

一、 数据挖掘的逻辑和流程描述

a) 数据挖掘逻辑

数据挖掘一般有两种思路。第一、通过网页获取所需要的数据，具体是从所爬取的网页上爬取所需要的数据进行挖掘分析，一般应用在静态网页或者没有重定向的网页中；第二、通过接口或者找到重定向后的网页，最终通过分析接口或最终网页数据来实现数据挖掘目的，一般应用在比较复杂的场景下。

通过上面的两种思路，最终可以获得所需要的数据，然后结合最终的目标对所采集到的数据进行加工、处理和存储，最终提供满足要求的分析数据。

具体到案例，我们的目标是要获取海底捞公众号的排号数据，我们需要清楚网页上的排号数据是哪一种情况，然后根据针对具体情况开展相应的数据挖掘工作。

b) 数据挖掘流程

首先，我们需要明确数据挖掘的目标是什么，这个是做数据挖掘最根本的问题。

其次，我们需要把目标分解成一个个小目标，这里主要是把大目标细分成一个个小目标，通过这些小目标和具体的数据建立联系。

然后，我们根据上一步的小目标，明确需要采集的数据。

再然后，我们根据所采集到的数据，搭建模型进行分析，通过分析每一个数据来实现一个个小目标，最终把所有小目标拼接起来，实现我们第一步的最终目标。

最后，我们需要验证建模分析有没有实现我们的数据挖掘目标，如果没有实现，我们要回过头一步步去梳理，确定那个环节出问题，然后在着力解决。

根据上面的思路，整理的流程图如下：

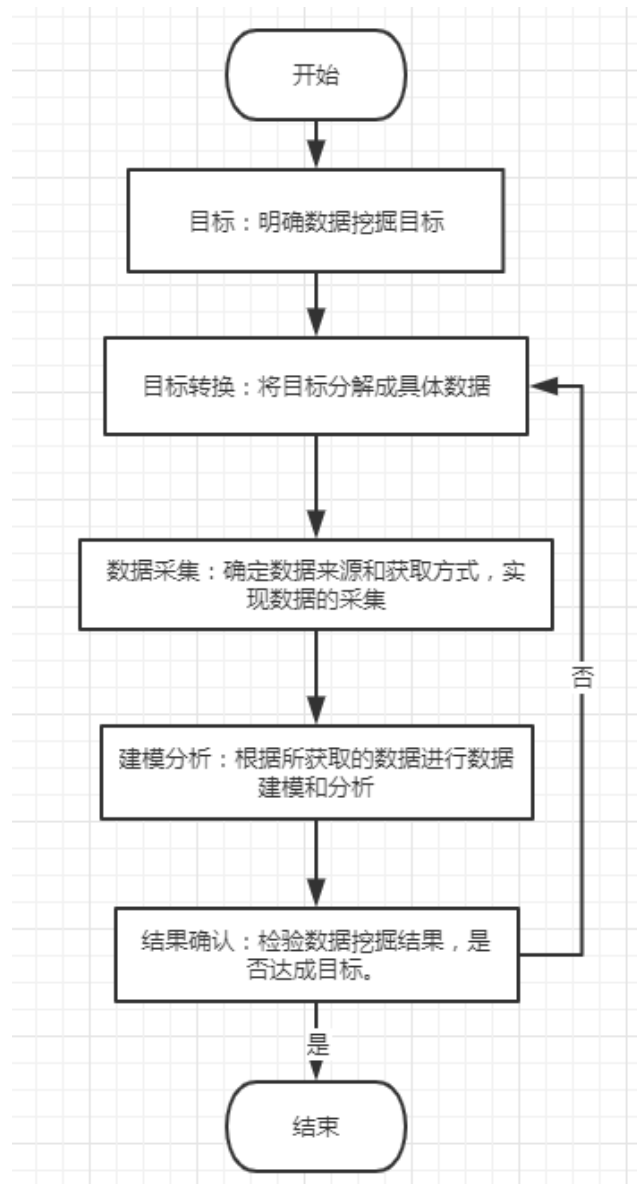


图 1 数据挖掘流程图

二、 使用工具和操作步骤的详细说明

a) 使用工具

Google Chrome 开发者工具

Postman

Python 3 (IDE: PyCharm)

