



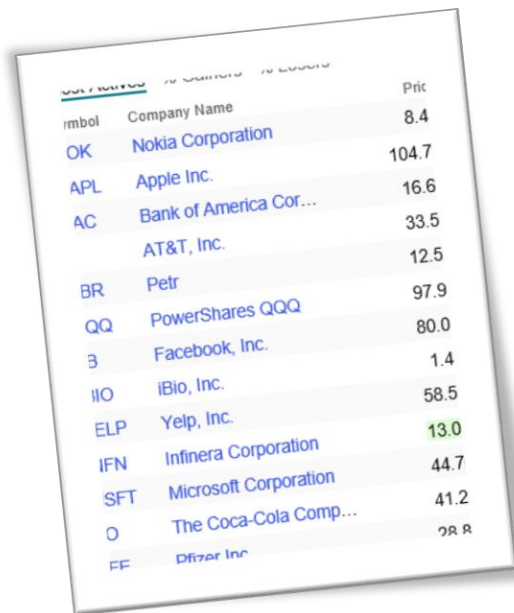
Where are data from?

网络数据获取入门

Zhang Li
Nanjing University

网络数据爬取

用Python获取网络数据



A tilted screenshot of a stock market data table. The table has three columns: 'Symbol', 'Company Name', and 'Price'. The data is as follows:

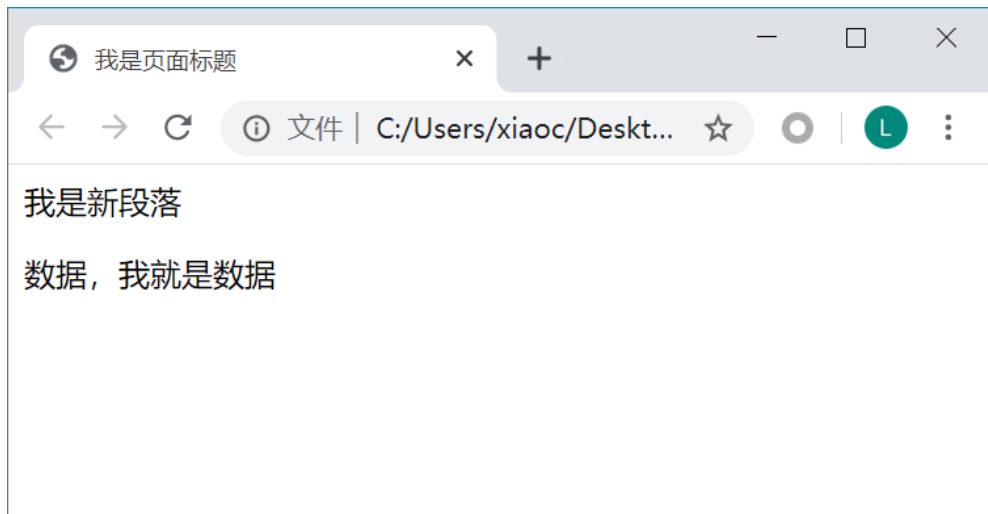
Symbol	Company Name	Price
OK	Nokia Corporation	8.4
APL	Apple Inc.	104.7
AC	Bank of America Cor...	16.6
	AT&T, Inc.	33.5
BR	Petr	12.5
QQ	PowerShares QQQ	97.9
3	Facebook, Inc.	80.0
IO	iBio, Inc.	1.4
ELP	Yelp, Inc.	58.5
IFN	Infinera Corporation	13.0
SFT	Microsoft Corporation	44.7
O	The Coca-Cola Comp...	41.2
PF	Pfizer Inc	28.8

