

# 我将实时疫情数据爬取下来并做了展示

原创 潮汐 Python技术 2月26日

今天是中国按下暂停键的第 25 天，在中国按下暂停键的日子里，主人翁每天早上睁眼第一件事就是打开手机看着疫情实时数据的变化，看看每一条催泪的新闻。揪着的心却在默默祈祷疫情赶快过去。

言归正传，对于一个热衷技术且大有前途的青年来说，数据看久了是不是想到要展示一个技术大白的真正技术了呢？今天的文章主人翁就抱着学习的态度将腾讯每天推送的实时疫情数据爬取下来进行数据展示。

思路：

1. 网页分析
2. 实时数据抓取
3. 数据可视化展示

## 网页分析：

在百度中搜索 <https://news.qq.com/zt2020/page/feiyang.htm> 即可获得疫情实时追踪展示信息：

在此网页任意地方 右键单击 ---> 检查或者审查元素 查看源代码,或者打开浏览器开发者模式，然后直接按 F12 查看源代码，再查看网络反馈的消息，如下图所示：

消息响应具体信息如下：

由上可知数据为 JSON 类型的数据。

## 实时数据抓取

通过网页分析后了解到疫情数据是 JSON 类型，因此爬取的主要原理是通过 Requests 获取 Json 请求。然后再获取到各省份的相关数据，数据抓取代码如下：

```
1 import time, json, requests
2 # 腾讯疫情实时数据数据 URL
3 url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5&callback=&_=' + str(int(time.time()*1000))
4 # 加载 JSON 数据并解析
5 data = json.loads(requests.get(url=url).json()['data'])
6 # 打印数据
7 print(data)
8 print(data.keys())
```

输出数据部分信息如下：

```
1 data: {
2   'lastUpdateTime': '2020-02-24 16:04:30',
3   'chinaTotal': {
```

```

3      'confirm': 77262,
4      'heal': 24839,
5      'dead': 2596,
6      'nowConfirm': 49827,
7      'suspect': 3434,
8      'nowSevere': 9915
9  },
10 'chinaAdd': {
11     'confirm': 416,
12     'heal': 1932,
13     'dead': 151,
14     'nowConfirm': -1667,
15     'suspect': -714,
16     'nowSevere': -1053
17 },
18 .....
19
20 keys: dict_keys(['lastUpdateTime', 'chinaTotal', 'chinaAdd', 'isShowAdd', 'showAddSwitch', 'areaTree
21
22

```

至此疫情数据就获取完毕，So yesy 的有木有，接下来需要统计疫情数据的 34 个省份信息，首先分析输出的数据，经过分析发现省份数据的 Json 头为 areaTree, 详细信息如下图所示：

获取省份详细代码如下：

```

1 import time, json, requests
2 # 腾讯疫情实时数据数据 URL
3 url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5&callback=&_=%d'%int(time.time()*1000)
4 # 加载 JSON 数据并解析
5 data = json.loads(requests.get(url=url).json()['data'])
6 # 打印数据
7 print(data)
8 print(data.keys())
9
10 # 统计省份信息(34个省份 湖北 广东 河南 浙江 湖南 安徽....)
11 num = data['areaTree'][0]['children']
12 print(len(num))
13 # 遍历所有数据后输出，直到输出结束
14 for item in num:
15     print(item['name'],end=" ")
16 else:
17     print("\n")

```

输出信息如下：

```

1 {'lastUpdateTime': '2020-02-24 17:06:26', 'chinaTotal': {'confirm': 77262, 'heal': 24839, 'dead': 259
dict keys(['lastUpdateTime', 'chinaTotal', 'chinaAdd', 'isShowAdd', 'showAddSwitch', 'areaTree', 'chi

```

```
2 34
3 湖北 广东 河南 浙江 湖南 安徽 江西 山东 江苏 重庆 四川 黑龙江 北京 上海 河北 福建 广西 陕西 云南 海南 贵州 天津
4
5
```

