

微信公众号消息的天气预报功能

1801210687-向建宇

返回真实的天气信息的话，得有真实的数据，所以第一步要写一个爬虫定时获取天气信息，第二步使用爬取的数据来生成 api 接口，第三步使用自己的 api 接口完成微信公众号消息的天气预报功能

一、数据库爬虫

因为老师给的 api 有每分钟 300 次的限制，所以必须自己写个爬虫将数据爬到数据库内，由于 <http://t.weather.sojson.com/api/weather/city/> 这个网站已经返回给我们 json 格式的数据了，所以这边直接将它写入数据库即可

1.首先需要安装：

我的代码使用的是 Python 3 而 linux 上面默认是 python2.7，所以需要自己重新安装环境，如我云端使用的是 Anaconda 自带的 python 版本，可以自己自行下载 python 3 的版本

```
Last login: Wed Dec 5 20:16:21 2018 from 124.205.76.239
ubuntu@VM-0-2-ubuntu:~$ python --version
Python 3.6.5 :: Anaconda, Inc.
```

然后 pip install Requests,pymysql 这两个 python 包，

2.爬虫设计

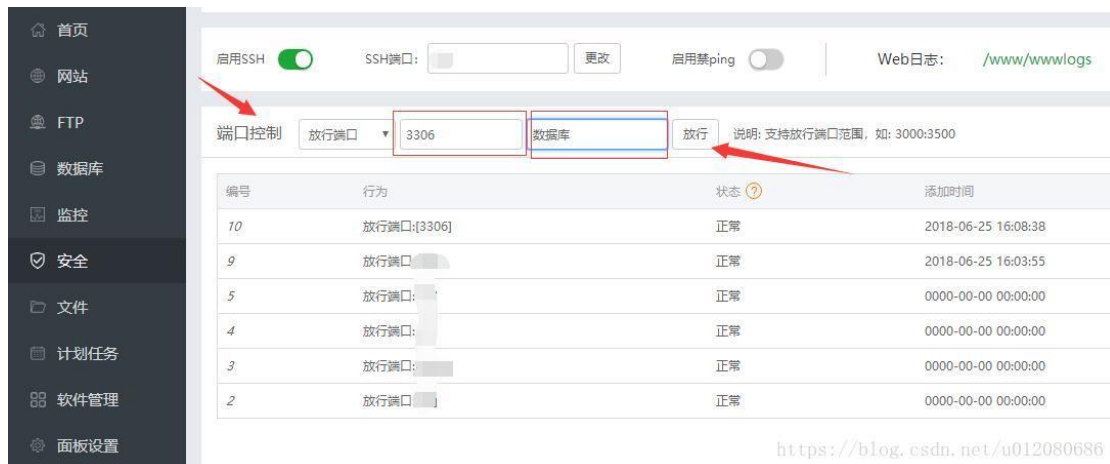
爬虫十分简单，分为三步：

(1) 连接数据库，这里可以在本地先调试好以后再上传到服务器，在本地调试时可以远程连接数据库：

在宝塔数据库页面打开权限设置，访问权限勾选：所有人



在安全页面，设置放行端口，将 3306 填入第一个对应位置，后面自己写备注，然后点击“放行”按钮即可，也可以指定范围，如 3000:3500。



数据库连接设置为：
将参数修改为你自己的参数

```
def connect_db():
    conn = pymysql.connect(host='123.207.168.242', #远程 ip 地址，传到云端前修改为 localhost
                           port=3306, #端口号
                           user='xxx', #数据库用户名
                           passwd='xxx', #数据库用户密码
                           db='sql123_207_168_', #数据库
                           charset='utf8')

# 使用 cursor() 方法获取操作游标
cursor = conn.cursor(cursor=pymysql.cursors.DictCursor)
```

