



项目3

采集巨潮资讯网的股票财经信息



项目目标

- 掌握网站 XHR 请求的地址获取方法。
- json 解析进阶，掌握复杂情况下的解析与存储方法。
- 掌握简单的数据清洗方式。

项目 2 中，我们通过一个简单的案例，对开放 API 的数据获取进行了初步的学习。本项目将主要学习在网站没有开放 API 的情况下，如何通过 Chrome 浏览器的数据包抓取工具来分析网站的 XHR 请求，从而获取接口地址。



项目描述

假设读者在一家制造业企业工作，年终时领导要对行业进行分析，并将分析结果写入工作报告中。以往的解决方式都是通过手工下载，并在 Excel 表格中进行复制粘贴。这种方式不仅效率低，并且数据采集不完整，无法做到一次性采集多家数据且进行合并。本项目要求读者基于数据采集技术，通过编写程序完成此任务，并且实现代码的可复用，减少重复性的工作。



项目实施

- (1) 通过 Chrome 浏览器的开发者工具的 Network 工具对数据包进行抓取。
- (2) 分析 HTTP 请求中 request（请求）和 response（响应）对象，抓取 XHR（XML-HttpRequest）请求。
- (3) 对返回的 json 格式数据进行解析，将 json 转换为 Python 的字典进行处理。



“1+X” 证书考点

数据采集职业技能等级要求（初级）：

- 熟悉不同互联网应用数据类型。
- 能够使用工具或编写程序获取不同类型互联网数据并进行数据抽取。

岗位技能要求

- 岗位：数据采集工程师。
- 要求：熟练使用 Python 语言编写数据采集程序，熟悉 requests 库，会通过分析 XHR 请求获取真实的请求地址，并采集 XML/json 数据。

课程思政要求

本项目是对财经信息进行采集，需要学生有较强的分析能力。在教学中要把马克思主义立场、观点、方法的教育与科学精神的培养结合起来，提高学生正确认识问题、分析问题和解决问题的能力。

知识链接

1. XHR 的概念

XHR 可以解释为可扩展超文本传输请求，其对象可以在不向服务器提交整个页面的情况下，实现局部更新网页。XHR 的对象用于客户端和服务端之间的异步通信。

2. 将 json 数据转换为 DataFrame

在项目 2 中，我们已经对 json 数据如何解析有了初步的了解。本项目将学习如何通过 pandas 库将解析完成的数据转换为 DataFrame。DataFrame 是 pandas 中一种类似表格的数据结构，是和 Excel 中的表格类似的二维表。接下来以一个示例，讲解什么样的数据可以转换为 DataFrame 的类型。

打开 jupyter notebook，新建一个文件，重命名为 jsonDemo，输入如下代码。

```
In [2]:
import json # 导入 json 解析库
s = '[{"name": "张三", "age": 18}, {"name": "李四", "age": 20}]'
data = json.loads(s) # 将 json 字符串解析为 Python 对象
data
Out [2]:
[{'name': '张三', 'age': 18}, {'name': '李四', 'age': 20}]
```

对以上代码逐行解析如下。

- `import json`

表示导入 Python 的 json 解析库，用于把 json 格式的字符串解析为 Python 对象。

- `s = '[{"name": "张三", "age": 18}, {"name": "李四", "age": 20}]'`

变量 `s` 是一个字符串，其中包含两组结构类型的数据，一种使用 `[]` 包含的数据称为数组（array），另一种用 `{}` 包含的数据称为对象（object）。所有的字符串类型数据，外面均为“双撇号”。json 格式的几个关键特征是字符串格式、对象中的字符必须用双撇号包围、数组与对象相互嵌套。

- `data = json.loads(s)`

使用 json 库的 `loads()` 方法，将 json 字符串转换为 Python 对象。

- `data`

输出 `data` 数据。

json 中的数组转换成了 Python 中的列表（list），json 中的对象转换成了 Python 中的字典（dict）。

索引数据时，列表用下标索引，字典用键索引。例如，获取索引下标为 0 的对象，代码如下。

```
In [3]:
# 获取列表中的第一组数据，下标为 0
data[0]
Out [3]:
{'name': '张三', 'age': 18}
```

获取索引下标为 0 对象的 `name` 键的值，代码如下。

```
In [5]:
# 获取第一组数据中 'name' 键的值
data[0]['name']
Out [5]:
'张三'
```

导入 Pandas 库，将数据转换成 DataFrame，代码如下。

```
In [7]:
import pandas as pd
pd.DataFrame(data)
Out [7]:
   name age
0  张三  18
1  李四  20
```



