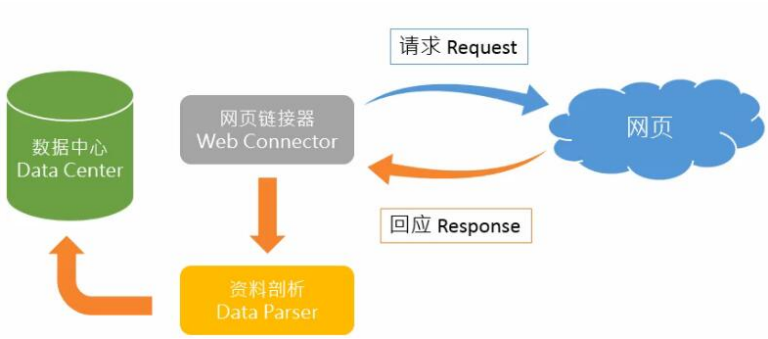
python 网络爬虫实战

课时1 非结构化数据的概念

- 网络中存在大量的非结构化数据
    - 没有固定的数据格式
    - 必须透过ETL(Extract, Transformation, Loading) 工具转化为结构化数据才能取用
- 原始资料(raw data) -> ETL脚本(ETL script) -> 结构化数据(tidy data)

课时2 非结构化数据处理与网络爬虫

- 如何处理非结构化数据
    - 借助网络爬虫（不是新技术，比如原始搜索引擎技术）
- 网络爬虫架构
    - 网页链接器向网页发出请求(Request)
    - 网页回应(Response)
    - 可通过**Inspect** in Chrome, Network 来查看Request 和 Response
    - 对回应的资料剖析(Data Parser)
    - 存入数据中心



课时3 了解网络爬虫背后的秘密

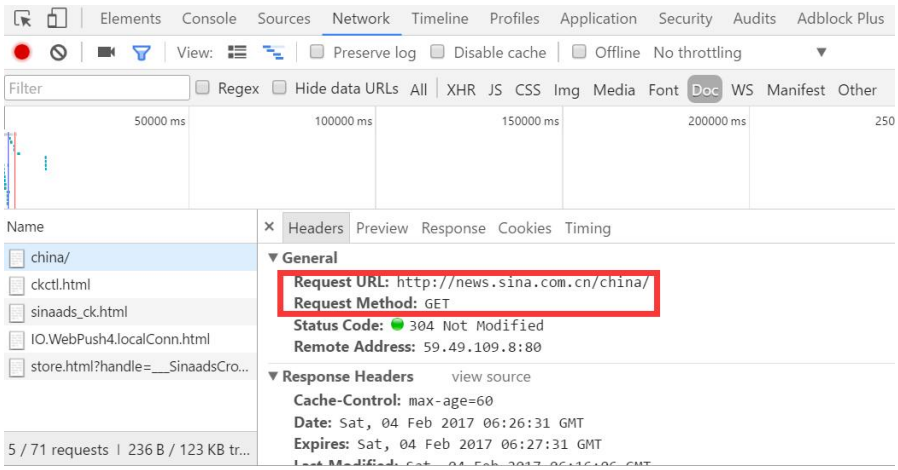
抓取新浪新闻标题和时间

- 使用开发人员工具（Inspect）
- 选择Network，and refresh website
- 用filter筛选Doc类别

> 因为媒体类信息有被搜索引擎抓取的需求，而且*Doc*类别下最容易被抓取。因此*90%*的媒体信息都可以在*Doc*下面找到

> 如何确定是否找到了正确的*Response*？
>
> 看源码里面有无对应信息

- 确认正确的request之后，切换到Headers tag



环境搭建 python

略

撰写第一只网络爬虫

```
import requests
res=resquests.get("http://news.sina.com.cn/china/")
res.encoding="utf-8" #encoding
print(res.text)
```

结果

```
<!DOCTYPE html>
<!-- [ published at 2017-02-04 14:45:23 ] -->
<html>
<head>
<meta http-equiv="Content-type" content="text/html;
charset=utf-8" />
<title>国内新闻_新闻中心_新浪网</title>
<meta name="keywords" content="国内时政,内地新闻">
<meta name="description" content="新闻中心国内频道，纵览国内
时政、综述评论及图片的栏目，主要包括时政要闻、内地新闻、港澳台新闻、
媒体聚焦、评论分析。">
<meta name="robots" content="noarchive">
<meta name="Baiduspider" content="noarchive">
<meta http-equiv="Cache-Control" content="no-transform">
<meta http-equiv="Cache-Control" content="no-siteapp">
<!--......-->
```