b) 操作步骤

i. 分析海底捞公众号排号链接

通过海底捞微信公众号，查看海底捞公众号排号链接地址，把链接拷贝到 PC 端 Google Chrome 浏览器，通过开发者工具进行分析。

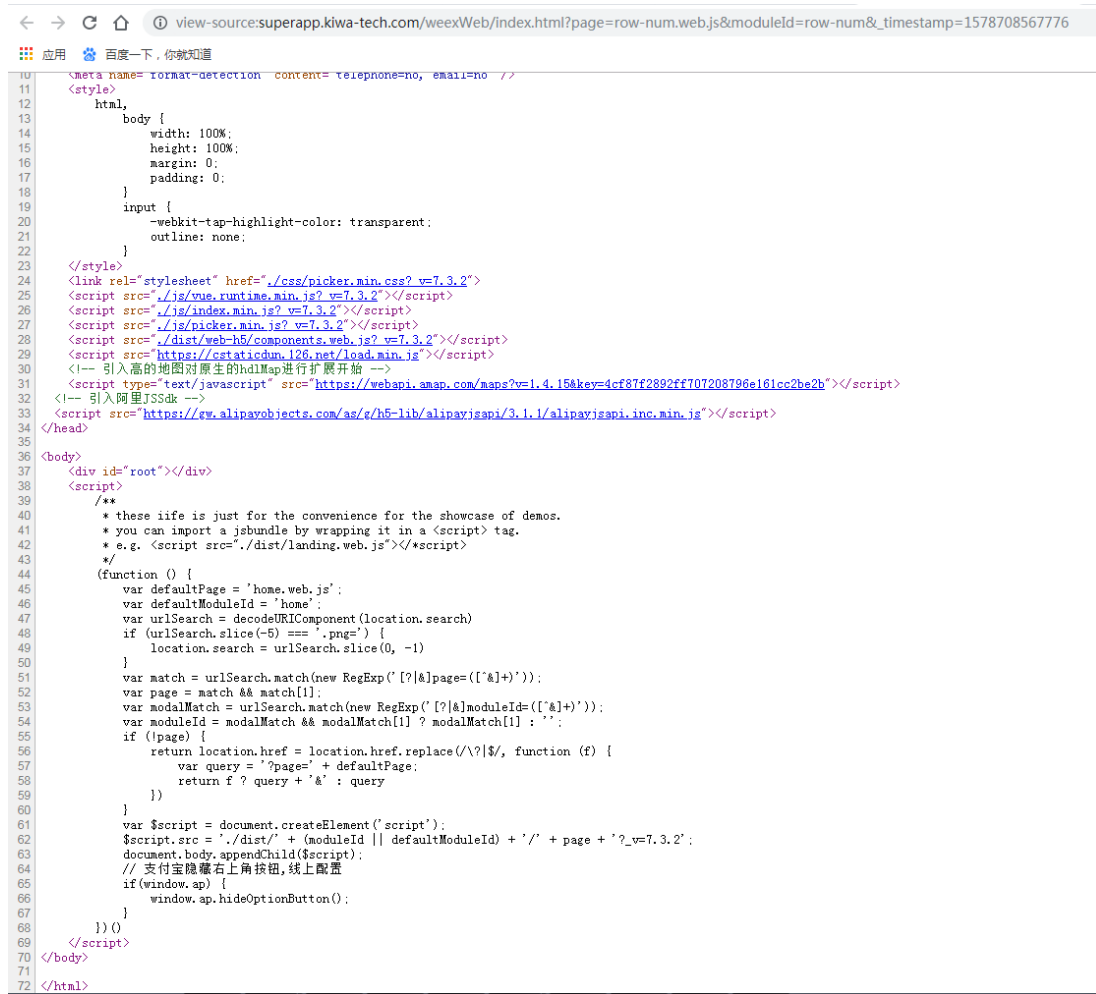


图 2 海底捞公众号排号链接

ii. 分析海底捞公众号排号数据来源

把上一步得到的链接拷贝到 Google Chrome，通过开发者工具以手机视图方式打开链接，分析网页收发数据逻辑，在查看网页源代码的时候我们并没有发现具体的数据，我们可以确认海海底捞排号系统采用的是接口或者是 javascript 异步方式展示