创建 `DataFrame` 的方法为 `pd.DataFrame()`，该方法可以将以下两种 Python 对象直接转换为 `DataFrame` 对象。

方法 1：变量 `data` 的最外层是列表，列表中有多个键相同的字典。

```
In [1]:
# 列表中多个字典
data = [{'name': '张三', 'age': 18}, {'name': '李四', 'age': 20}]
pd.DataFrame(data)

Out[1]:
   name age
0 张三  18
1 李四  20
```

方法 2：变量 `data` 的最外层是字典，字典中键的值是一个列表。

```
In [1]:
# 字典的值是列表
data = {'name': ['张三', '李四'], 'age': [18, 20]}
pd.DataFrame(data)

Out[1]:
   name age
0 张三  18
1 李四  20
```

任务 3.1 Chrome 网络抓包工具的使用

常用的财经信息网站有新浪财经、雪球财经、巨潮资讯网等，本书选择巨潮资讯网作为数据源，通过分析网站的 XHR 请求，获取真实的请求地址，然后对返回的 json 数据进行整理清洗，最终将数据保存为 Excel 文件。

当用户通过浏览器输入一个网址时，浏览器会呈现该网址对应的官方页面。这个过程称为 HTTP 请求，其完整生命周期如下。

- (1) 对输入的网址进行 DNS 域名解析，找到网址对应的 IP 地址与端口。
- (2) 根据这个 IP 与端口，找到服务器上的应用，发起 TCP 的三次握手。
- (3) 建立 TCP 连接后发起 HTTP 请求。
- (4) 服务器响应 HTTP 请求，浏览器得到 HTML 代码。

(5) 浏览器解析响应 HTML 代码，并请求 HTML 代码中的资源（JavaScript、CSS、图片等）。

(6) 浏览器对页面进行渲染，呈现给用户。

(7) 服务器关闭 TCP 连接，四次挥手。

在整个周期当中，数据采集仅关心两个过程：一个是请求（request），浏览器向服务器请求了什么数据；另一个是响应（response），服务器为浏览器响应了什么数据。通过 Chrome 浏览器的开发者工具可以截获 HTTP 的请求与响应数据。

单击 Chrome 浏览器右上方的“设置”图标，选择下拉菜单中的“更多工具”→“开发者工具”选项，就可以打开开发者工具，如图 3.1 所示。



图 3.1 打开开发者工具

也可以通过快捷键 F12，快速打开开发者工具。单击 Network 选项卡，打开 HTTP 数据抓包工具，如图 3.2 所示。

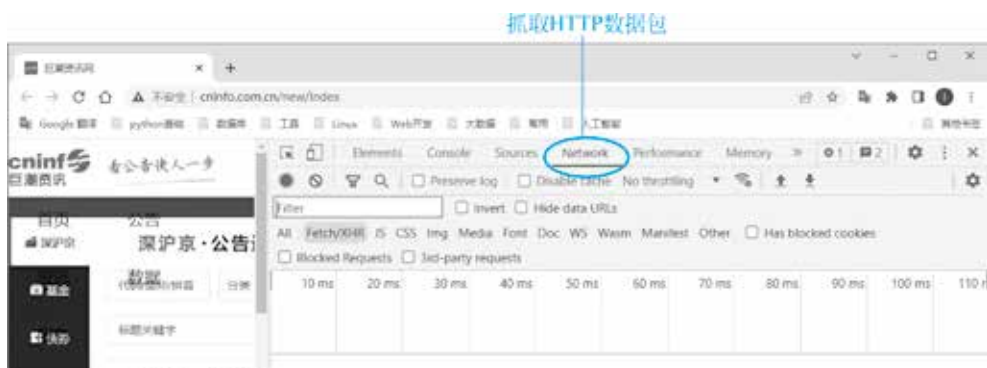


图 3.2 HTTP 抓包工具



任务 3.2 通过截获 XHR 请求采集数据

步骤 1 通过抓包工具获取页面真实请求地址。

打开巨潮资讯官网首页，在搜索框输入股票代码，如 600893，单击下方出现的上市公司，如图 3.3 所示。



图 3.3 打开网址，输入股票代码

进入股票详情页面后，按 F12 键打开开发者工具，单击 Network 选项卡，如图 3.4 所示。



图 3.4 打开 Network HTTP 抓包工具

先单击右侧 Fetch/XHR 选项卡，目的是在抓取到的数据包中只保留 XHR 请求数据，然后单击页面左侧导航菜单中的财务数据——财务报表，如图 3.5 所示。注意图中单击的顺序。



图 3.5 筛选 XHR 请求

在右侧的开发者工具中，会抓取到相应的 HTTP 数据，单击其中的第一项来查看请求的详情。在详情中，单击右侧的 Response（响应），查看返回的数据，如图 3.6 所示。

