

通过编程获取历史股价的方式很多，但是免费的方式可以简单地分为四类：

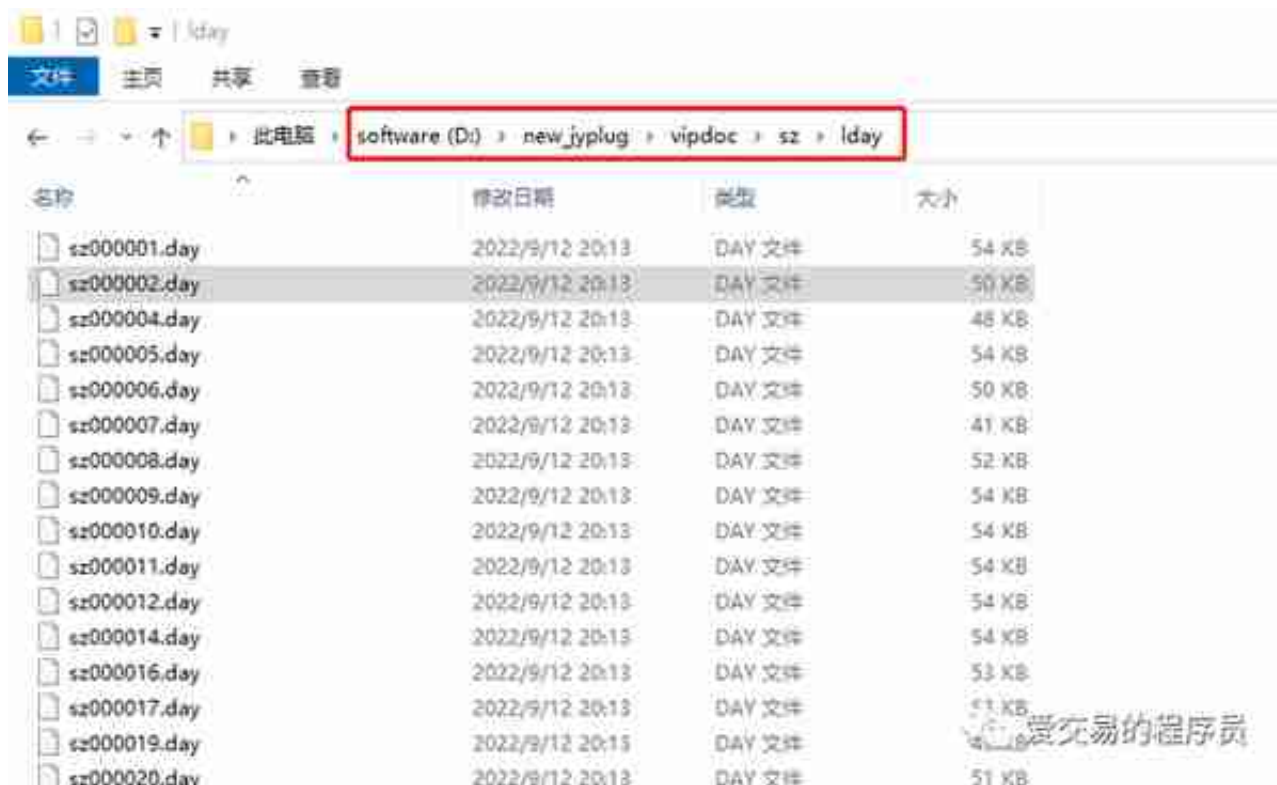
1. 使用某种网友开发的插件：比如Python就有TuShare, akshare, 这种只需要写几行代码就能获取，但是不同的库有不同的利弊。比如TuShare中有很多数据需要一定的积分才能获取。
2. 通过API接口，比如新浪，腾讯等等。
3. 从专门的行情软件上面提取数据。
4. 从财经网站上面爬取行情数据。

## 使用TuShare获取股价

TuShare是非常方便使用的，比如在官网拿到Token后，一行调用就可以获取股价信息：

```
import tushare as ts
ts.set_token('***')
pro = ts.pro_api()
price = ts.pro_bar(ts_code='000020.SZ', start_date='2022-09-01',
                  end_date='2022-09-09')
price
```

