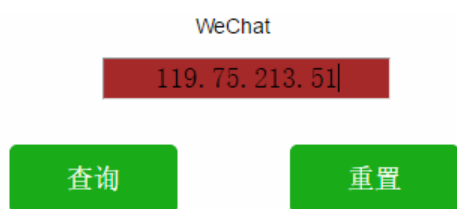


微信小程序之运维小项目

自从微信推出小程序以来，现在业界炒的非常的火，具说叫微信小程序是因为某公司不让叫应用号，我在朋友圈也看过 pony 的和张小龙朋友圈关于名字的讨论截图，不知道是真是假，反正名字是定了，叫啥无所谓，还是知道它是干啥的比较简单重要，像网上说什么新风口了，原生 APP 以后没活路了等相关文章喜欢的可以多看看，我们今天不会在产品层面上去介绍小程序，因为我们是搞运维的，所以我还是在运维层面怎么使用小程序，今天我们用实际例子来讲解和普及小小程序如何使用，来实现类似 IP138 的功能，输入 IP 可以查看 IP 的详细信息，包括归属，地点等，我们先看下完成后的效果：

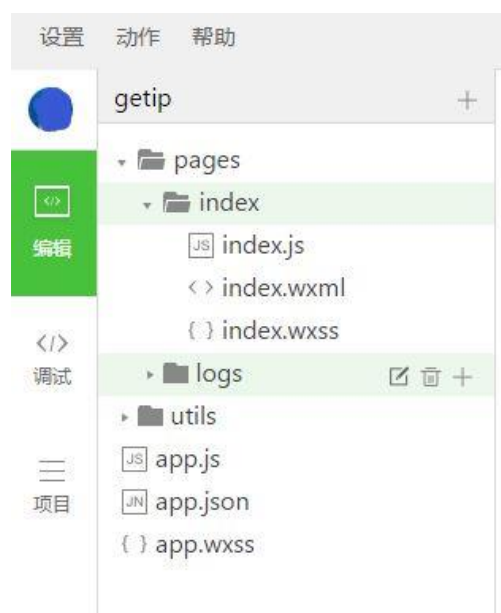


The screenshot shows a WeChat mini-program interface. At the top, the text 'WeChat' is visible. Below it, there is a text input field containing the IP address '119.75.213.51'. Underneath the input field, there are two green buttons: '查询' (Query) on the left and '重置' (Reset) on the right. The interface is simple and clean, with a white background and a light gray border.

基本功能就是做一个 IP 查询的页面，输入 ip，能够查询 IP 的详细信息，后台调用的淘宝的 IP 库，OK，界面和功能就是这样了，比较简单，接下来我们来介绍小程序的基本概念和使用。

第一步，先下载微信小程序开发者工具，这个是必须的，因为只能在这个工具里去调试你的代码，但写代码不一定要在这个工具里去写，在 Sublime 里写也可以，但调试你必须在这工具里，关于怎么下怎么装这我就不多说了，如果这步都没搞定我觉得往下看也没啥意义了，当然是开个玩笑，如果有问题的，可以给我留言。

安装后，第一次运行需要用微信扫描，来识别开发者，然后添加项目，微信开发者工具可以帮你生成一个简单的 demo 项目，生成项目后如图：



我们先看 app.js、app.json、app.wxss 这三个，其中

app.js 是小程序的脚本代码,可以定义全局变量，指定小程序的生命周期函数(onLaunch ,onShow ,onHide , 其它)，

app.json 文件是配置文件，主要配置小程序的页面，所有的页面设置都要写在这个文件里

app.wxss 是公共样式表文件。

除了这些文件，我们还有两个目录，这 2 个目录展示的是 index 页面和 logs 页面，每个目录下如果完整的情况下都会有 4 个文件：

如上图所示 ,现在每个目录下分别是 index.js、index.wxml、index.wxss ,其中.js 后缀的文件是脚本文件 ,.json 后缀的文件是配置文件（非必须），.wxss 后缀的是样式表文件（非必须），.wxml 后缀的文件是页面结构文件。

