Web数据交换

北京理工大学计算机学院　高玉金

2019年3月

Web数据获取的形式

- 第三方封装库，如Tushare，可以直接返回DataFrame对象
- 各类API网站，返回JSON格式文件或字符串，通过简单爬虫获取
- 普通静态Web页面，需要爬虫抓取并自行摘取数据
- 动态页面中的数据往往是由JavaScript脚本根据用户交互的不同而动态生成，普通的爬虫无法提供这种交互能力，在无真实用户交互的情况下，需要使用代码控制浏览器环境，模拟用户输入，从而才能得到所需数据

使用tushare财经库

- 如果尝试分析金融大数据，可以使用TuShare三方库工具
- 该三方库通过重新清洗公开免费的财经数据，提供了方便快捷的接入方式

```
import tushare as ts
```

- Dir(ts)查看相关信息
- 用help查看help(ts.get_gdp_year)
- data =ts.get_gdp_year()

```
In [10]: data.head()
Out[10]:
   year       gdp   pc_gdp   ...          ti  trans_industry     lbdy
0  2017  820754.0  59660.0   ...    425912.0        37173.00  92348.0
1  2016  743585.5  53935.0   ...    383365.0        33058.80  84648.8
2  2015  689052.1  50251.0   ...    346149.7        30487.80  78340.4
3  2014  643974.0  47203.0   ...    308058.6        28500.90  73582.0
4  2013  588018.8  43852.0   ...    275887.0        27282.93  66512.4
```

```
get_gdp_year()
        获取年度国内生产总值数据
    Return
    --------
    DataFrame
        year :统计年度
        gdp :国内生产总值(亿元)
        pc_gdp :人均国内生产总值(元)
        gnp :国民生产总值(亿元)
        pi :第一产业(亿元)
        si :第二产业(亿元)
        industry :工业(亿元)
        cons_industry :建筑业(亿元)
        ti :第三产业(亿元)
        trans_industry :交通运输仓储邮电通信业(亿元)
        lbdy :批发零售贸易及餐饮业(亿元)
```

使用tushare财经库

- Tushare返回的数据格式是DataFrame
- 可以使用索引、切片等各种方法进行数据选择和处理
- 保存到数据库、本地CSV文件等
- data.to_csv(“gdp.csv”)或
  Data.to_excel(“gdp.xslx”)
- 通过dir(data)可以查看DataFrame的输出方法

```
In [12]: type(data)
Out[12]: pandas.core.frame.DataFrame
```

```
In [20]: type(data['gdp'])
Out[20]: pandas.core.series.Series
```

```
'to_csv',
'to_dense',
'to_dict',
'to_excel',
'to_feather',
'to_gbq',
'to_hdf',
'to_html',
'to_json',
'to_latex',
'to_msgpack',
'to_panel',
'to_parquet',
'to_period',
'to_pickle',
'to_records',
'to_sparse',
'to_sql',
'to_stata',
'to_string',
'to_timestamp',
'to_xarray',
```

使用网站API接口

- 一些网站如豆瓣、百词斩等网站，提供API接口，根据用户查询信息，提供JSON格式的字符串或文件
- 浏览器输入http://mall.baicizhan.com/ws/search?w=dog
- 返回数据为JSON格式：

{ "word": "dog", "img": "http://assets.baicizhan.com/r/20140416_14_10_02_551.jpg", "st": "The dog rides the frog.", "sttr": "这只狗骑在了青蛙身上。", "mean_cn": "c. 狗", "tv": "http://assets.baicizhan.com/word_tv/noun_dog.mp4", "accent": "/dɔ:g/", "errorCode": 0 }

从JSON串获取数据

- Requests库的json()方法可以直接生成一个字典
- 即可按照字典的方式进行数据处理


```
1  import requests
2
3  words = ["dog","cat"]
4  for word in words:
5      r = requests.get("http://mall.baicizhan.com/ws/search?w={}".format(word)
6      print(r.text)
7      print(r.json()['img'])   #r.json is different from r.json()
8      print(u'http://baicizhan.qiniucdn.com/word_audios/{}.mp3'.format(word))
```

普通静态页面数据获取

- 通过requests库，可以很方便的获取静态页面数据
- Requests库只负责爬取页面数据转换成字符串，不负责解析
- bS4 库是一个解析和处理HTML 和XML 的第三方库
- from bs4 import beautifulsoup

```
>>>import requests
>>>from bs4 import BeautifulSoup
>>>r = requests.get("http://www.baidu.com")
>>>r.encoding = "utf-8"    #为了简化代码，没有考虑异常情况
>>>soup = BeautifulSoup(r.text)    #soup 就是一个 BeautifulSoup 对象
>>> type(soup)
<class 'bs4.BeautifulSoup'>
```

通过bs4库查找需要的数据

- 创建的BeautifulSoup 对象是一个树形结构，它包含HTML 页面里的每一个Tag（标签）元素，如<head>、<body>等
- 通过BeautifulSoup 的find()和find_all()方法遍历整个HTML 文档
- 通过tag 对象的name、attrs 和string 属性获得相应内容

```
from bs4 import BeautifulSoup

words = ["旗帜","计算机","乾坤","理工"]
for word in words:
    r = requests.get("http://hanyu.baidu.com/s?wd="+word+"&from=zici")
    r.encoding ="utf-8"
    soup = BeautifulSoup(r.text,'lxml')
    print("{:-^60}".format(word))
    for p in soup.find(id="basicmean-wrapper").div.dd:
        print(p.string.strip())
```

爬取动态网页内容

- 动态网页与静态网页不同，一般是由JavaScript代码根据用户交互的不同，动态产生页面数据
- Selenium是一个原本用于测试的自动化程序，可以用于操控无头浏览器产生真实的浏览器环境
- 需下载对应的driver，如chrome下chromedriver

```
chrome_options = Options()
chrome_options.add_argument("--headless")
base_url = "http://www.baidu.com/"
driver = webdriver.Chrome(chrome_options=chrome_options)
driver.get(base_url + "/")
driver.find_element_by_id("kw").send_keys("Python程序设计")
driver.find_element_by_id("su").click()
driver.save_screenshot('screen.png')
driver.close()
```