

构建天气API

要求是自行开发一个天气信息的接口

而且老师已经给出了[免费天气API地址](#)，所以难度不大，实际编码时间很短。

爬取各地区今日天气信息

思路很直白，就是构建URL，爬取，存储返回的json数据。需要注意的是**每分钟阈值为300次，如果超过会禁用3600秒。请谨慎使用**。所以为了不要给免费的接口太多压力，同时也避免被封，我测了一下时间爬一次在0.05~0.1秒左右，不加以限制，马上就封了（~~虽然不知道会不会真的被封~~），我就简单的每次爬取都休眠一段时间，当然你也可以计数300次停止，等下一次1分钟开始。

网站提供了一个city.json的每个地区的cityCode，但是我下载下来有部分地区的cityCode不存在，所以我就自己想办法了。

因为有老师免费提供的ins_city.sql文件，所以我就直接**正则表达式**匹配得到cityCode，然后带上身份伪装的header头，就可以开心的爬取存在文件里。当然你也可以不存，直接就写进数据库里面，其实都无所谓。

```
import requests #请求用的
import time #计时用的
import re,io #正则匹配用的,io是因服务器python是2.7版本的避免open()报错

#正则匹配所有的citycode
cityCode = []
with io.open('ins_city.sql', 'r', encoding="utf-8") as f:
    str = f.read()
    cityCode = re.findall(r"'(\d+?)'", null, '2018/7/29", str)
#伪装一下
headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:62.0) Gecko/20100101 Firefox/62.0'}

#开始计时
startTime = time.time();
#读取每一个cityCode构建一个url请求，将返回json数据存入文件
with io.open('cityCode.txt', 'w', encoding="utf-8") as f:
    for code in cityCode:
        time.sleep(0.2)
        url = "http://t.weather.sojson.com/api/weather/city/" + code
        r = requests.get(url, headers=headers)
        print(url)
        f.write(r.text+'\n')

#打印花费时间
print('[info]耗时: %s' % (time.time() - startTime))
```

效率的话，正则匹配所有的cityCode，秒级吧，1，2秒；后面的话每次爬取我都先睡0.2秒，最后好像花了1300s左右（忘记子）

存进数据库

已经有啦数据，就可以存进数据库了，我这个代码都是在本机写好的，然后上传到服务器，然后运行，老样子先写好代码

```
import MySQLdb,io #填你自己的数据库信息，不知道去application/datebase.php里面看
db = MySQLdb.connect("127.0.0.1", "你的用户名", "你的密码", "你的数据库名", charset=
#中规中矩的更新操作，但还是有小细节需要注意
with io.open('cityCode.txt','r',encoding='utf-8') as f:
    cursor = db.cursor()
    num = 0
    for line in f.readlines():
        num = num + 1
        strline = line.strip() #去掉最后'\n'
        #下一行是记得加两个'', 对我这里已经把数据库weather_info改成varchar (2000)
        sql = "update ins_county set weather_info = '"+strline+"'" where id = "+
        print(num)
        #执行提交操作，异常就回滚
        try:
            cursor.execute(sql)
            db.commit()
        except:
            db.rollback()
#最后关闭数据库连接
db.close()
```

效率的话10s左右吧（没有计时）。

当然你现在还只是写好代码，还需要上传到服务器上，让服务器跑。

我这里在application/api目录下新建了一个apiUpdate的文件夹，里面包含了这些代码和文件，下面截个图：

然后我是ssh登录到服务器，cd到对应的文件夹，运行

```
ssh -p 2022 root@wuchaoqun.xyz #我这里是把端口号改成了2022，正常的话就直接不用写出端口号
cd 你存在文件的目录 #然后ls看看对不对
python updateSQL.py #运行即可
```

php构建api接口

首先在model文件夹下，新建一个php文件，名字随你（注意开头大写，和类名相同），我这里是Weather.php，

```

<?php
namespace app\api\model;

use think\Model;
use think\Db;

class Weather extends Model
{
    public function getCityCode($county_name = '北京')
    {
        $res = Db::name('ins_county')->where('county_name', $county_name)->column('city_code');
        return $res;
    }

    public function getWeather($weather_code = '101010100')
    {
        $res = Db::name('ins_county')->where('weather_code', $weather_code)->column('weather_desc');
        return $res;
    }
}

```

没什么好说的，就是查找的sql语句

然后在controller目录下新建一个php文件（名字无所谓，就是url里面的控制器部分的名字），我这里是Weather.php

← 3.87.94.81_test01
application
api
controller
↺
(共0个目录与0个文件,大小:1.07 KB 获取)
🔍

上传
远程下载
新建 ▾
←
↺
📁
📁 根目录(43G)
🗑️ 回收站
🔍

☐	文件名	大小	修改时间	权限	所有者	操作
☐	 News.php	441 B	2018/11/27 20:30:42	755	www	
☐	 Weather.php	654 B	2018/12/03 15:03:15	755	www	

1
1/1
从1-2条
共2条数据
每页 100 条

```

<?php
namespace app\api\controller;

use think\Controller;

class Weather extends Controller
{
    public function read()
    {
        $weather_code = input('weather_code');
        $model = model('Weather');
        $data = $model->getWeather($weather_code);
        return $data[0];
    }

    public function getCityCode(){
        $county_name = input('county_name');
        $model = model('Weather');
        $data = $model->getCityCode($county_name);
        $data = [
            $county_name => $data[0]
        ];
        return json($data);
    }
}

