

Python 抓取天气预报数据

1801210834 黄捷

主要思路：1.使用 Android 课程提供的 city.db 文件与 API 接口获取天气数据；2. API 返回的 JSON 数据转成 String 存入 MySql，方便后期前端 Ajax 解析；3.在云服务器上持续运行进程，更新数据库。

实现步骤：

1. 数据库准备

- 1) 使用 Sqlite3 .dump 命令，将 Android 课提供的 city.db 转换成 city.sql

如何配置为方便演示，笔者将 city.db 放在 e:/ 根目录

sqlite3 在 Android Studio 配置 adb 环境变量后即可使用。adb 参考《第一行代码》第 2 版 P.214

打开 cmd，切换到 e:盘，用 sqlite3 打开 city.db，输出其中的 city table 到 city.sql

```
E:\>sqlite3 e:/city.db //打开 city.db
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite> .output e:/city.sql //给出输出文件位置及格式
sqlite> .dump city //只转换 city 表，还有一个表是 metadata，不管
sqlite> .quit //退出 sqlite3
```

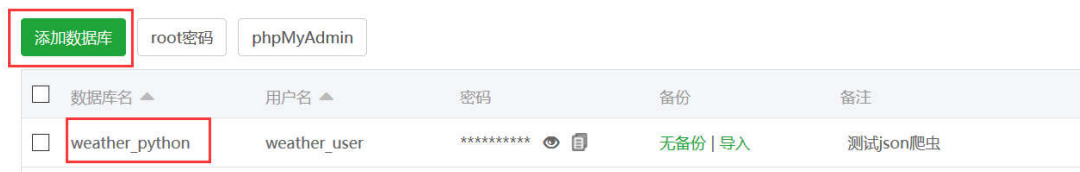
city.sql	2018/12/3 12:38	SQL 文件	207 KB
city.db	2013/9/18 16:50	Data Base File	123 KB

- 2) 调整 city.sql 开头内容，记事本打开：

删除开头：PRAGMA foreign_keys=OFF; BEGIN TRANSACTION，注意保留分号“；”
AUTOINCREMENT 改为 AUTO_INCREMENT

```
city.sql - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
PRAGMA foreign_keys=OFF;BEGIN TRANSACTION;CREATE TABLE city (_id
INTEGER PRIMARY KEY AUTOINCREMENT, province TEXT, city TEXT,
number TEXT, allpy TEXT, allfirstpy TEXT, firstpy TEXT);INSERT INTO city
VALUES(1,'北京','北京','101010100','BEIJING','BJ','B');
```

- 3) 宝塔导入 city.sql, 获得 city 表
新建测试用 database:



phpmyadmin 导入 city.sql 表



增加 weather 列, 保存天气数据:

执行 SQL 语句

ALTER TABLE `city` ADD `weather` TEXT NOT NULL AFTER `firstpy`;



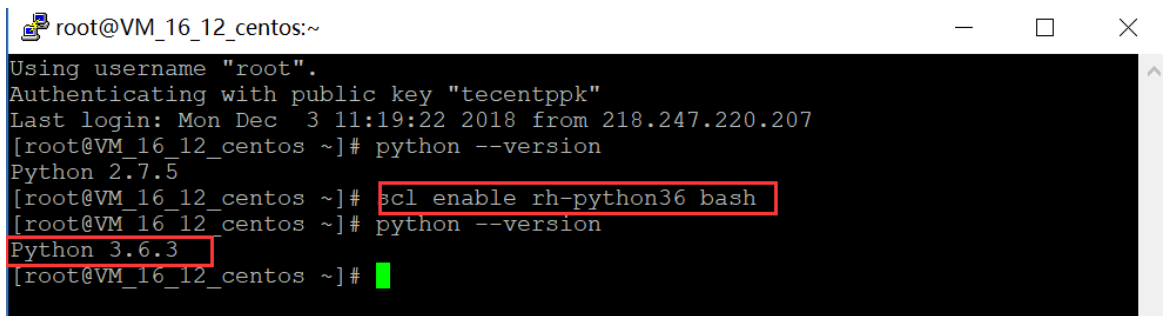
2. Python 3.6 准备

CentOS 安装 Python 3.6 见参考文档: <https://linuxize.com/post/how-to-install-python-3-on-centos-7/#enable-software-collections-scl>

本文使用了 ssh 远程连接安装, 具体方法可参考腾讯云文档:

<https://cloud.tencent.com/document/product/213/5436>

CentOS 自带 python 2.7, 由于本文代码使用了 python3, 故每次想用使用 python3 时, 先运行命令: `scl enable rh-python36 bash`