API获取数据

网络数据如何获取（爬取）？

抓取网页，解析网页内容

- 抓取
 - Requests第三方库
 - Scrapy框架
- 解析
 - BeautifulSoup库
 - re模块



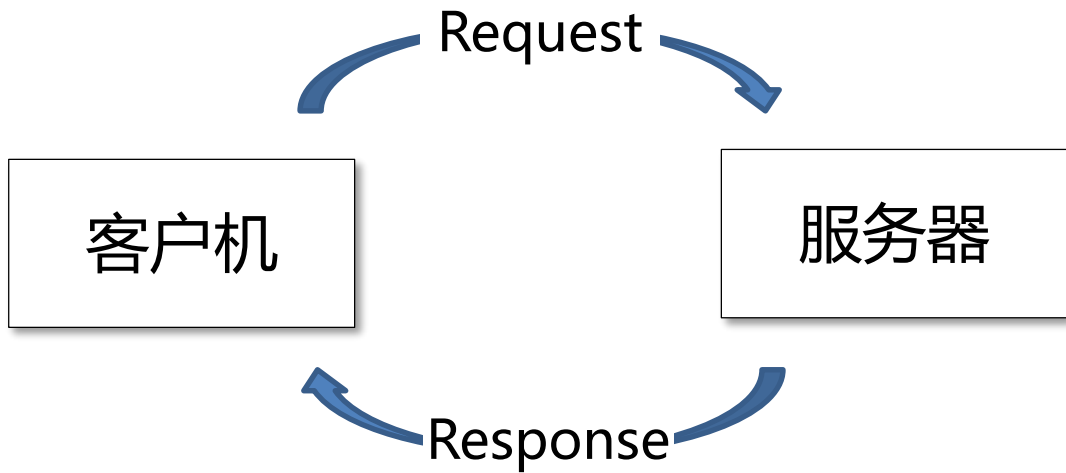
```
<html>
<head>
<title>我是页面标题</title>
</head>

<body>
<p>我是新段落</p>
数据，我就是数据
</body>

</html>
```

参考：

http://www.w3school.com.cn/html/html_backgrounds.asp

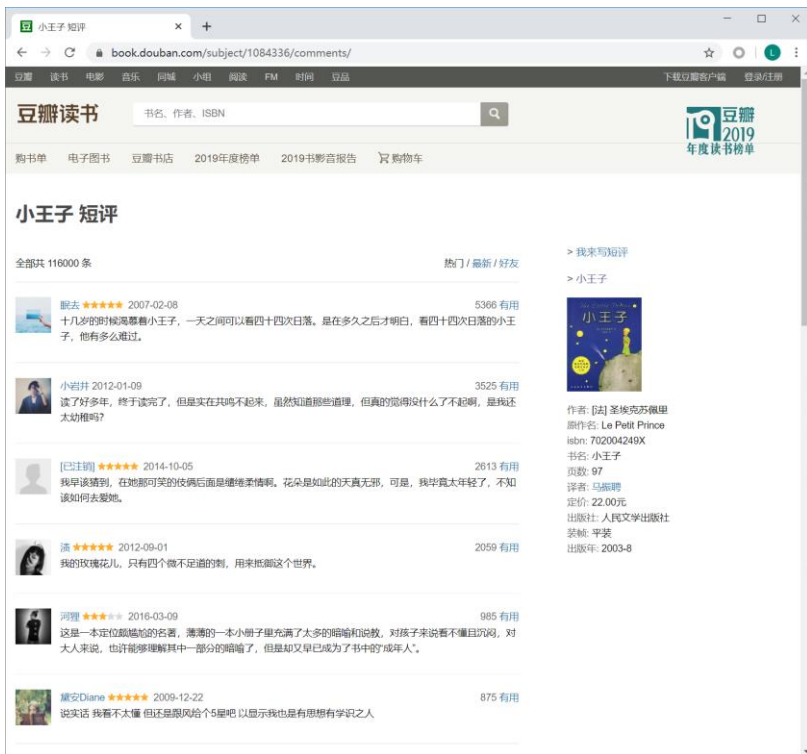


Requests库

- Requests库是更简单、方便和人性化的Python HTTP第三方库
- Requests官网：
<http://www.python-requests.org/>

\$ pip install requests
(Anaconda中预装)





豆瓣读书 《小王子》短评

requests.get()