以上就是小程序的基本概念了，了解了这些，就可以开始我们的小项目开发。

首先我们先写 index.wxml。

代码如下：

index.wxml •

```
1 <form bindsubmit="formBindsubmit" bindreset="formReset">
2   <view style="display:flex;margin-left: 100px; ">
3     <!--<label>请输入要查询的IP: </label>-->
4     <input name="ip" placeholder="请输入要查询的IP!" class="txt"/>
5   </view>
6
7   <view style="display:flex;margin-top:30px;margin-bottom:30px;">
8     <button type="primary" style="width:30%;" formType="submit" class="btn">查询</button>
9     <button type="primary" style="width:30%" formType="reset" class="btn">重置</button>
10  </view>
11 </form>
12
13
14 <view style="margin-left:20px;">
15   <view>{{tip}}</view>
16   <view>{{ip}}</view>
17   <view>{{country}}</view>
18   <view>{{country_id}}</view>
19   <view>{{area}}</view>
20   <view>{{area_id}}</view>
21   <view>{{region}}</view>
22   <view>{{region_id}}</view>
23   <view>{{city}}</view>
24   <view>{{city_id}}</view>
25   <view>{{county}}</view>
26   <view>{{county_id}}</view>
27   <view>{{isp}}</view>
28   <view>{{isp_id}}</view>
29
30 </view>
```

因为界面我加了一部分样式，所以要写 index.wxss 样式文件，代码如下：

index.wxss •

```
1  /**index.wxss**/  
2  .txt {  
3    border: 1px inset;  
4    background-color: brown;  
5    text-align: center;  
6  
7  }  
8  .btn {  
9    color: #00008B;  
10   background-color: #ADD8E6;  
11   border: 1px 2px 1px 2px;  
12 }
```

有了这些只是个壳子，输入 IP 点击查询时没有效果的，因为我们还需要从后台获取数据，现在开始写我们的 index.js 文件，因为 js 文件内容比较多，我们分开介绍，先说下初始化部分，

index.js •

```
1  Page({  
2    data:{  
3      tip:'',  
4      ip:'',  
5      area:''  
6    },  
7  },
```

刚开始都是空，然后是我们的核心函数，先判断 IP 是否为空，如果是就提示 IP 不能为空，如果有 IP，就调用 wx.request 发请求,将获得的数据然后通过 setData 来赋值，如网络有问题，就报网络请求失败，代码如下：

```

8  formBindsubmit:function(e){
9      if(e.detail.value.ip.length==0){
10         this.setData({
11             tip:'提示: IP不能为空!',
12             ip:'',
13
14         })
15     }else{
16         var that = this;
17         wx.request( {
18             url: 'http://ip.taobao.com/service/getIpInfo.php?ip=' + e.detail.value.ip,
19             header: {
20                 "Content-Type": 'application/json'
21             },
22
23             complete: function( res ) {
24                 that.setData( {
25                     tip:'',
26                     ip:'ip是:' + e.detail.value.ip,
27                     country:'国家属于:' + res.data.data.country,
28                     country_id:'国家ID:' + res.data.data.country_id,
29                     area:'地区属于:' + res.data.data.area,
30                     area_id:'地区ID:' + res.data.data.area_id,
31                     region:'区域:' + res.data.data.region,
32                     region_id:'区域ID:' + res.data.data.region_id,
33                     city:'城市:' + res.data.data.city,
34                     city_id:'城市ID:' + res.data.data.city_id,
35                     county:'县:' + res.data.data.county,
36                     county_id:'县ID:' + res.data.data.county_id,
37                     isp:'ISP:' + res.data.data.isp,
38                     isp_id:'ISPID:' + res.data.data.isp_id
39
40                 });
41                 if( res == null || res.data == null ) {
42                     console.error( '网络请求失败' );
43                     return;
44                 }
45             }
46         })
47     }
48 },

```

最后，本淘宝后需要重新输入，当用户点击重置时清空所有的内容，包括刚获取的值，代码如下：