结果如图：



可以使用程序读出来。附完整解析代码:

```
import struct
import datetime
class stock_price:
    def __init__(self, stock_date, stock_open, stock_high, stock_low, stock_close, stock_amount, stock_vol):
        self.stock_date = stock_date
        self.stock_open = stock_open
        self.stock_high = stock_high
        self.stock_low = stock_low
        self.stock_close = stock_close
        self.stock_amount = stock_amount
        self.stock_vol = stock_vol
    def get_stock_price(filepath):
        data = []
        with open(filepath, 'rb') as f:
            while True:
                stock_date = f.read(4)
                stock_open = f.read(4)
                stock_high = f.read(4)
                stock_low = f.read(4)
                stock_close = f.read(4)
                stock_amount = f.read(4)
                stock_vol = f.read(4)
                stock_reservation = f.read(4)
                if not stock_date:
                    break
                stock_date = struct.unpack("l", stock_date)
                # 4?? ?20220909
                stock_open1 = struct.unpack("l", stock_open)
                #???*100
                stock_high1 = struct.unpack("l", stock_high)
                #???*100
                stock_low = struct.unpack("l", stock_low)
                #???*100
                stock_close = struct.unpack("l", stock_close)
                #???*100
                stock_amount = struct.unpack("f", stock_amount)
                #??
                stock_vol = struct.unpack("f", stock_vol)
                # ???
                stock_reservation = struct.unpack("f", stock_reservation)
                #???
                date_format = datetime.datetime.strptime(str(stock_date[0]), '%Y%M%d')
                #?????
                data.append(stock_price(date_format.strftime('%Y-%M-%d'),
                    stock_open1[0]/100.0,
                    stock_high1[0]/100.0,
                    stock_low[0]/100.0,
                    stock_close[0]/100.0,
                    stock_amount[0]/100.0,
                    stock_vol[0]/100.0))
            return data
```

找一个文件试试:

```
data_list = get_stock_price('./sz000002.day')
print(f'?????: {data_list[-1:][0].stock_date}')
print(f'?????: {data_list[-1:][0].stock_open}')
print(f'?????: {data_list[-1:][0].stock_high}')
print(f'?????: {data_list[-1:][0].stock_low}')
```

```
{ data_list[-1:][0].stock_close}')
```

???

????: 2022-09-09????: 17.49????: 18.39????: 17.41????: 18.15

在同花顺里面验证了下价格信息，是正确的。不过这种方式的弊端在于，需要手工在通达信里面执行更新数据的操作，并且通达信在未来某天变更了读写文件的方式，这种方式就不能用了。其它的软件也是类似的。这里要注意的是，如果需要读取指数的信息，

```
struct.unpack("l", stock_open)
```

这行代码里面的格式需要变动一下，目前变成

```
struct.unpack("f", stock_open)
```

即可。

## 从财经网站上爬取行情数据

这种是最靠谱的方式的，但是实现也是比较繁琐的，某种程度上，也有法律上面的风险，参考:中国爬虫违法违规案例汇总 ([https://github.com/HiddenStrawberry/Crawler\\_Illegal\\_Cases\\_In\\_China](https://github.com/HiddenStrawberry/Crawler_Illegal_Cases_In_China))。股票价格数据都是公开的，不是什么隐私，但是程序写的不好，把目标网站爬宕机了，就有法律风险了。上面例子中的新浪API接口还是很稳定，支持高并发的。貌似可能是某个性能高手开发的, 参考: 杨建：网站加速--服务器编写篇（上）(<https://blog.csdn.net/lishenglong666/article/details/39028093>), 原文不见了，放个转载。

## 提供股价信息的部分网站

1. 雪球, <https://xueqiu.com>
2. 集思录, <https://www.jisilu.cn/>
3. 巨潮资讯网, <http://www.cninfo.com.cn>
4. 百度股市通, <https://gushitong.baidu.com>
5. 东方财富网, <https://www.eastmoney.com/>

6. 同花顺, <https://www.10jqka.com.cn/>

.....