请求获取指定URL位置的资源，对应HTTP协议的GET方法，返回一个Response对象



```
>>> import requests
```

```
>>> r = requests.get('https://www.nju.edu.cn')
```

```
>>> r.status_code
```

```
200
```

```
>>> print(r.text)
```

```
>>> r = requests.get('https://book.douban.com/subject/1084336/comments/')
```

```
>>> r.status_code
```

```
418
```




```
>>> headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac  
OS X 10_14_0) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/71.0.3578.98 Safari/537.36'}  
>>> r = requests.get(url, headers = headers) #定制请求头  
>>> payload = {'key1': 'value1', 'key2': 'value2'}  
>>> r = requests.get('http://httpbin.org/get', params = payload)  
>>> r = requests.post(url, data = payload)
```

```
>>> r.encoding      # 根据HTTP头部自动推测
```

```
'UTF-8'
```

```
>>> r.encoding = 'gb2312'
```

```
>>> r.encoding = r.apparent_encoding
```

```
>>> r.content      # 以字节方式访问Response对象
```

```
>>> r.json()
```

r.content

以字节方式访问Response对象



```
r = requests.get('http://...sample.jpg')  
with open('pic.jpg', 'wb') as f:  
    f.write(r.content)
```

- JSON格式

- JavaScript Object Notation, JS对象标记)
- 一种轻量级的数据交换格式

r.json()

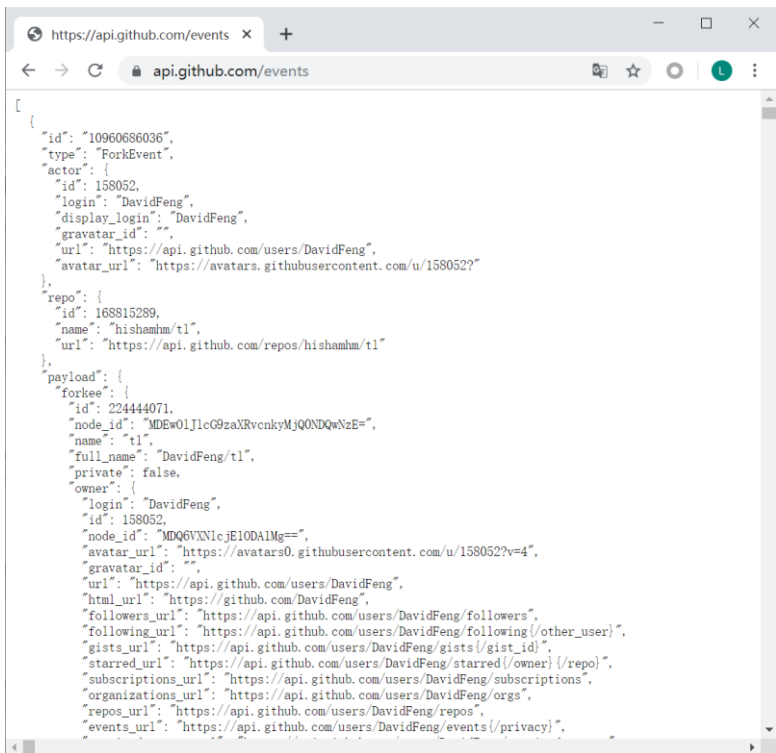
```
'{"name":"Niuyun", "address":{"city": "Beijing", "street":  
"Chaoyang Road"}}'
```

解析后

```
>>> x = {"name":"Niuyun",  
         "address":{"city":"Beijing","street":"Chaoyang Road"}}  
>>> x['address']['street']  
'Chaoyang Road'
```