```

read()接受一个weather_code的参数，关联model，调用model('Weather')的数据库操作，返回json数据

getCityCode()同理，接受一个county_name参数，调用model('Weather')的数据库操作，最后返回json数据

都很好懂。

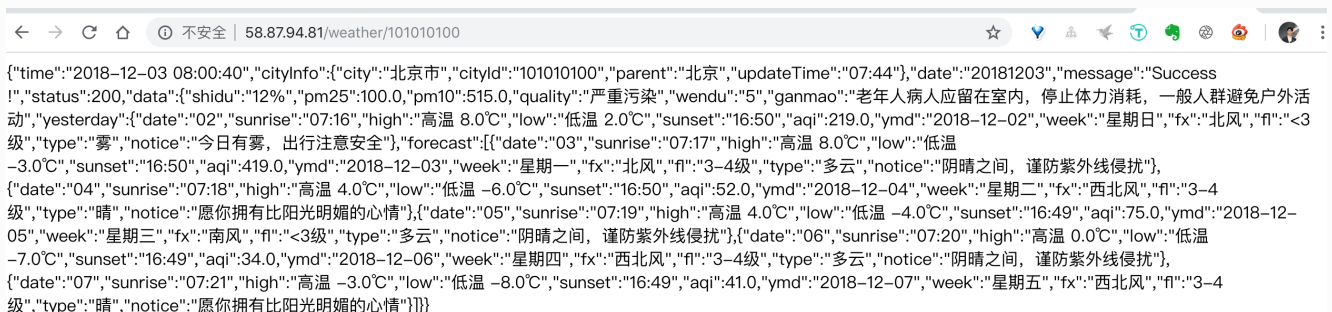
最后按照作业要求去application/route.php文件里面添加路由规则即可。

```

Route::rule('weather/:weather_code', 'api/weather/read');
Route::rule('city/:county_name', 'api/weather/getCityCode');

```

最后api运行结果符合预期



```

{"time":"2018-12-03 08:00:40","cityInfo":{"city":"北京市","cityId":"101010100","parent":"北京","updateTime":"07:44"},"date":"20181203","message":"Success!","status":200,"data":{"shidu":"12%","pm25":100.0,"pm10":515.0,"quality":"严重污染","wendu":"5","ganmao":"老年人病人应留在室内，停止体力消耗，一般人群避免户外活动","yesterday":{"date":"02","sunrise":"07:16","high":"高温 8.0℃","low":"低温 2.0℃","sunset":"16:50","aqi":219.0,"ymd":"2018-12-02","week":"星期日","fx":"北风","fl":"<3级","type":"雾","notice":"今日有雾，出行注意安全"},"forecast":[{"date":"03","sunrise":"07:17","high":"高温 8.0℃","low":"低温 -3.0℃","sunset":"16:50","aqi":419.0,"ymd":"2018-12-03","week":"星期一","fx":"北风","fl":"3-4级","type":"多云","notice":"阴晴之间，谨防紫外线侵扰"}, {"date":"04","sunrise":"07:18","high":"高温 4.0℃","low":"低温 -6.0℃","sunset":"16:50","aqi":52.0,"ymd":"2018-12-04","week":"星期二","fx":"西北风","fl":"3-4级","type":"晴","notice":"愿你拥有比阳光明媚的心情"}, {"date":"05","sunrise":"07:19","high":"高温 4.0℃","low":"低温 -4.0℃","sunset":"16:49","aqi":75.0,"ymd":"2018-12-05","week":"星期三","fx":"南风","fl":"<3级","type":"多云","notice":"阴晴之间，谨防紫外线侵扰"}, {"date":"06","sunrise":"07:20","high":"高温 0.0℃","low":"低温 -7.0℃","sunset":"16:49","aqi":34.0,"ymd":"2018-12-06","week":"星期四","fx":"西北风","fl":"3-4级","type":"多云","notice":"阴晴之间，谨防紫外线侵扰"}, {"date":"07","sunrise":"07:21","high":"高温 -3.0℃","low":"低温 -8.0℃","sunset":"16:49","aqi":41.0,"ymd":"2018-12-07","week":"星期五","fx":"西北风","fl":"3-4级","type":"晴","notice":"愿你拥有比阳光明媚的心情"}]}}

```

```
{"杭州": "101210101"}
```

每日零点更新

crontab 一个Linux命令

可以用来提交和管理用户的需要周期性执行的任务

很明显我们可以把我们刚刚的爬取操作和更新数据库操作作为脚本每天零点让服务器默默更新

首先，我在刚刚的application/api/apiUpdate文件夹里面新建一个update.sh文件

```
vi update.sh
#执行下面的python脚本
python weatherApi.py
python updateSQL.py
```

然后在用户任务调度级别，也就是/var/spool/cron/里面可以看到所有的以用户为名的crontab文件，这里我服务器上只有一个用户就是root，

```
cd /var/spool/cron/
vi root
#添加下面这个任务,具体的路径你自己改成自己的, * * * * * 的含义
#minute: 表示分钟, 可以是从0到59之间的任何整数。
#hour: 表示小时, 可以是从0到23之间的任何整数。
#day: 表示日期, 可以是从1到31之间的任何整数。
#month: 表示月份, 可以是从1到12之间的任何整数。
#表示星期几, 可以是从0到7之间的任何整数, 这里的0或7代表星期日。
#这里我设定零点执行
0 0 * * * /www/wwwroot/58.87.94.81_test01/application/api/apiUpdate/update.sh
```

by chaoqun(吴超群)