的数据，这里主要是观察 Network 下的 XHR 请求，通过网页分析，我们可以知道海底捞排号系统采用的是 weex 框架的 Ajax XHR 方式交互数据，具体截图如下：



```
10 <meta name="format-detection" content="telephone=no, email=no" />
11 <style>
12   html,
13     body {
14       width: 100%;
15       height: 100%;
16       margin: 0;
17       padding: 0;
18     }
19     input {
20       -webkit-tap-highlight-color: transparent;
21       outline: none;
22     }
23   </style>
24   <link rel="stylesheet" href="/css/picker.min.css?v=7.3.2">
25   <script src="/js/vue.runtime.min.js?v=7.3.2"></script>
26   <script src="/js/index.min.js?v=7.3.2"></script>
27   <script src="/js/picker.min.js?v=7.3.2"></script>
28   <script src="/dist/web-h5/components.web.js?v=7.3.2"></script>
29   <script src="https://cstaticdm.l26.net/load.min.js"></script>
30   <!-- 引入高的地图对原生的h5Map进行扩展开始 -->
31   <script type="text/javascript" src="https://webapi.amap.com/maps?v=1.4.15&key=4cf87f2892ff707208796e161cc2be2b"></script>
32   <!-- 引入阿里JS SDK -->
33   <script src="https://gw.alipayobjects.com/as/e/h5-lib/alipayjsapi/3.1.1/alipayjsapi.inc.min.js"></script>
34 </head>
35
36 <body>
37   <div id="root"></div>
38   <script>
39     /**
40      * these ife is just for the convenience for the showcase of demos.
41      * you can import a jsbundle by wrapping it in a <script> tag.
42      * e.g. <script src="/dist/landing.web.js"></script>
43      */
44     (function () {
45       var defaultPage = 'home.web.js';
46       var defaultModuleId = 'home';
47       var urlSearch = decodeURIComponent(location.search);
48       if (urlSearch.slice(-5) === '.png') {
49         location.search = urlSearch.slice(0, -1)
50       }
51       var match = urlSearch.match(new RegExp('(?:&|)page=([&|]+)'));
52       var page = match && match[1];
53       var modalMatch = urlSearch.match(new RegExp('(?:&|)moduleId=([&|]+)'));
54       var moduleId = modalMatch && modalMatch[1] ? modalMatch[1] : '';
55       if (!page) {
56         return location.href = location.href.replace(/(?:&|)$/, function (f) {
57           var query = '?page=' + defaultPage;
58           return f ? query + '&' : query
59         })
60       }
61       var $script = document.createElement('script');
62       $script.src = './dist/' + (moduleId || defaultModuleId) + '/' + page + '?_v=7.3.2';
63       document.body.appendChild($script);
64       // 支付宝隐藏右上角按钮，线上配置
65       if (window.ap) {
66         window.ap.hideOptionButton();
67       }
68     })()
69   </script>
70 </body>
71
72 </html>
```