(2) 从数据库查找 citycode 并更新数据库

下面代码注释已经很详细了，首先去得到 citycode 然后根据老师给的 api 接口获得 json 文件并更新就行

```
def get_info(citycode='101010100'):
    url = "http://t.weather.sojson.com/api/weather/city/" + citycode
    r = requests.get(url)
    return r.text

更新天气信息内容
def update_info():
    #连接数据库
    cursor, conn= connect_db()
    #去数据库查询所有的 citycode
    cursor.execute('select weather_code from ins_county')
    listcodes = cursor.fetchall() # 得到多行记录
    conn.commit()
    #更新数据库的 weather_info
    for citycode in listcodes:
        time.sleep(0.3) # 设置时间间隔为 0.3 秒, 因为这个接口阈值为每分钟 300 次, 也就是每
        0.2 秒一次, 这里保险点用 0.3 秒
        info = get_info(citycode['weather_code'])
```

```

# 将字符串数据存入数据库
sql = "UPDATE ins_county SET weather_info = '%s' WHERE weather_code = %s" %(info,citycode['weather_code'])
try: # 执行 SQL 语句
    cursor.execute(sql)
    # 提交到数据库执行
    conn.commit()
    print("{} 写入成功".format(citycode['weather_code']))
except:
    # 发生错误时回滚
    conn.rollback()

```

完整代码参见附录 1

(3) 设置自动执行

将 python 文件上传至服务器，并可以使用宝塔的计划任务来定时执行爬虫

写入：

```
cd /www/wwwroot/123.207.168.242_03/public;
```

```
python get_weatherInfo.py
```

上面的 cd 后面跟你爬虫的绝对路径，python 后你的文件名

注意执行前要修改 weather_info 的大小，json 数据大概 1000 字符，我把大小改为了 2000

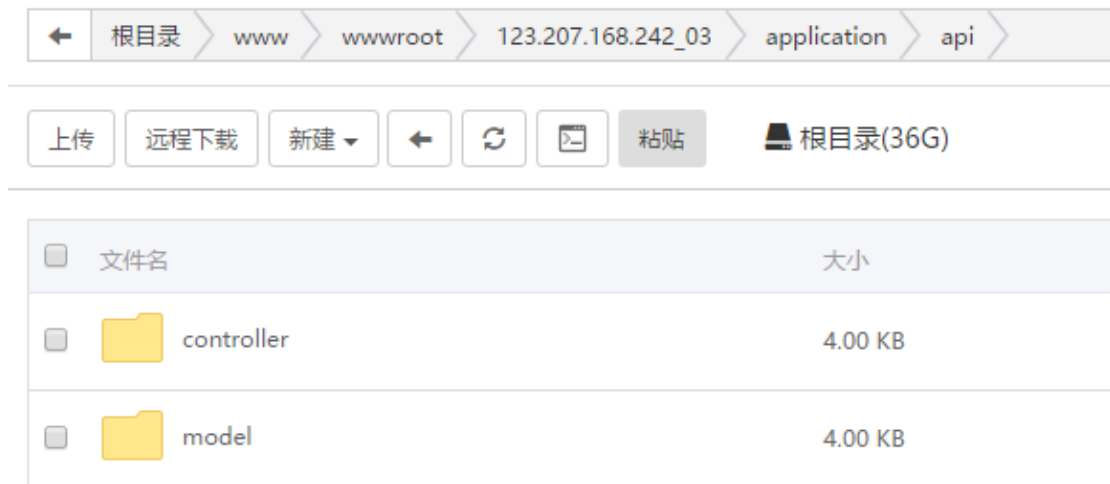
id	名称	类型	字符集	排序	是否主键	是否自增	备注	操作
1	id	bigint(22)			否	无	AUTO_INCREMENT	修改 -
2	ct_id	varchar(22)	utf8_general_ci		否	无		修改 -
3	county_id	varchar(22)	utf8_general_ci		否	无		修改 -
4	county_name	varchar(22)	utf8_general_ci		否	无		修改 -
5	weather_code	varchar(22)	utf8_general_ci		否	无		修改 -
6	weather_info	varchar(2000)	utf8_general_ci		是	NULL		修改 -
7	msg_info	varchar(120)	utf8_general_ci		是	NULL		修改 -

