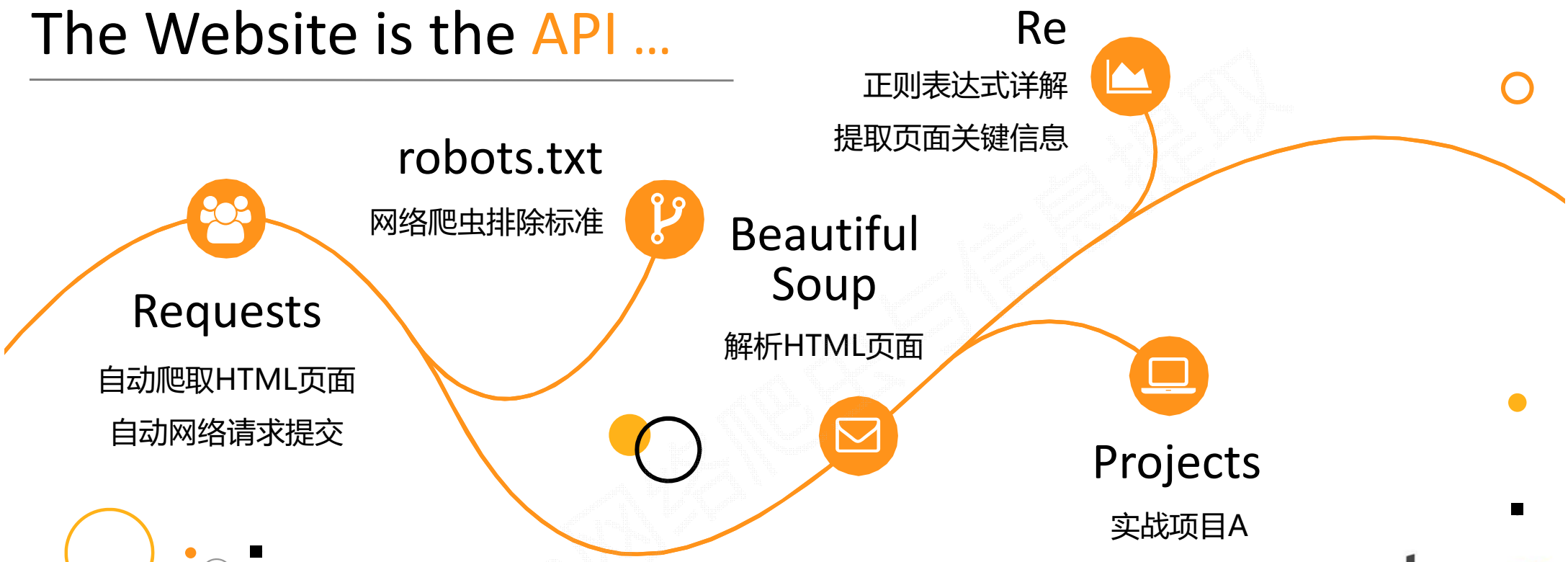
实例3：股票数据定向爬虫

WS09

嵩天

www.python123.org

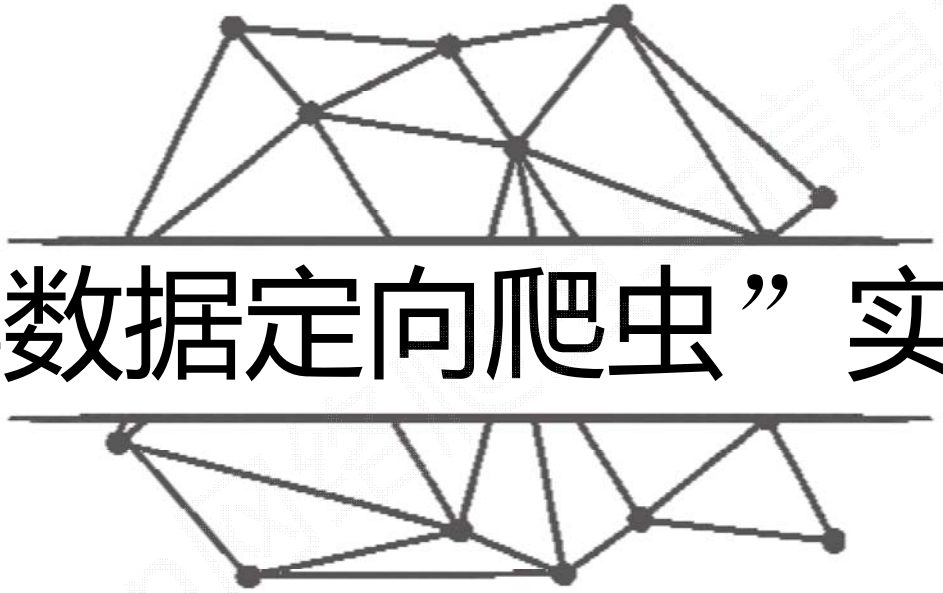


掌握定向网络数据爬取和网页解析的基本能力

python
弹指之间 · 享受创新

Python网络爬虫与信息提取

O4X -Tian



“股票数据定向爬虫”实例介绍

功能描述

目标：获取上交所和深交所所有股票的名称和交易信息

输出：保存到文件中

技术路线：`requests-bs4-re`

候选数据网站的选择

新浪股票：http://finance.sina.com.cn/stock/

百度股票：https://gupiao.baidu.com/stock/

选取原则：股票信息静态存在于HTML页面中，非js代码生成

没有Robots协议限制

选取方法：浏览器 F12，源代码查看等

选取心态：不要纠结于某个网站，多找信息源尝试

候选数据网站的选择

新浪股票：http://finance.sina.com.cn/stock/

百度股票：https://gupiao.baidu.com/stock/

请查看视频理解网站的选取过程

数据网站的确定

获取股票列表：

东方财富网：http://quote.eastmoney.com/stocklist.html

获取个股信息：

百度股票：https://gupiao.baidu.com/stock/

单个股票：https://gupiao.baidu.com/stock/sz002439.html

程序的结构设计

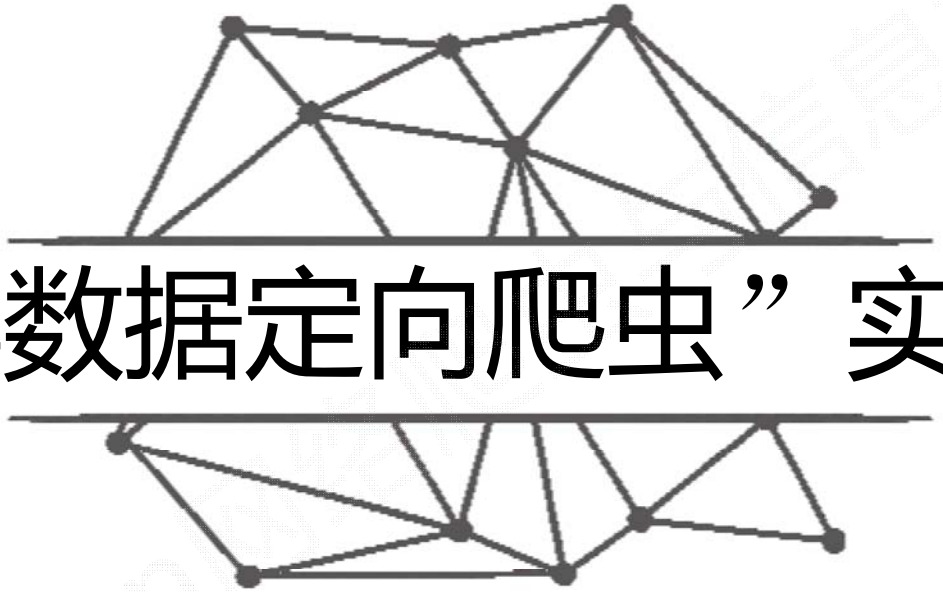
步骤1：从东方财富网获取股票列表

步骤2：根据股票列表逐个到百度股票获取个股信息

步骤3：将结果存储到文件

```
<div class="bets-content">
                                <div class="line1">
    <dl><dt>今开</dt><dd class="s-up">21.38</dd></dl>
    <dl><dt>成交量</dt><dd>11.16万手</dd></dl>
    <dl><dt>最高</dt><dd class="s-up">21.99</dd></dl>
    <dl><dt>涨停</dt><dd class="s-up">23.53</dd></dl>
    <dl><dt>内盘</dt><dd>5.16万手</dd></dl>
    <dl><dt>成交额</dt><dd>2.41亿</dd></dl>
    <dl><dt>委比</dt><dd>-31.68%</dd></dl>
    <dl><dt>流通市值</dt><dd>122.79亿</dd></dl>
    <dl><dt class="mt-1">市盈率<sup>MRQ</sup></dt><dd>313.49</dd></dl>
    <dl><dt>每股收益</dt><dd>0.05</dd></dl>
    <dl><dt>总股本</dt><dd>8.97亿</dd></dl>