3. 爬虫及 API 选择

Android 课提供的天气 API 除提供 xml 形式返回外，还提供 JSON 格式返回，而且支持 cityname 与 citycode 两种检索方式。

为简单计，本文使用 JSON 数据，将其抓换为字符串后，存入 mysql。后期方便前台 JavaScript 解析字符串为 JSON 格式。

当然 Mysql 5.7 后，支持存 JSON 对象，可以不用这么做。

程序引入了 PyMySQL 包，安装过程见：<http://www.runoob.com/python3/python3-mysql.html>

爬虫程序如下：

```
1 #!/usr/bin/python3
2 #-*- coding: utf-8 -*-
3 import urllib.request
4 import json
5 import gzip
6 import pymysql
7 import time
8
9 cnt=1
10 # 运行99次，按需更改
11 while(cnt<100):
12     print("=====第 %s 次抓天气预报数据====="%(cnt))
13     # 打开数据库连接
14     db = pymysql.connect("localhost", "数据库用户名", "数据库密码", "数据库名称", charset='utf8' )
15     # 使用cursor()方法获取操作游标
16     cursor = db.cursor()
17     for idPri in range(1,2587):
18         sql = "SELECT * FROM city WHERE _id = %s"%(idPri)
19         try:
20             cursor.execute(sql)
21             data = cursor.fetchall()
22
23             for rec in data:
24                 cityname = rec[2]
25                 #访问的url, 其中urllib.parse.quote是将城市名转换为url的组件
26                 url = 'http://wthrcdn.etouch.cn/weather_mini?city='+urllib.parse.quote(cityname)
27
28                 time.sleep(3)#设置时间间隔为3秒
29
30                 #发出请求并读取到weather_data
31                 weather_data = urllib.request.urlopen(url).read()
32                 #将json数据转化为dict
33                 weather_dict = json.loads(weather_data)
34                 #将dict转成str
35                 weather_str = json.dumps(weather_dict,ensure_ascii=False)
36                 #将字符串数据存入数据库
37                 sql2 = "UPDATE city SET weather = '%s' WHERE _id = %s" %(weather_str,idPri)
38                 try:
39                     # 执行SQL语句
40                     cursor.execute(sql2)
41                     # 提交到数据库执行
42                     db.commit()
43                     print("%s 数据存入成功"%(idPri))
44                 except:
45                     # 发生错误时回滚
46                     db.rollback()
47                     print("%s 更新失败"%(idPri))
48             except:
49                 print ("Error: unable to fetch data %s"%(idPri))
50
51     # 关闭数据库连接
52     db.close()
53     # 4小时抓一次
54     time.sleep(14400)
55     cnt+=1
56 else:
57     print("Program stopped")
```

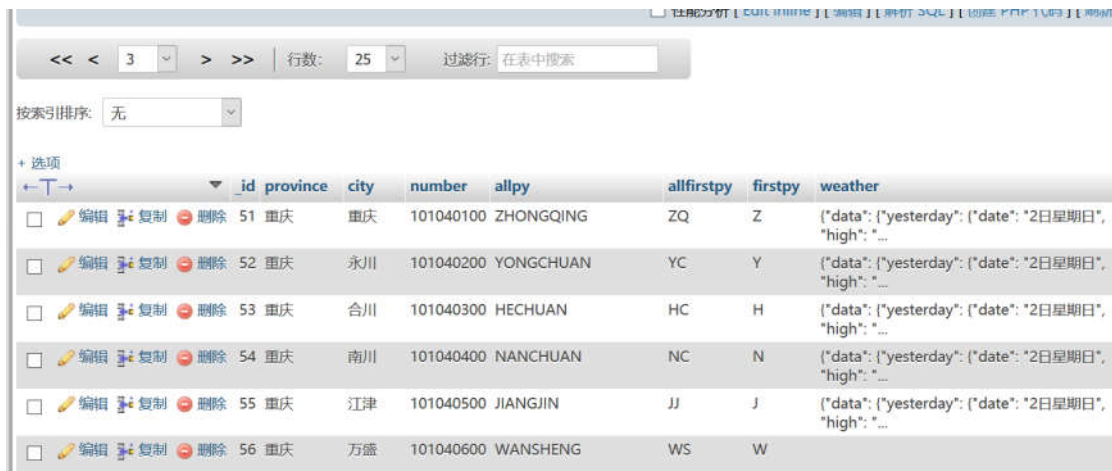
4. 持续运行

可以用宝塔上传 weather.py 文件到 /usr/pythonproject 目录，也可以 linux 命令行操作先后运行以下两个命令：

```
scl enable rh-python36 bash
```

```
nohup python -u /usr/pythonproject/weather.py > test.log 2>&1 &
```