用BeautifulSoup4 剖析网页元素

如何把所获取的数据进行结构化？

- 透过Document Object Model可以将网页转化为DOM Tree， 之后我们可以对DOM Tree的节点进行操作

BeautifulSoup4 范例

- 将网页读入BS

```
from bs4 import BeautifulSoup
html_sample=' \
<html> \
 <body> \
 <h1 id="title">Hello World</h1> \
 <a href="#" class="link">This is link1</a> \
 <a href="# link2" class="link">This is link2</a> \
 </body> \
</html>'
soup=BeautifulSoup(html_sample, "html.parser")
print(soup.text)
```

> 在这段*html*代码之中， *'h1' 'a'* 为‘标签’；*id="title", class="link"* 为*CSS*属性，其中*id*为独一无二，而*class*可重复

---

result:

```
<class 'bs4.BeautifulSoup'>
   Hello World This is link1 This is link2
```

结果为去掉了所有标签，仅保留了文字部分

但是真正网页之中往往还有大量无用信息

因此下一步为

找出所有含特定标签的HTML元素

- 使用select找出含有**h1**标签的元素

```
soup=BeautifulSoup(html_sample,'html.parser')
hearder=soup.select('h1')
print(header)
print(header[0])
print(header[0].text)
```

> *header*为*python*内置的*'list'*类型
>
> *header[0]* 为类型
>
> *header[0].text* 为 类型

result:

```
[<h1 id="title">Hello World</h1>]
<h1 id="title">Hello World</h1>
Hello World
```

- 使用select找出含有**a**标签的元素

```
AllLink=soup.select('a')
for link in AllLink:
    print(link.text)
```

result:

```
This is link1
This is link2
```

取得含有特定CSS属性的元素

- 使用select找出所有id为‘title’的元素(id前面需要加#)

```
atitle=soup.select('#title')
for title in atitle:
    print(title.text)
```

result:

```
Hello World
```

- 使用select找出所有class为‘link’的元素(class前面需要加.)

```
alink=soup.select('.link')
for link in alink:
    print(link.text)
```

result:

```
This is link1
This is link2
```

取得所有a标签内的链接

- 使用select找出所有a tag的href链接

```
for link in alinks:
    print(link['href'])
```

result:

```
#
# link2
```

> 实质：*dict*
>
> ```
> a='<a href="#" qoo=123 abc=456>this is a link</a>'
> soup2=BeautifulSoup(a,'html.parser')
> print(soup2.select('a')[0]['abc'] )
> print(soup2.select('a')[0]['qoo'] )
> print(soup2.select('a')[0]['href'] )
> ```
>
> *result:*

```
456
123
3
```

注意事项

- soup.select['a']的结果类型为list类型，操作需要先取元素

source code：抓取新浪国内新闻的标题、时间和链接

```
import requests
res=requests.get("http://news.sina.com.cn/china/")
res.encoding="utf-8" #easy to forget
from bs4 import BeautifulSoup
soup=BeautifulSoup(res.text,"html.parser")#res.text
potential error
for news in soup.select(".news-item"):
     if(len(news.select('h2'))>0):
        header=news.select('h2')[0].text
        time=news.select('.time')[0].text
        url=news.select('a')[0]['href']
        print(time,header,url)
        #print(news.select('h2')[0]['href'])#wrong,
since href is in tag 'a'
```

result:

```
2月4日 18:45 浙江苍南开近5千人干部大会 人员规模前所未有
http://news.sina.com.cn/o/2017-02-04/doc-
ifyafenm2732440.shtml
2月4日 18:36 这个春节国人花了8400亿 银联网络交易达4620亿
http://news.sina.com.cn/o/2017-02-04/doc-
ifyaexzn8917810.shtml
2月4日 18:26 南方供暖尝试"破冰":杭州推天然气分户式供暖
http://news.sina.com.cn/o/2017-02-04/doc-
ifyafenm2731114.shtml
2月4日 18:23 全军首个押运兵模拟训练基地建成并投入运行
http://news.sina.com.cn/c/nd/2017-02-04/doc-
ifyafcyx6953782.shtml
2月4日 18:22 中国将首次种植全基因组育种芯片新稻种
http://news.sina.com.cn/c/nd/2017-02-04/doc-
ifyafcyx6953707.shtml
2月4日 18:19 美对中国钢铁产品作出双反仲裁 商务部表示不满
http://news.sina.com.cn/c/nd/2017-02-04/doc-
ifyafcyx6953320.shtml
2月4日 18:17 香港一垃圾站700箱过期薯条遭市民哄抢(图)
http://news.sina.com.cn/c/gat/2017-02-04/doc-
ifyafcyw0183170.shtml
2月4日 18:12 未成年网络保护条例将出台 专家:电击治疗非法
http://news.sina.com.cn/c/nd/2017-02-04/doc-
ifyafcyw0182437.shtml
...
```

