数据服务器和 Web 服务器之间建立更好的连接。数据库服务器和 Web 服务器最好是使用同一台机器，调整数据服务器，以实现最佳的性能和缓存数据的检索。此外还可以在服务端优化查询语句，比如" select Category, Product, Sales from Sales" 查询改为" select Category, Product, sum (Sales) from Sales group by Category, Product" .这样很大程度上能减少 PivotGrid 本身计算的时间。

当使用分析服务器作为数据源的时候，对于每次请求是不存在重载和重新计算数据的。数据和计算结果都是作为缓存，返回的仅仅是请求的那部分数据。尽管这样，如果是小规模的查询那速度还是低于 SQL Server，但在大型数据集上将产生更好的效果。事实上，如果没有使用 Analysis Services 的数据源，对于上百万条的数据是很难得到满意结果的。

参考文献

[1] http://www.devexpress.com.
[2] http://www.cnblogs.com.

（收稿日期：2013-12-12）

（上接第 53 页）

```
content = urllib2.urlopen(req).read()
```

此外还可以在状态下面回复、点赞等。方法类似，此处也不表了。

7 结语

Python 作为脚本语言，有强大的网络编程功能。人人网上的著名公共主页小黄鸡就是由 Python 实现的。上面就是以人人网为例子，叙述了 Python 在社交网站中发布和获取信息的方法，用简单的语言能够实现强大的功能，如果附加上爬虫程序还能获取更加多的信息。

参考文献

[1] Python Software Foundation. Python v2.7.6 documentation. http://docs.python.org/ Mar 21, 2014.
[2] w3school.com.cn. W3school 在线教程. http://w3school.com.cn/.
[3] http://baike.baidu.com/view/21087.htm.

（收稿日期：2013-12-29）