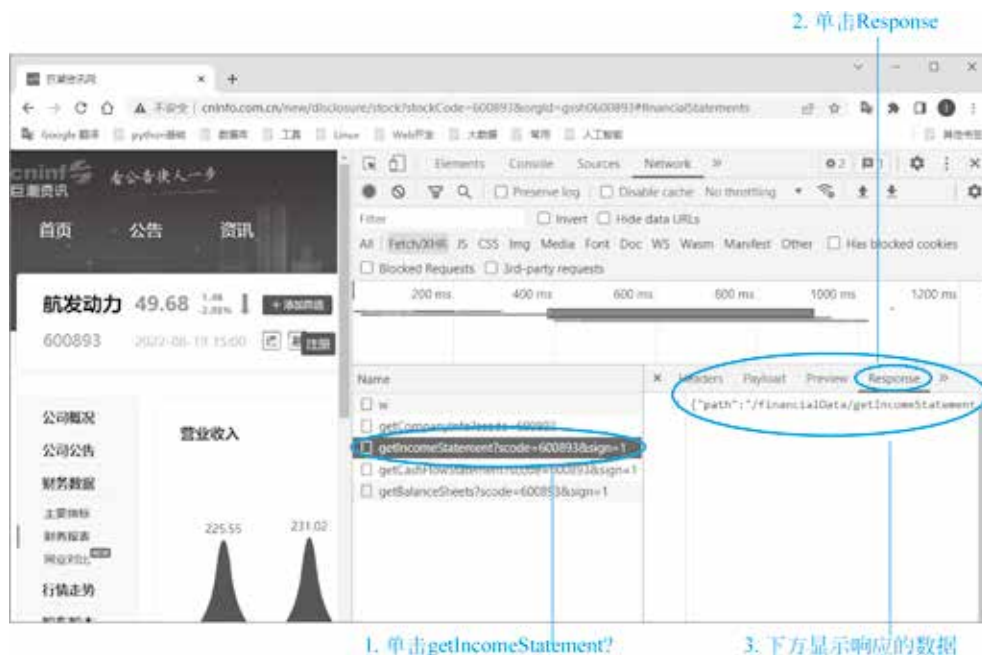


图 3.6 查看返回数据

右击数据包链接，选择 Copy → Copy link address 命令并单击，复制链接地址，用于下一步的 HTTP 请求，如图 3.7 所示。



图 3.7 复制链接地址

步骤2 分析 json 数据的层次结构。

打开一个新的浏览器窗口，将任务 3.2 中复制的网址粘贴到浏览器中打开。在浏览器中查看数据，如图 3.8 所示。

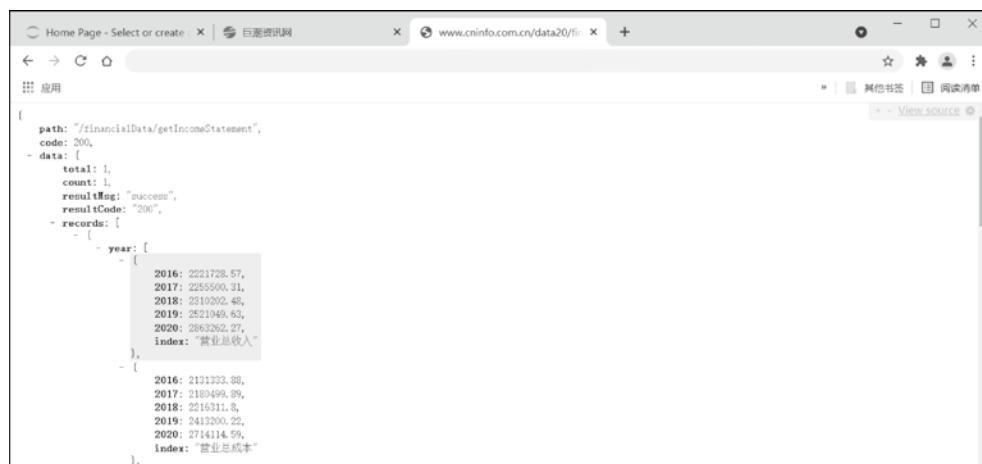


图 3.8 在浏览器中查看数据

单击图 3.8 中 - 号，通过对数据进行折叠与展开，分析 json 的层级结构。如果数据的外层被 {} 包含，则表示为一个对象数据，最终会转换为 Python 的字典来处理，索引方式是通过“键”索引出数据。如果数据的外层被 [] 包含，则表示数据为一个数字，最终会转换为 Python 列表来进行处理，通过列表的下标索引出数据，如图 3.9 所示。