图 3 静态网页源代码

iii. 确认海底捞排号数据接口

通过上一步确认，当我们刷新页面的时候，我们找到了 XHR 调用的三个接口，结合具体接口命名和返回的数据，我们大体上知道这三个接口的功能。

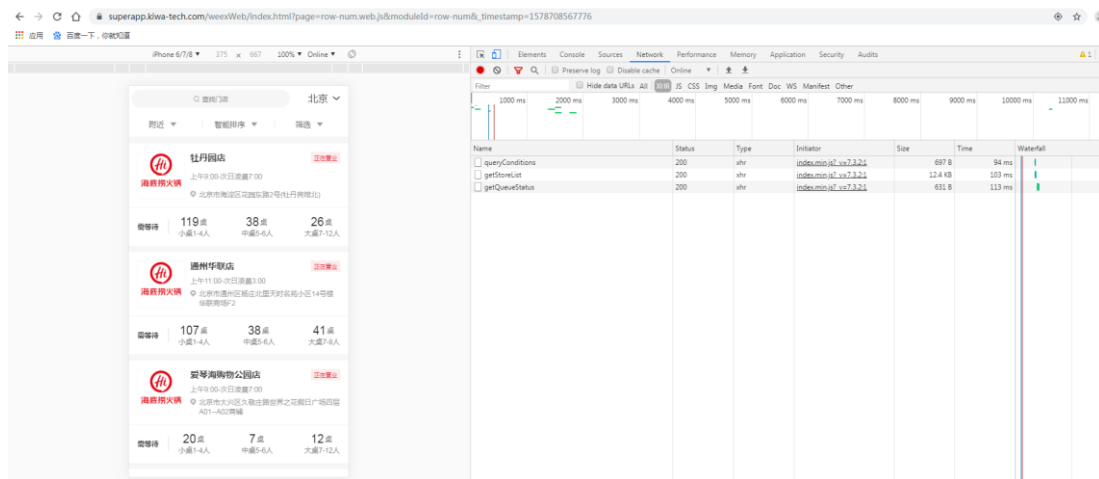


图 4 XHR 请求接口

通过观察这三个 XHR 的 Headers 和 Response 的信息，基本上能确定下来这三个接口的具体功能。

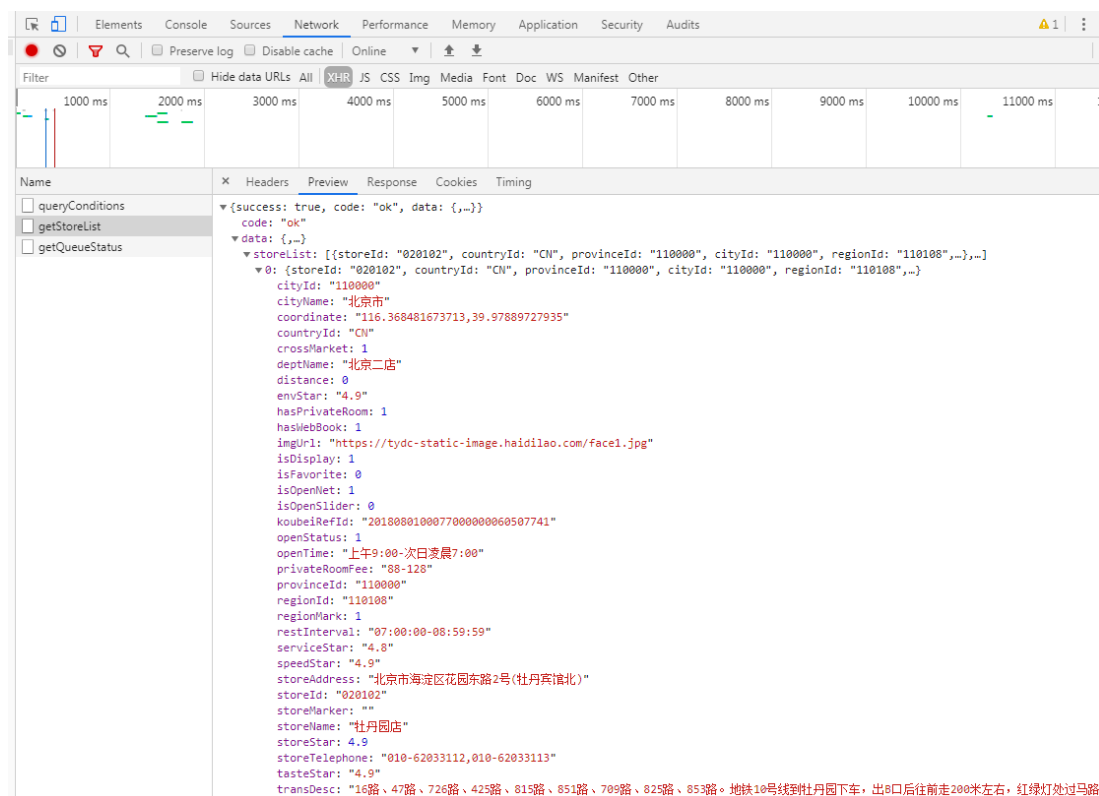


图 5 queryStoreList 接口信息

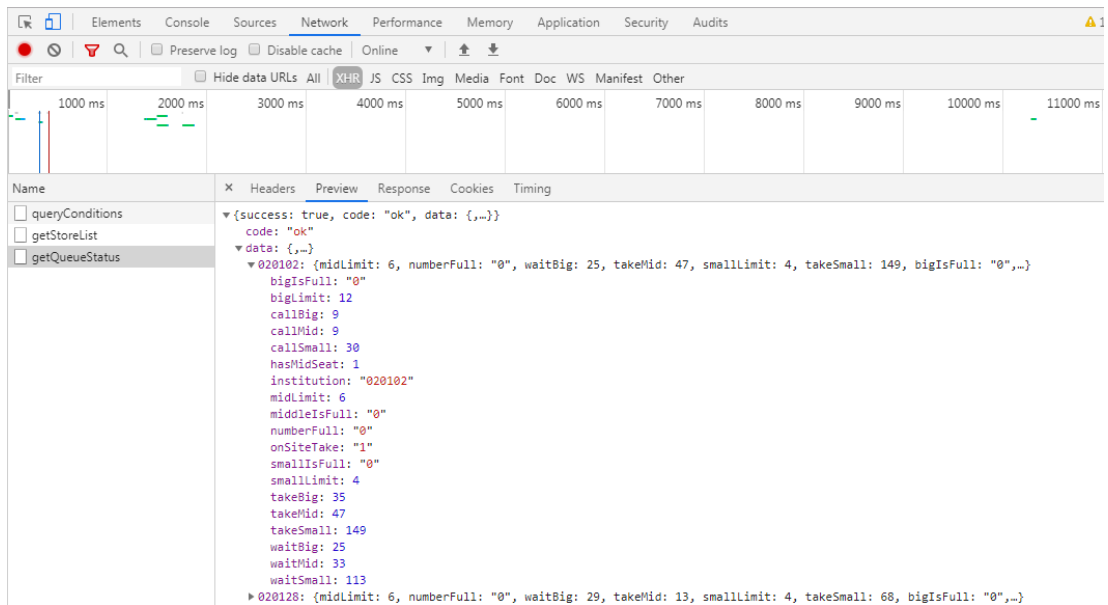


图 6 getQueueStatus 接口信息

接口	链接	功能
queryConditions	https://superapp.kiwa-tech.com/app/v2/queryConditions	查询并返回城市及区域信息
getStoreList	https://superapp.kiwa-tech.com/app/v2/getStoreList	查询并返回区域门店信息，包括国家、省市、位置和营业等数据