34 个省份数据获取完毕后接下来需要解析全国已确诊的省份对应数据，详细代码如下：

```
1 import time, json, requests
2 # 腾讯疫情实时数据数据 URL
3 url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5&callback=&_=%d'%int(time.time()*1000)
4 # 加载 JSON 数据并解析
5 data = json.loads(requests.get(url=url).json()['data'])
6 # 打印数据输出数据
7 print(data)
8 print(data.keys())
9
10 # 统计省份信息(34个省份 湖北 广东 河南 浙江 湖南 安徽....)
11 num_area = data['areaTree'][0]['children']
12 print(len(num_area))
13 # 遍历所有数据后输出，直到输出结束
14 for item in num_area:
15     print(item['name'],end=" ")
16 else:
17     print("\n")
18
19 # 解析所有确诊数据
20 all_data = {}
21 for item in num_area:
22     # 输出省市名称
23     if item['name'] not in all_data:
24         all_data.update({item['name']:0})
25     #输出省市对应的数据
26     for city_data in item['children']:
27         all_data[item['name']] +=int(city_data['total']['confirm'])
28 # 输出结果
29 print(all_data)
30
```

结果输出为：

```
1 {'lastUpdateTime': '2020-02-24 17:06:26', 'chinaTotal': {'confirm': 77262, 'heal': 24839, 'dead': 259
2 dict_keys(['lastUpdateTime', 'chinaTotal', 'chinaAdd', 'isShowAdd', 'showAddSwitch', 'areaTree', 'chi
3 34
4 湖北 广东 河南 浙江 湖南 安徽 江西 山东 江苏 重庆 四川 黑龙江 北京 上海 河北 福建 广西 陕西 云南 海南 贵州 天津
5
6 {'湖北': 64287, '广东': 1345, '河南': 1271, '浙江': 1205, '湖南': 1016, '安徽': 989, '江西': 934, '山东':
7
```

接下来我们对比一下抓取的数据是否正确：实时数据为：

抓取的数据为：

```
'湖北': 64287,  
'广东': 1345,  
'河南': 1271,  
'浙江': 1205,  
'湖南': 1016,  
'安徽': 989,  
'江西': 934,  
'山东': 755,  
'江苏': 631,  
'重庆': 575,  
'四川': 527,
```

由此可见抓取的数据和实时的数据无差异。

## 数据可视化展示

接下来我们将对已经解析好的数据进行可视化展示，本文可视化知识点使用的是公众号 [100 天学习计划的知识点](#) Matplotlib，详情参考文章：[第91天：Python matplotlib introduction](#) 和 [第92天：Python Matplotlib 进阶操作](#)

数据展示思路和代码如下：

```
1 # 使用 Matplotlib 绘制全国确诊病例柱状图  
2 import matplotlib.pyplot as plt  
3 import numpy as np  
4  
5 plt.rcParams['font.sans-serif'] = ['SimHei'] #正常显示中文标签  
6 plt.rcParams['axes.unicode_minus'] = False #正常显示负号  
7  
8 #获取数据  
9 names = all_data.keys()  
10 nums = all_data.values()  
11 print(names)  
12 print(nums)  
13  
14 # 绘图  
15 plt.figure(figsize=[11,7])  
16 plt.bar(names, nums, width=0.8, color='purple')  
17  
18 # 设置标题  
19 plt.xlabel("地区", fontproperties='SimHei', size=15)  
20 plt.ylabel("人数", fontproperties='SimHei', rotation=90, size=12)  
21 plt.title("全国疫情确诊图", fontproperties='SimHei', size=16)  
22 plt.xticks(list(names), fontproperties='SimHei', rotation=-60, size=10)  
23  
24 # 显示数字  
25 for a, b in zip(list(names), list(nums)):  
26     plt.text(a, b, b, ha='center', va='bottom', size=6)  
27  
28 # 图形展示
```

```
29 plt.show()
```

疫情数据可视化展示完毕！

## 总结

本文用到的思路及知识点总结如下：

数据爬取思路：网站分析 --> 获取数据 --> 解析数据

数据可视化思路：Matplotlib 基本知识的运用 --> 分析数据-->画图--> 展示图形

最后希望疫情早点结束，没有一个冬天不可逾越，没有一个春天不会到来，中国加油！武汉加油！

## 参考

<https://blog.csdn.net/Eastmount/article/details/104298388>

<https://blog.csdn.net/xufive/article/details/104282093>

<https://blog.csdn.net/xufive/article/details/104093197>

<https://blog.csdn.net/shineych/article/details/104173449>

文中示例代码：<https://github.com/JustDoPython/python-100-day>

**PS：**公号内回复：Python，即可进入Python 新手学习交流群，一起**100天计划**！

-END-

**Python 技术**

关于 Python 都在这里