    <div class="clear"></div>
</div>
<div class="line2">
    <dl><dt>昨收</dt><dd>21.39</dd></dl>
    <dl><dt>换手率</dt><dd>2.00%</dd></dl>
    <dl><dt>最低</dt><dd class="s-down">21.20</dd>
    </dl>
    <dl><dt>跌停</dt><dd class="s-down">
        19.25</dd></dl>
    <dl><dt>外盘</dt><dd>6.00万手</dd></dl>
    <dl><dt>振幅</dt><dd>3.69%</dd></dl>
    <dl><dt>量比</dt><dd>0.55</dd></dl>
    <dl><dt>总市值</dt><dd>197.00亿</dd></dl>
    <dl><dt>市净率</dt><dd>9.52</dd></dl>
    <dl><dt>每股净资产</dt><dd>2.31</dd></dl>
    <dl><dt>流通股本</dt><dd>5.59亿</dd></dl>
</div>
                            <div class="clear"></div>
```

个股信息
采用键值对维护



“股票数据定向爬虫”实例编写

main()

```
import requests
from bs4 import BeautifulSoup
import traceback
import re

def getHTMLText(url):
    return ""

def getStockList(lst, stockURL):
    return ""

def getStockInfo(lst, stockURL, fpath):
    return ""

def main():
    stock_list_url = 'http://quote.eastmoney.com/stocklist.html'
    stock_info_url = 'https://gupiao.baidu.com/stock/'
    output_file = 'D://BaiduStockInfo.txt'
    slist=[]
    getStockList(slist, stock_list_url)
    getStockInfo(slist, stock_info_url, output_file)

main()
```

getHTMLText()

```
def getHTMLText(url):
    try:
        r = requests.get(url)
        r.raise_for_status()
        r.encoding = r.apparent_encoding
        return r.text
    except:
        return ""
```

getStockList()

东方财富网：http://quote.eastmoney.com/stocklist.html

```
<li><a target="_blank" href="http://quote.eastmoney.com/sh502020.html">国金50(502020)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502021.html">国金50A(502021)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502022.html">国金50B(502022)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502023.html">钢铁分级(502023)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502024.html">钢铁A(502024)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502025.html">钢铁B(502025)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502026.html">新丝路(502026)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502027.html">新丝路A(502027)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502028.html">新丝路B(502028)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502031.html">高铁A(502031)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502032.html">高铁B(502032)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502036.html">互联金融(502036)</a></li>
<li><a target="_blank" href="http://quote.eastmoney.com/sh502037.html">网金A(502037)</a></li>
```

getStockList()

```
def getStockList(lst, stockURL):
    html = getHTMLText(stockURL)
    soup = BeautifulSoup(html, 'html.parser')
    a = soup.find_all('a')
    for i in a:
        try:
            href = i.attrs['href']
            lst.append(re.findall(r"[s][hz]\d{6}", href)[0])
        except:
            continue
```