getQueueStatus	https://superapp.kiwa-tech.com/app/v2/getQueueStatus	查询并返回具体门店的排队信息，门店号跟 getStoreList 对应。

iv. 分析排队等位数据逻辑

通过前面几个步骤我们解决了数据来源的问题，接下来要解决的是数据准确性问题，通过三个接口返回来的数据，我们并没有发现有跟页面显示完全一致的数据，可以确认的是显示的数据肯定是经过加工处理的，经过多个数据验证我们最终发现了这个规律。

网页上显示的排队数据和接口返回的数据存在以下关系：对应类型的网页排队数据等于对应的 take 数据减去 call 数据，具体关系见图 7。

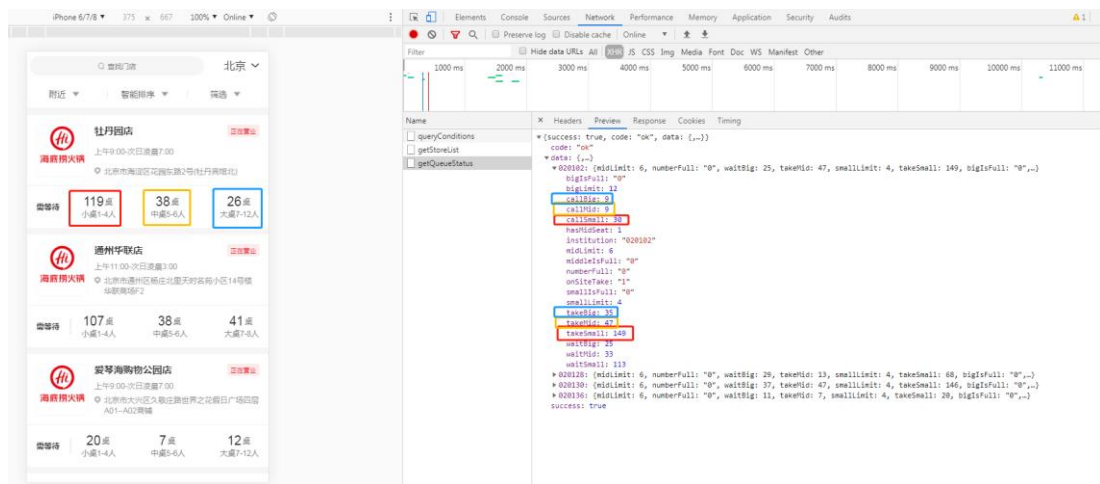


图 7 显示数据和接口数据逻辑关系

至此，数据挖掘分析工作已经基本完成，接下来是通过 Python 编码实现。

v. 通过 Python 代码实现数据采集工作

在这里我们通过 Postman 工具向接口发送 Request 请求，查看返回的结果。在构造请求的时候，我们需要把之前在网页上查看到的接口的 Header 和 Request Payload 信息拷贝到 Postman 中去，我们看到返回的数据是 json 格式的。