日志文件在/root 目录下，退出 ssh 后，进程仍将持续运行；如需终止，可用 kill 命令 weather 数据持续更新中：



The screenshot shows a web interface for a database. At the top, there are navigation controls: '<< < 3 > >>' and a '行数: 25' (Rows: 25) indicator. Below this is a search bar labeled '过滤行: 在表中搜索'. A dropdown menu for '按索引排序:' (Sort by index) is set to '无' (None). The main area displays a table with columns: id, province, city, number, allpy, allfirstpy, firstpy, and weather. The table contains 6 rows of data, each with a checkbox and action icons (edit, copy, delete) to its left. The 'weather' column contains JSON-like strings representing weather data for 'yesterday'.

	id	province	city	number	allpy	allfirstpy	firstpy	weather
☐	51	重庆	重庆	101040100	ZHONGQING	ZQ	Z	{"data": {"yesterday": {"date": "2日星期日", "high": "...
☐	52	重庆	永川	101040200	YONGCHUAN	YC	Y	{"data": {"yesterday": {"date": "2日星期日", "high": "...
☐	53	重庆	合川	101040300	HECHUAN	HC	H	{"data": {"yesterday": {"date": "2日星期日", "high": "...
☐	54	重庆	南川	101040400	NANCHUAN	NC	N	{"data": {"yesterday": {"date": "2日星期日", "high": "...
☐	55	重庆	江津	101040500	JIANGJIN	JJ	J	{"data": {"yesterday": {"date": "2日星期日", "high": "...
☐	56	重庆	万盛	101040600	WANSHEG	WS	W	

参考资料：

1. city.db 转 city.sql: <http://www.sqlitetutorial.net/sqlite-dump/>
2. python 脚本抓取天气: <https://blog.csdn.net/wenqiang1208/article/details/75331732>
3. 安装 python3.6.3:
<https://linuxize.com/post/how-to-install-python-3-on-centos-7/#enable-software-collections-scl>
4. PyMySQL 安装: <http://www.runoob.com/python3/python3-mysql.html>
5. 腾讯云 ssh 远程登录服务器: <https://cloud.tencent.com/document/product/213/5436>
6. python 快速入门: <http://www.runoob.com/w3cnote/python-ten-minute-introductory-tutorial.html>
7. python 写入数据库: <https://blog.csdn.net/Oscer2016/article/details/70257956>
8. 持续运行 .py 文件: https://blog.csdn.net/qg_36962569/article/details/77455602
9. 前台 JavaScript 解析字符串为 JSON 格式: <https://www.w3cschool.cn/json/waz81pix.html>

源码：

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
import urllib.request
import json
import gzip
import pymysql
import time

cnt=1
# 运行 99 次，按需更改
while(cnt<100):
    print("===第 %s 次抓天气预报数据======"%(cnt))
    # 打开数据库连接
```

```

db = pymysql.connect("localhost", "数据库用户名", "数据库库密码", "数据库名称",
charset='utf8' )
# 使用 cursor()方法获取操作游标
cursor = db.cursor()
for idPri in range(1,2587):
    sql = "SELECT * FROM city WHERE _id = %s"%(idPri)
    try:
        cursor.execute(sql)
        data = cursor.fetchall()

        for rec in data:
            cityname = rec[2]
            #访问的 url, 其中 urllib.parse.quote 是将城市名转换为 url 的组件
            url =
            'http://wthrcdn.etouch.cn/weather_mini?city='+urllib.parse.quote(cityname)

            time.sleep(3)#设置时间间隔为 3 秒

            #发出请求并读取到 weather_data
            weather_data = urllib.request.urlopen(url).read()
            #以 utf-8 的编码方式解压数据
            weather_data = gzip.decompress(weather_data).decode('utf-8')
            #将 json 数据转化为 dict
            weather_dict = json.loads(weather_data)
            #将 dict 转成 str
            weather_str = json.dumps(weather_dict,ensure_ascii=False)

            #将字符串数据存入数据库
            sql2 = "UPDATE city SET weather = '%s' WHERE _id = %s" %(weather_str,idPri)

            try:
                # 执行 SQL 语句
                cursor.execute(sql2)
                # 提交到数据库执行
                db.commit()
                print("%s 数据存入成功"%(idPri))
            except:
                # 发生错误时回滚
                db.rollback()
                print("%s 更新失败"%(idPri))
        except:
            print ("Error: unable to fetch data %s"%(idPri))

# 关闭数据库连接

```

```
db.close()
# 4 小时抓一次
time.sleep(14400)
cnt+=1
else:
    print("Program stopped")
```