getStockInfo()

```
<div class="stock-info" data-spm="2">
    <div class="stock-bets">
        <h1>
            <a class="bets-name" href="/stock/sz002439.html">
            启明星辰 (<span>002439</span>)
            </a>
            <span class="state f-up">已休市 2017-03-03  15:00:03
            </span>
        </h1>
        <div class="price s-up ">
                    <strong  class="_close">21.97</strong>
            <span>+0.58</span>
            <span>+2.71%</span>
                    </div>
        <div class="bets-content">

                                        <div class="line1">
                    <dl><dt>今开</dt><dd class="s-up">21.38</dd></dl>
                    <dl><dt>成交量</dt><dd>11.16万手</dd></dl>
                    <dl><dt>最高</dt><dd class="s-up">21.99</dd></dl>
                    <dl><dt>涨停</dt><dd class="s-up">23.53</dd></dl>
                    <dl><dt>内盘</dt><dd>5.16万手</dd></dl>
                    <dl><dt>成交额</dt><dd>2.41亿</dd></dl>
                    <dl><dt>委比</dt><dd>-31.68%</dd></dl>
                    <dl><dt>流通市值</dt><dd>122.79亿</dd></dl>
                    <dl><dt class="mt-1">市盈率<sup>MRQ</sup></dt><dd>313.49</dd></dl>
                    <dl><dt>每股收益</dt><dd>0.05</dd></dl>
                    <dl><dt>总股本</dt><dd>8.97亿</dd></dl>
                    <div class="clear"></div>
                </div>
                <div class="line2">
                    <dl><dt>昨收</dt><dd>21.39</dd></dl>
                    <dl><dt>换手率</dt><dd>2.00%</dd></dl>
                    <dl><dt>最低</dt><dd class="s-down">21.20</dd>
                    </dl>
                    <dl><dt>跌停</dt><dd class="s-down">
                        19.25</dd></dl>
                    <dl><dt>外盘</dt><dd>6.00万手</dd></dl>
                    <dl><dt>振幅</dt><dd>3.69%</dd></dl>
                    <dl><dt>量比</dt><dd>0.55</dd></dl>
                    <dl><dt>总市值</dt><dd>197.00亿</dd></dl>
                    <dl><dt>市净率</dt><dd>9.52</dd></dl>
                    <dl><dt>每股净资产</dt><dd>2.31</dd></dl>
                    <dl><dt>流通股本</dt><dd>5.59亿</dd></dl>
                </div>
                                        <div class="clear"></div>
        </div>
    </div>
    <ul class="stock-add">
        <li data-spm="1"><button class="">+ 加自选</button></li>
        <li class="hint invisible"><span class="add-stock-count">0</span>人关注该股票</li>
    </ul>
    </div>
```