JSON文件

13



```
[
  {
    "id": "10960686036",
    "type": "ForkEvent",
    "actor": {
      "id": 158052,
      "login": "DavidFeng",
      "display_login": "DavidFeng",
      "gravatar_id": "",
      "url": "https://api.github.com/users/DavidFeng",
      "avatar_url": "https://avatars.githubusercontent.com/u/158052?"
    },
    "repo": {
      "id": 168815289,
      "name": "hishamhm/t1",
      "url": "https://api.github.com/repos/hishamhm/t1"
    },
    "payload": {
      "forkee": {
        "id": 224444071,
        "node_id": "MDEwO1JlcG9zaXRvcnkyMjQ0NDQwNzE=",
        "name": "t1",
        "full_name": "DavidFeng/t1",
        "private": false,
        "owner": {
          "login": "DavidFeng",
          "id": 158052,
          "node_id": "MDQ6VXNlcjE1ODAlMg==",
          "avatar_url": "https://avatars0.githubusercontent.com/u/158052?v=4",
          "gravatar_id": "",
          "url": "https://api.github.com/users/DavidFeng",
          "html_url": "https://github.com/DavidFeng",
          "followers_url": "https://api.github.com/users/DavidFeng/followers",
          "following_url": "https://api.github.com/users/DavidFeng/following{/other_user}",
          "gists_url": "https://api.github.com/users/DavidFeng/gists{/gist_id}",
          "starred_url": "https://api.github.com/users/DavidFeng/starred{/owner}{/repo}",
          "subscriptions_url": "https://api.github.com/users/DavidFeng/subscriptions",
          "organizations_url": "https://api.github.com/users/DavidFeng/orgs",
          "repos_url": "https://api.github.com/users/DavidFeng/repos",
          "events_url": "https://api.github.com/users/DavidFeng/events{/privacy}",

```

```
>>> r =
requests.get('https://ap
i.github.com/events')
>>> data = r.json()
>>> data[0]['id']
```

- Robots协议也称为爬虫协议，全称为爬虫排除协议 (The Robots Exclusion Protocol)
- 检查站点根目录下是否存在robots.txt

```
User-agent: *  
Disallow: /subject_search  
Disallow: /amazon_search  
Disallow: /search  
...  
Disallow: /link2/  
Disallow: /recommend/  
Disallow: /doubanapp/card  
Disallow: /update/topic/  
Allow: /ads.txt  
Sitemap: https://www.douban.com/sitemap_index.xml  
Sitemap: https://www.douban.com/sitemap_updated_index.xml  
# Crawl-delay: 5
```

```
User-agent: Wandoujia Spider  
Disallow: /  
...
```

网页数据解析

- **Beautiful Soup**是一个可以从HTML或XML文件中提取数据的Python库
- 官方网站：
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>



Beautiful Soup

- \$ pip install beautifulsoup4 (Anaconda中预装)

```
>>> import requests
```

```
>>> from bs4 import BeautifulSoup
```

```
>>> r = requests.get('https://book.douban.com/subject/1084336/comments/',  
headers = headers)
```

```
>>> soup = BeautifulSoup(r.text, 'lxml')
```

lxml: HTML解析器
\$ pip install lxml

Python内置的HTML解析器
BeautifulSoup(markup, 'html.parser')

Beautiful Soup

- BeautifulSoup对象
 - Tag

```
>>> markup = '<p class="title"><b>The Little Prince</b></p>'
>>> soup = BeautifulSoup(markup, 'lxml')
>>> soup.b
<b>The Little Prince</b>
```
 - NavigableString
 - BeautifulSoup
 - Comment