应对另一种情况：拆分一个大标签下的两部分内容

span 里面镶嵌有 span

```
<span class="time-source" id="navtimeSource">2017年02月04
日22:08       <span>
<span data-sudaclick="media_name"><a
href="http://mp.weixin.qq.com/s?
__biz=MzIzNjIwNzI2Mw==&amp;mid=2653187721&amp;idx=1&amp;
sn=fc0ba8cc8182e718d9e58e58bb6c4421&amp;scene=0#wechat_r
edirect" rel="nofollow" target="_blank">法制晚报</a>
</span></span>
</span>
```

方法：

```
a=  '<span class="time-source" id="navtimeSource">2017年
02月04日22:08      <span>\
<span data-sudaclick="media_name"><a
href="http://mp.weixin.qq.com/s?
__biz=MzIzNjIwNzI2Mw==&mid=2653187721&idx=1&sn=fc0ba8cc8
182e718d9e58e58bb6c4421&scene=0#wechat_redirect"
target="_blank" rel="nofollow">法制晚报</a></span></span>\
    </span>'
soup2=BeautifulSoup(a,'html.parser')
time2=soup2.select('.time-source')
[0].contents[0].strip()
from datetime import datetime
objTime2=datetime.strptime(time2,'%Y年%m月%d
日%H:%M')#convert string to datetime object
source2=soup2.select('.time-source span a')[0].text
print("at",objTime2,"from",source2)
```

result:

```
at 2017-02-04 22:08:00 from 法制晚报
```

抓取新闻正文部分，并合并至一个string内

```
>>> ' '.join([p.text.strip() for p in
soup.select('#articleContent .left p')[:-1]])
'原标题：你们警察在街上到处"溜达"有什么用？网友评论亮了！ 走在大街
上，我们经常会看到这样的情景： 没错，就是三五警察在街上来回的"溜达"
最近，有网友不理解警察这样"溜达"有什么作用，于是向中国警察网发起了
提问： 你能放心地溜达正是因为这些尽职尽责警察的"溜达"！ 这个回答赢
得了很多网友以及警察同行的认同 他们"溜达"，一些别有用心的人才不敢出
来"溜达"： 他们"溜达"，人们才可以安心地享受生活： 他们"溜达"，迷失
的孩子才能找到家： 他们"溜达"，其实是拿生命在做代价： 风里雨里节日
里，他们从未停止"溜达"，正是因为这样，我们才能放心随意的溜达。向所
有奋战在一线的人民警察致敬！'
```

> 其中 `soup.select('#articleContent .left p')` 选中所有正文部分的标签， `[:-1]` 剔除了最后一条'责任编辑:*xxx*'
>
> 使用一行语句创建一个*list* ： `[x*x for x in range(10)]`
>
> 最后用 *'[char]'.join()* 连接*list*的所有元素至一个*string*类型中

抓取编辑人

```
>>> soup.select('.article-editor')[0].text.strip('责任编辑：').strip()
'隗俊'
```

[难] 抓取评论数

这个评论个数是通过js获取的，不会直接出现在doc里面。因此首先需要在Network里面找到对应的response（老师只提了地毯式搜索评论数字）

找到之后得到的是json格式，需要调用json.loads()解析为python dict类型然后操作

```
comment=requests.get('http://comment5.news.sina.com.cn/page/info?version=1&\
                     format=js&channel=gn&newsid=comos-fyafcyw0191857&group=&compress=0&\
                     ie=utf-8&oe=utf-8&page=1&page_size=20&jsvar=loader_1486283153548_11337534')
import json
json.loads(comment.text)['result']['count']['total']
```

result：

```
6768
```

剖析新闻标识符

给定链接: `http://news.sina.com.cn/c/nd/2017-02-04/doc-ifyafcyw0191857.shtml` 其中 `fyafcyw0191857` 为新闻的id，如何对string操作然后提取出id？