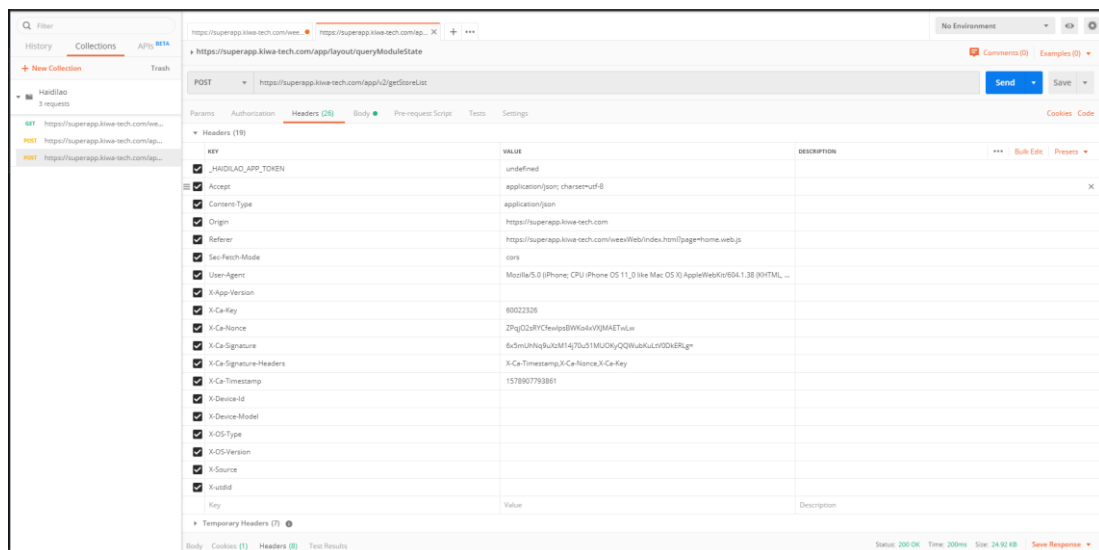


图 8 postman 请求参数设置

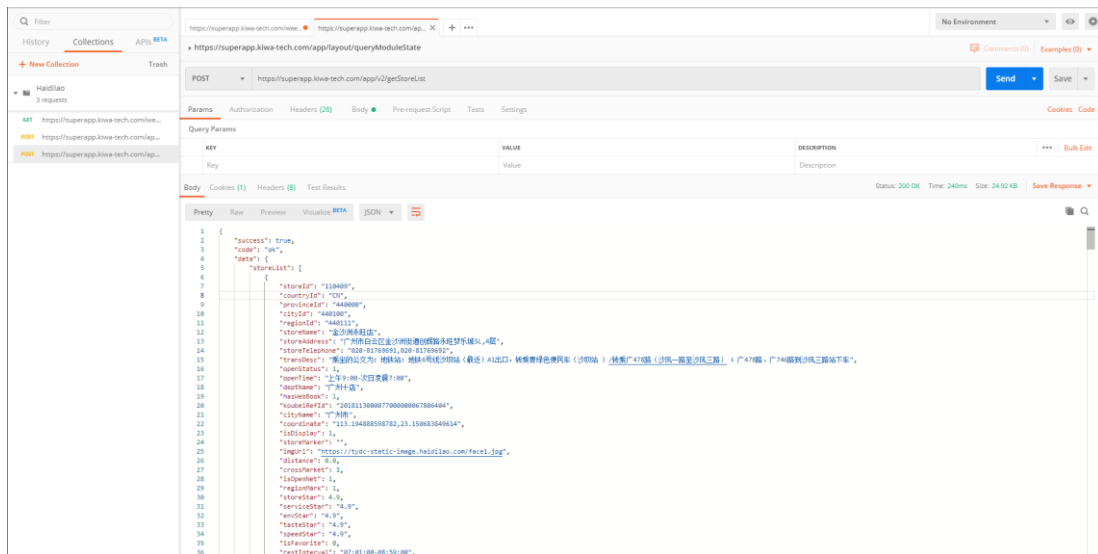


图 9 接口返回的数据

在设置 Request Payload 参数的时候，我们发现四个参数，cityId,coordinate,regionId,sort 这也是我们通过 POST 传入的参数，当我们一个个参数调试的时候，我们发现当我们只保留一个参数 coordinate 的时候，返回的是所有门店的信息包括国内和国外的，然后我们可以通过得到的所有门店信息，根据 countryId 等于 CN 的筛选出全国的所有海底捞门店。（注意：这里包括港澳台的门店，只是实际上港澳台的等位信息为空。）

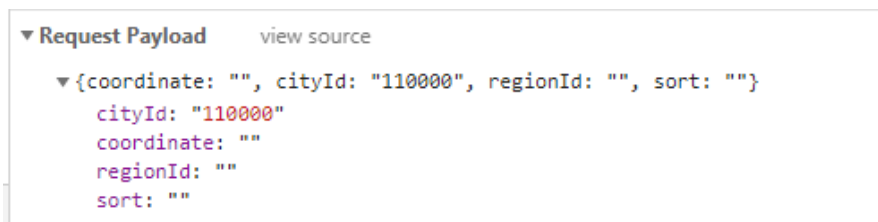


图 10 Request Payload 参数

整个流程打通之后，我们就可以编写 Python 程序了，这个时候我们把刚才用到的数据当做 header、url 参数在 Python 中直接定义，通过 Python 的 requests 模块的 post 方法直接访问接口，最后，采集到的数据如下：

序号	城市+门店名	小桌等位数	中桌等位数	大桌等位数
1	佛山市佛魁奇二路店	58	26	16
2	深圳市新沙天虹店	0	0	0
3	深圳市龙岗星河店	71	18	33
4	聊城市五星百货店	22	4	11
5	莆田市城厢万达广场店	160	53	28
6	天津市嘉茂店	216	44	38
7	广州市金沙洲永旺店	64	20	17
8	无锡市苏宁广场店	181	32	22
9	广州市珠影星光城店	265	93	105
10	北京市牡丹园店	203	49	51
11	西安市文景店	109	63	93