标签内容访问方式
BeautifulSoup对象.Tag

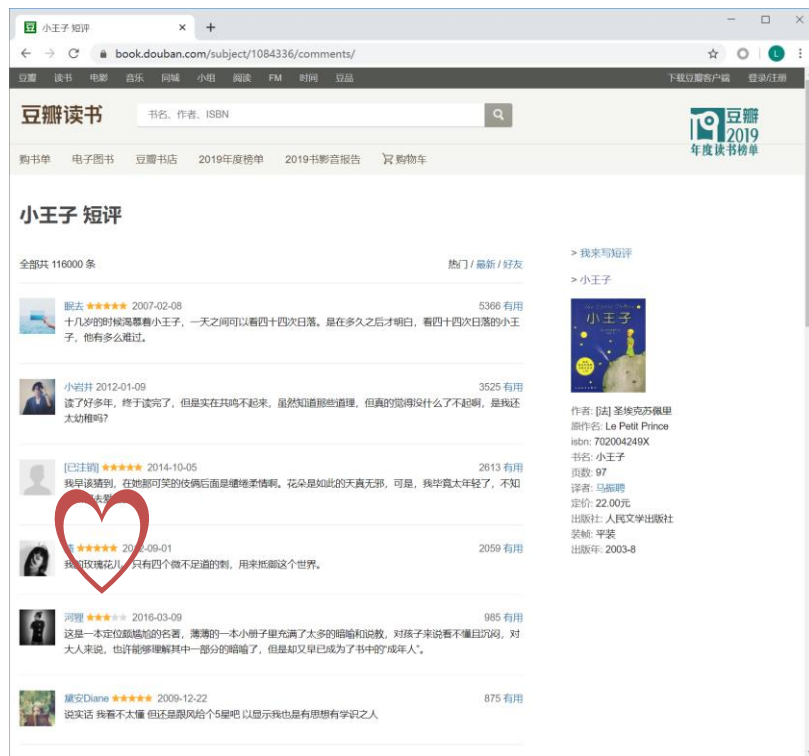
Beautiful Soup

```
>>> markup = '<p class="title"><b>The Little Prince</b></p>'
>>> soup = BeautifulSoup(markup, 'xml')
>>> tag = soup.p
>>> tag
<p class="title"><b>The Little Prince</b></p>
>>> tag.name
'p'
>>> tag['class']
['title']
>>> tag.attrs
{'class': ['title']}
>>> tag.string
'The Little Prince'
>>> soup.find_all('b')
[<b>The Little Prince</b>]
```

``不知道第几次重读。每过一段时间再读，都有新的收获。心变得很柔软，脑里的迷雾被驱散。更多的关注他人，关心这个世界，自私是多么无趣的事情啊。我想，写一本能温暖人心，帮助困难的人们的书，比世界上很多事情都有意义。``

```
pattern = soup.find_all('span', {'class':'short'})
for item in pattern:
    print(item.string)
```

```
r = requests.get('https://book.douban.com/subject/1084336/comments/',  
headers=headers)  
  
soup = BeautifulSoup(r.text, 'lxml')  
  
pattern = soup.find_all('span', {'class':'short'})  
  
for item in pattern:  
    print(item.string)
```



豆瓣读书 《小王子》推荐星级

- 正则表达式是对字符串（包括普通字符和特殊字符）操作的一种逻辑公式
- **re**正则表达式模块进行各类正则表达式处理
- 参考网站：
<https://docs.python.org/3.5/library/re.html>



元字符	描述
.	匹配除换行符外的任意字符
*	重复前面的子表达式0次或多次
+	重复前面的子表达式1次或更多次
?	重复前面的子表达式0次或1次
^	匹配字符串的开始
\$	匹配字符串的结束
{n}	重复n次
{n, }	重复n次或更多次
{n, m}	重复n到m次
\b	匹配单词的开始或结尾即单词边界, “\B” 匹配非单词边界
\d	匹配数字, “\D” 匹配任意非数字字符
\s	匹配任意空白符, “\S” 匹配任意非空白符
\w	匹配任意字母、数字或下划线的标识符字符, “\W” 匹配任意非标识符字符
[a-z]	匹配指定范围内的任意字符
[^a-z]	匹配任何不在指定范围内的任意字符



```
<span class="user-stars  
allstar50 rating" title="力荐  
></span>
```

```
'<span class="user-stars allstar(.*) rating'
```

```
pattern = re.compile('<span class="user-stars allstar(.*) rating')  
p = re.findall(pattern, r.text)
```

```
r = requests.get('https://book.douban.com/subject/1084336/comments', headers = headers)
```

```
soup = BeautifulSoup(r.text, 'lxml')
```

```
pattern = soup.find_all('span', {'class':'short'})
```