方法一 str.split() + str.strip() 分割，切除

```
>>> newsid='http://news.sina.com.cn/c/nd/2017-02-04/doc-ifyafcyw0191857.shtml'
>>> newsid.split('/')[-1].rstrip('.shtml').lstrip('doc-i')
```

result:

```
'fyafcyw0191857'
```

> 其中*newsid.split('/')*将字符串按'/'切分，再用*[-1]*取*list*最后一个元素，即*"doc-ifyafcyw0191857.shtml'*

方法二 正则表达式

```
import re
m=re.search('doc-i(.*).shtml',newsid)
m.group(1)
```

result:

```
'fyafcyw0191857'
```

> *m.group(0)*为匹配到的字符串
>
> *m.group(1)*为*(.*)*的内容

更进一步：建立评论数抽取函式

```
import re
import json
import requests
def getID(url):
    m=re.search('doc-i(.*).shtml',url)
    newsID=m.group(1)
    key='http://comment5.news.sina.com.cn/page/info?
version=1&\
format=js&channel=gn&newsid=comos-{}&group=&compress=0&\
ie=utf-8&oe=utf-8&page=1&page_size=20'
    key=key.format(newsID)
    Comment=requests.get(key)
    m=re.search('var(.*)={',Comment.text)
    Comment=Comment.text.lstrip(m.group(0)[:-1])
    Comment=json.loads(Comment)
    return Comment['result']['count']['total']

#call
url='http://news.sina.com.cn/o/2017-02-05/doc-
ifyafenm2755085.shtml'
getID(url)
#result:
#42
```

作业： 将以上所有内容写为一个函数，返回字典类型。

- 传入：新闻链接
- 传出：新闻details

```
import requests
from bs4 import BeautifulSoup
import re
import json
from datetime import datetime
def GetNewsDetails(url):
    #including title, time , source, body, writer,
number of comments
    result={}
    res=requests.get(url)
    res.encoding='utf-8'
    soup1=BeautifulSoup(res.text,'html.parser')
    result['hearder']=soup1.select('#artibodyTitle')
[0].text#header
    result['time']=soup1.select('.time-source')
[0].contents[0].strip()#time
    result['time']=datetime.strptime( result['time'],'%Y
年%m月%d日%H:%M')
    result['source']=soup1.select('.time-source a')
[0].text#source
    result['body']='\n'.join([p.text for p in
soup1.select('#artibody p')[:-1]])#body
    result['writer']=soup1.select('#artibody p')
[-1].text.lstrip('责任编辑：')#writer
    key='http://comment5.news.sina.com.cn/page/info?
version=1&\
format=js&channel=gn&newsid=comos-{}&group=&compress=0&\
ie=utf-8&oe=utf-8&page=1&page_size=20'
    news_id=re.search('doc-i(.*).shtml',url).group(1)
    key=key.format(news_id)
    res2=requests.get(key)
    #soup2=BeautifulSoup(res2.text,'html.parser')# not
needed
    flag=re.search('var(.*)={',res2.text)
    if(flag):
       res2=res2.text.lstrip(flag.group(0)[:-1])
    result['NumOfCom']=json.loads(res2)['result']
['count']['total']
    return result
```

```
>>>> GetNewsDetails('http://news.sina.com.cn/o/2017-02-05/doc-ifyafenm2755085.shtml')
{'NumOfCom': 47,
 'body': '\u3000\u3000原标题：今晨至上午河北等8省有大雾 局地能见度不足50米\n\u3000\u3000中国天气网讯 中央气象台2月5日06时继续发布大雾黄色预警：预计，5日早晨至上午，河北南部、河南中东部、山东西部、安徽北部、湖北中部、湖南北部、贵州中东部、广西中部等地有大雾，其中，河北南部、河南中东部、山东西南部、湖北中部、湖南北部、贵州中部等地的部分地区有能见度低于500米的浓雾，局地有能见度低于50米的特强浓雾。\n\u3000\u3000防御指南：\n\u3000\u30001、由于能见度较低，驾驶人员应控制速度，确保安全；\n\u3000\u30002、机场、高速公路、轮渡码头采取措施，保交通安全。',
 'hearder': '今晨至上午河北等8省大雾 局地能见度不足50米',
 'source': '中国天气网',
 'time': datetime.datetime(2017, 2, 5, 6, 41),
 'writer': '张冬 '}
```