12	北京市通州华联店	130	56	44
13	宜昌市万达广场店	11	12	12
14	上海市五角场店	289	80	107
15	桂林市东西巷店	64	14	6
16	杭州市下沙银泰店	190	96	110
17	北京市爱琴海购物公园店	38	23	14
18	北京市大钟寺店	17	2	66
19	上海市海宁路店	153	41	61
20	大理市泰业国际店	19	8	7
21	宝鸡市宝鸡银泰店	71	34	17
22	北京市太阳宫店	156	60	70
23	潍坊市泰华假日店	143	34	12
24	深圳市卓越店	19	11	0
25	西安市大明宫店	124	79	71
26	上海市沪太路店	47	6	5
27	无锡市海澜财富中心店	64	58	45
28	上海市逸仙路店	131	53	31
29	上海市淮海路店	488	0	281
30	郑州市丹尼斯三天地店	4	7	7
31	深圳市海雅缤纷店	136	23	40
32	厦门市世贸商城店	0	0	0
33	杭州市中大银泰城店	247	88	80
34	北京市紫竹桥店	291	55	65
35	洛阳市泉舜店	71	48	46
36	长沙市华创店	69	13	13
37	深圳市佳华领汇店	118	47	41
38	武汉市世界城店	324	81	67
39	南京市中山南路店	269	74	104
40	郑州市大商国贸店	210	112	134
41	北京市龙湖大兴天街店	32	13	11
42	南京市仙林大学城店	196	83	72

图 11 海底捞门店等位数据

三、自行编辑使用的 PYTHON 程序代码文件

代码文件见 Haidilao.py

四、 以 EXCEL 形式呈现的数据抓取结果

数据格式说明：（n 行*5 列，列名分别为“序号”、“城市+门店名”、“小桌等位数”、“中桌等位数”、“大桌等位数”）

数据文件见 [Haidilao.xls](#)