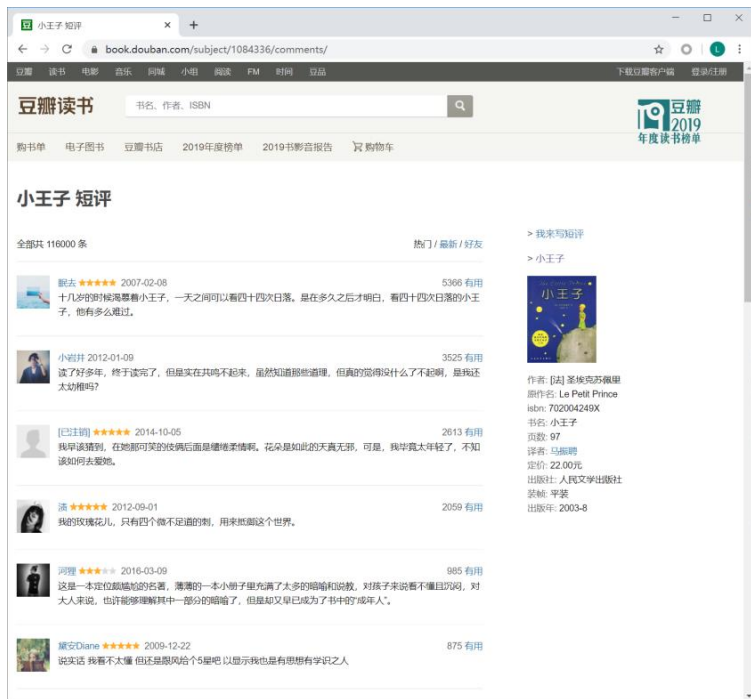
```
for item in pattern:
```

```
    print(item.string)
```

```
pattern_s = re.compile('<span class="user-stars allstar.*?) rating"')
```

```
p = re.findall(pattern_s, r.text)
```

思考：抓取图书短评前5页



<https://book.douban.com/subject/1084336/>

```
r = requests.get('https://book.douban.com/subject/1084336/comments/hot?p=' + str(i+1))
```



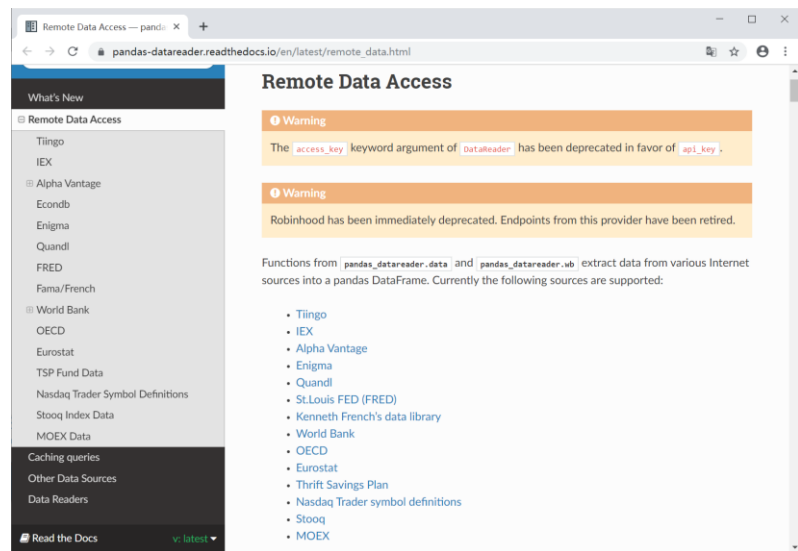
<http://money.cnn.com/data/dow30/>

抓取道指成分股数据并将30家公司的代码、公司名称和最近一次成交价放到一个列表中输出



<http://www.volleyball.world/en/vnl/2018/women/results-and-ranking/round1>
抓取TEAMS and TOTAL, WON, LOST of MATCHES

API 获取数据



Source

```
>>> import pandas_datareader.data as web
>>> f = web.DataReader('AXP', 'stooq')
>>> f.head(5)
```

Date	Open	High	Low	Close	Volume
2019-10-04	112.62	114.530	112.60	114.41	2753195
2019-10-03	112.52	112.955	111.06	112.55	3549232
2019-10-02	115.76	115.810	112.75	112.86	4931560
2019-10-01	118.70	119.500	116.61	116.70	2857528
2019-09-30	119.05	119.240	118.14	118.28	2353731

TuShare 0.4.3 documentation »