百度股票：

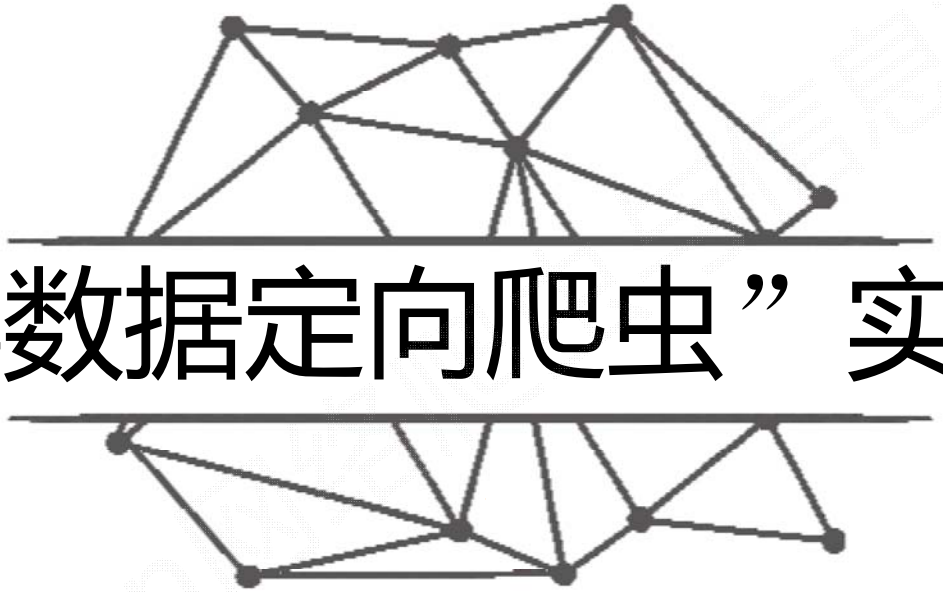
https://gupiao.baidu.com/stock/

getStockInfo()

```
def getStockInfo(lst, stockURL, fpath):
    for stock in lst:
        url = stockURL + stock + ".html"
        html = getHTMLText(url)
        try:
            if html=="":
                continue
            infoDict = {}
            soup = BeautifulSoup(html, 'html.parser')
            stockInfo = soup.find('div',attrs={'class':'stock-bets'})

            name = stockInfo.find_all(attrs={'class':'bets-name'})[0]
            infoDict.update({'股票名称': name.text.split()[0]})

            keyList = stockInfo.find_all('dt')
            valueList = stockInfo.find_all('dd')
            for i in range(len(keyList)):
                key = keyList[i].text
                val = valueList[i].text
                infoDict[key] = val

            with open(fpath, 'a', encoding='utf-8') as f:
                f.write( str(infoDict) + '\n' )
        except:
            traceback.print_exc()
            continue
```

全代码

请阅读全代码



“股票数据定向爬虫”实例优化

如何提高用户体验？

速度提高：编码识别的优化

```
def getHTMLText(url):
    try:
        r = requests.get(url)
        r.raise_for_status()
        r.encoding = r.apparent_encoding
        return r.text
    except:
        return ""
```

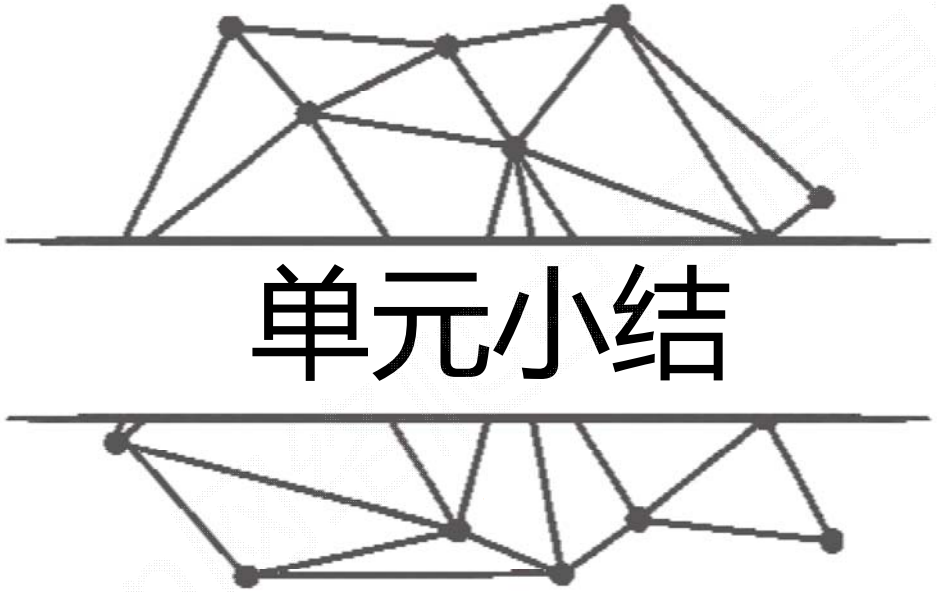
r.apparent_encoding需要分析文本，运行较慢，可辅助人工分析

速度提高：编码识别的优化

```
def getHTMLText(url, code="utf-8"):
    try:
        r = requests.get(url)
        r.raise_for_status()
        r.encoding = code
        return r.text
    except:
        return ""

def getStockList(lst, stockURL):
    html = getHTMLText(stockURL, "GB2312")
    soup = BeautifulSoup(html, 'html.parser')
    a = soup.find_all('a')
    for i in a:
        try:
            href = i.attrs['href']
            lst.append(re.findall(r"[s][hz]\d{6}", href)[0])
        except:
            continue
```

体验提高：增加动态进度显示

```
def getStockInfo(lst, stockURL, fpath):
    count = 0


        with open(fpath, 'a', encoding='utf-8') as f:
            f.write( str(infoDict) + '\n' )
            count = count + 1
            print('\r当前进度: {:.2f}%".format(count*100/len(lst)),end="")
    except:
        count = count + 1
        print("\r当前进度: {:.2f}%".format(count*100/len(lst)),end="")
        continue
```



单元小结

实例3：股票数据定向爬虫

采用requests-bs4-re路线实现了股票信息爬取和存储

实现了展示爬取进程的动态滚动条