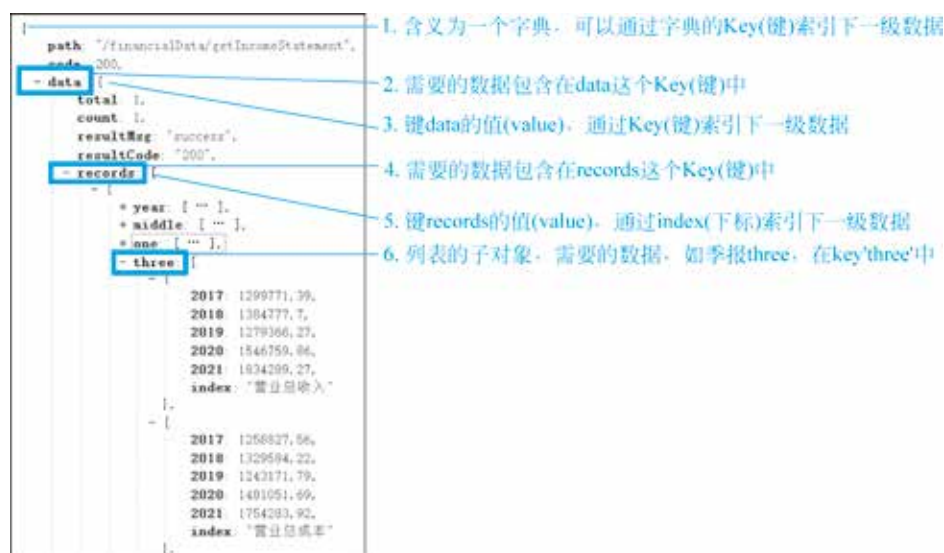


图 3.9 json 数据解析

步骤 3 提取数据。

在“此电脑/D 盘”建立项目文件夹并命名为“财经数据”，在文件夹中启动 jupyter notebook，如图 3.10 所示。



图 3.10 启动 jupyter notebook

新建 python 3 文件，更改文件名为“财经数据 API”，在 jupyter notebook 中，编写代码。导入 requests 库，设置变量 url 为请求地址（也就是任务 3.2 中复制的网址），请求数据和返回的数据是 json 格式，将返回值赋值给变量 r，通过 r 的 json() 方法，将 json 字符串转换为 Python 对象，通过多级索引，得到最终的列表数据 data，代码如下。

```
In [1]:
# 导入库
import requests
# 设置网址
url = 'http://www.cninfo.com.cn/data20/financialData/' \
      'getIncomeStatement?scode=600893&sign=1'
# 对 url 发起请求，请求方式为 get
r = requests.get(url)
# 将返回数据通过 json() 方法，转换为 Python 对象
s = r.json()
```



```
# 索引出需要的数据, 季报
data = s['data']['records'][0]['three']
data
Out [1]:
[{'2019': 1279366.27,
  '2018': 1384777.7,
  '2017': 1299771.39,
  '2016': 1259901.42,
  'index': ' 营业总收入 ',
  '2020': 1546759.86},
 ...
  'index': ' 归属母公司净利润 ',
  '2020': 63352.18}]
```

从输出结果可以看到, 已经得到一个 **DataFrame** 所需要的格式, 即一个列表包含了多个字典, 且字典的键都相同。

此时, 如果想得到某一项单独的数据, 可以利用列表的下标, 或者键来索引。例如, 想知道该公司 2016 年的营业总收入, 代码如下。

```
In [2]:
data[0]['2016']
Out [2]:
1259901.42
```

在上述代码中, 营业总收入为列表中的第一个对象, 下标为 0。2016 年的数据在字典中, 键为 2016, 输出结果为 1259901.42。

当然, 更多的时候, 我们需要所有的数据, 而不是某一项数据, 这时就可以利用 **Pandas**, 将这种形式的数据转换为 **DataFrame** 进行处理, 代码如下。

```
In [3]:
import pandas as pd
df = pd.DataFrame(data)
df
Out [3]:
```

	2019	2018	2017	2016	index	2020
0	1279366.27	1384777.70	1299771.39	1259901.42	营业总收入	1546759.86
1	1243171.79	1329584.22	1258827.56	1228895.06	营业总成本	1481051.69
2	54408.17	75250.40	43280.36	36041.60	营业利润	78063.16
3	53432.29	76060.63	46947.92	45623.37	利润总额	77691.25
4	8096.14	8897.74	9304.99	10809.13	所得税	11743.74
5	41326.79	65049.80	36493.08	30026.37	归属母公司净利润	63352.18

输出 **DataFrame**, 将原始数据中的营业总收入、营业总成本、营业利润、利润总额、所得税、归属母公司净利润共 6 项数据, 从 2016~2020 年共 5 年的财报数据, 全部显示在表中完成。