Table Of Contents

- 前言
- 致谢
- 使用对象
- 使用前提
- 下载安装
- 版本升级
- 版本信息
- 友情链接
- 交易数据
- 历史行情
- 实时行情
- 实时分笔
- 当日历史分笔
- 大盘指数行情列表
- 大单交易数据
- 股票参考数据
- 分配预案
- 业绩预告
- 限售股解禁
- 基金持股
- 新股数据
- 融资融券 (沪市)
- 融资融券 (深市)
- 股票分类数据
- 行业分类
- 概念分类
- 地域分类
- 中小板分类
- 创业板分类
- 风险提示板分类
- 沪深300成份及权重
- 上证50成份股
- 中证500成份股
- 停止上市股票列表
- 暂停上市股票列表
- 基本面数据

前言

TuShare是一个免费、开源的python财经数据接口包。主要实现对股票等金融数据从数据采集、清洗加工 到 数据存储的过程，能够为金融分析人员提供快速、整洁、和多样的便于分析的数据，为他们在数据获取方面极大地减轻工作量，使他们更加专注于策略和模型的研究与实现上。考虑到Python pandas包在金融量化分析中体现出的优势，TuShare返回的绝大部分的数据格式都是pandas DataFrame类型，非常便于用pandas/NumPy/Matplotlib进行数据分析和可视化。当然，如果您习惯了用Excel或者关系型数据库做分析，您也可以通过TuShare的数据存储功能，将数据全部保存到本地后进行分析。应一些用户的需求，从0.2.5版本开始，TuShare同时兼容Python 2.x和Python 3.x，对部分代码进行了重构，并优化了一些算法，确保数据获取的高效和稳定。

TuShare从发布到现在，已经帮助很多用户在数据方面降低了工作压力，同时也得到很多用户的反馈，TuShare将一如既往的免费和开源的形式分享出来，希望对有需求的人带来一些帮助。如果您觉得TuShare好用并有所收获，请通过微博、微信或者网站博客的方式分享出去，让更多的人了解和使用它，使它能在大家的使用过程中逐步得到改进和提升。TuShare还在不断的完善和优化，后期将逐步增加港股、期货、外汇和基金方面的数据，所以，您的支持和肯定才是TuShare坚持下去的动力。

TuShare的数据主要来源于网络，如果在使用过程中碰到数据无法获取或发主数据错误的情况请联系我，如果有什么好的建议和意见，也请及时联系我，在此谢过。如果在pandas/NumPy技术上有问题，欢迎加入“pandas数据分析”QQ群：297882961（已满），TuShare用户群：658562506，我会和大家一起帮忙为您解答。另外，请扫码关注“挖地兔”的微信公众号，定期会发布TuShare的最新动态及有价值的金融数据分析与处理方面的教程和文章。



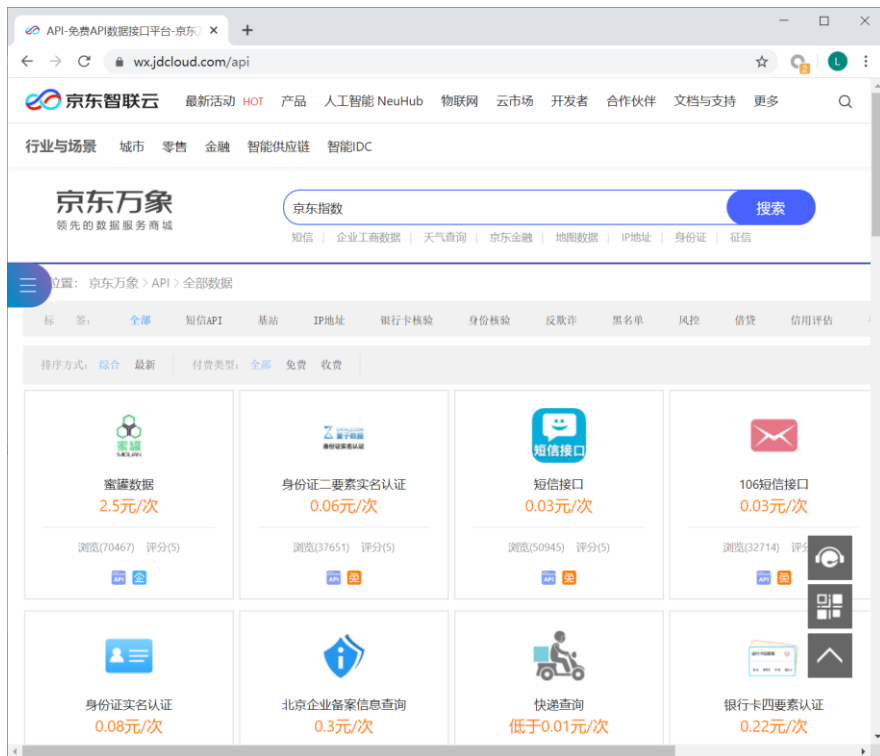
从最新本开始，tushare将接受第三方数据的接入，欢迎供应商通过微信公众号“挖地兔”与我联系。

TUSHARE 功能概览



京东万象API

34



<https://wx.jdcloud.com/api>

网络爬虫案例



新浪滚动新闻标题热点挖掘

37

新闻中心滚动新闻_新浪网

news.sina.com.cn/roll/#pageid=153&lid=2509&k=&num=50&page=1

sina 新闻中心 新浪新闻 | 新浪首页 | 新浪导航

2020年5月4日 全部滚动新闻

输入关键字，多关键字以空格分隔 按标题 全部时间 搜索

28 秒后刷新 刷新

栏目	标题	时间
[全部]	即使被它淹没也不会窒息，这是什么神奇液体？	05-04 14:17
[全部]	江南布衣跌逾10% 主动卖盘66%	05-04 14:14
[全部]	骗局多？是玄学？已实现霸权？带你走进量子的真相	05-04 14:14
[全部]	易易喜升逾40% 创52周新高	05-04 14:11
[全部]	科比亮相乔丹纪录片：没有他我赢不了5个总冠军	05-04 14:08
[全部]	"美国队长"也拯救不了的Apple TV +，苹果该弃了吗？	05-04 14:07
[全部]	河北新乐一男子持械行凶，造成1人死亡11人受伤	05-04 14:06
[全部]	澳新两国将于5日商讨商业航线恢复计划	05-04 14:04
[全部]	疫情之下油价持续疲软，专家却暗示牛市已在酝酿中！	05-04 14:01
[全部]	高盛“耐心看多” 油价 受基本面改善迹象鼓舞	05-04 13:58
[全部]	招行跌逾5% 主动卖盘44%	05-04 13:57
[全部]	欧市盘前：多重利空因素施压英镑 美布两油大幅分化	05-04 13:57
[全部]	台湾大批失业者改送外卖 人员激增致薪资砍半(视频)	05-04 13:57
[全部]	美国银行:投资基金流向显示全球股市涨势曲线趋于平缓	05-04 13:56
[全部]	加拿大央行下任总裁拍板定案 危机克星麦克勒姆出线	05-04 13:55

全部 国内 国际 社会 体育 娱乐 军事 科技 财经 股市 美股

往日回顾 自动隐藏



结巴分词和词云

- **结巴分词器**

`$ pip install -i https://pypi.tuna.tsinghua.edu.cn/simple jieba`

- **词云包**

`$ pip install -i https://pypi.tuna.tsinghua.edu.cn/simple wordcloud`

或 `$ conda install -c conda-forge wordcloud` （推荐）

补充：JSON模块与JSON格式文件存取

```
import json
data = {
    'name' : 'xiaohua',
    'age' : 18,
    'id' : 11121
}
json_str = json.dumps(data)
data = json.loads(json_str)
```



```
with open('data.json', 'w') as f:  
    json.dump(data, f)
```

```
with open('data.json', 'r') as f:  
    data = json.load(f)
```

```
n01001 = json.load(open('01001.json'))
```

```
>>> import json
```

```
>>> n01001 = json.load(open('01001.json'))
```

```
path = ...
```

```
files = os.listdir(path)
```

```
for file in files:
```

```
    file_name = path + file
```

```
    data = json.load(open(file_name))
```