最后结果如图：

county_name	weather_code	weather_info	msg_info	gmt_create	gmt_modi
城市名字	城市天气编号《获取天气专用》	天气情况	城市情况	创建时间	数据修改时间
北京	101010100	<pre>{ "time": "2018-12-05 19:18:52", "cityInfo": { "city": "北京市", "cityId": "101010100", "parent": "北京", "updateTime": "19:05", "date": "20181205", "message": "Success!", "status": 200, "data": { "shidu": 67%, "pm25": 54.0, "pm10": 130.0, "quality": "良", "wendu": 0, "ganmao": "极少数敏感人群应减少户外活动", "yesterday": { "date": "04", "sunrise": "07:18", "high": "高温 4.0℃", "low": "低温 -6.0℃", "sunset": "16:50", "aqi": 81.0, "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" }, "forecast": [{ "date": "05", "sunrise": "07:19", "high": "高温 0.0℃", "low": "低温 -3.0℃", "sunset": "16:49", "aqi": 95.0, "ymd": "2018-12-05", "week": "星期三", "fx": "南风", "fl": "<3级", "type": "晴", "notice": "愿你拥有比阳光明媚的心情" }, { "date": "06", "sunrise": "07:20", "high": "高温 -2.0℃", "low": "低温 -9.0℃", "sunset": "16:49", "aqi": 54.0, "ymd": "2018-12-06", "week": "星期四", "fx": "无持续风向", "fl": "3-4级", "type": "晴", "notice": "愿你拥有比阳光明媚的心情" }, { "date": "07", "sunrise": "07:21", "high": "高温 -4.0℃", "low": "低温 -10.0℃", "sunset": "16:49", "aqi": 29.0, "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" }, { "date": "08", "sunrise": "07:22", "high": "高温 -4.0℃", "low": "低温 -10.0℃", "sunset": "16:49", "aqi": 28.0, "ymd": "2018-12-08", "week": "星期六", "fx": "西北风", "fl": "<3级", "type": "晴", "notice": "愿你拥有比阳光明媚的心情" }, { "date": "09", "sunrise": "07:23", "high": "高温 -2.0℃", "low": "低温 -9.0℃", "sunset": "16:49", "aqi": 54.0, "ymd": "2018-12-09", "week": "星期日", "fx": "南风", "fl": "<3级", "type": "晴", "notice": "愿你拥有比阳光明媚的心情" }] } } }</pre>			
海淀	101010200				
朝阳	101010300				
顺义	101010400				
怀柔	101010500				
通州	101010600				
昌平	101010700				
延庆	101010800				
丰台	101010900				

二、api 实现

现在可以直接用数据库的数据来构建 api 而摆脱阈值的束缚了
仿照 news，即在下图两个文件夹里面写一个构造器和 model



1.model -- 完整代码见附录二

主要是两个功能：通过 citycode 取得天气信息和通过城市名称取得城市 code

```
//通过城市 code 取得天气信息
public function getWeathers($id = 1)
{
    $res = Db::name('ins_county')->where('weather_code',
$id)->column('weather_info');
    return $res;
}

//通过城市名称取得城市 code
public function getCityCode($city = 1)
{
    $res = Db::name('ins_county')->where('county_name',
$city)->column('weather_code');
    return $res;
}
```

2.controller -- 完整代码见附录三

对应的在 controller 中添加对这两个功能的使用，使其成为 api 接口

```
public function read()
{
    $id = input('id');
    $model = model('Weathers');
    $data = $model->getWeathers($id);
    #这里必须先转成数组，然后再以 json 格式返回，不然 html 格式会出错，
    #导致使用这个接口获得的数据在 json_decode () 时会变成 null
    $html = json_decode($data[0]);
    return json($html);
}
```

```

}
//得到城市 id
public function getcitycode()
{
    $id = input('name');
    $model = model('Weathers');
    $data = $model->getCityCode($id);
    if ($data) {
        $code = 200;
    } else {
        $code = 404;
    }
    $data = [
        'code' => $code,
        'data' => $data
    ];
    return json($data);
}

```

3.在微信处理页面增加逻辑 -- 完整代码见附录四

先通过城市名找到对应的 citycode，再使用 citycode 找到天气信息，不过这里其实绕圈子了，可以直接根据城市名查数据库就行了，不过既然老师要求做上面两个接口，就用这两个吧，下面只简单的返回了几个信息，可自行添加

```

//通过城市名找到 citycode
$url_citycode =
'http://www.redpanda233.xyz/api/weathers/getcitycode/name/'. $content;
$html2 = json_decode(file_get_contents($url_citycode));
$citycode = $html2->data[0];

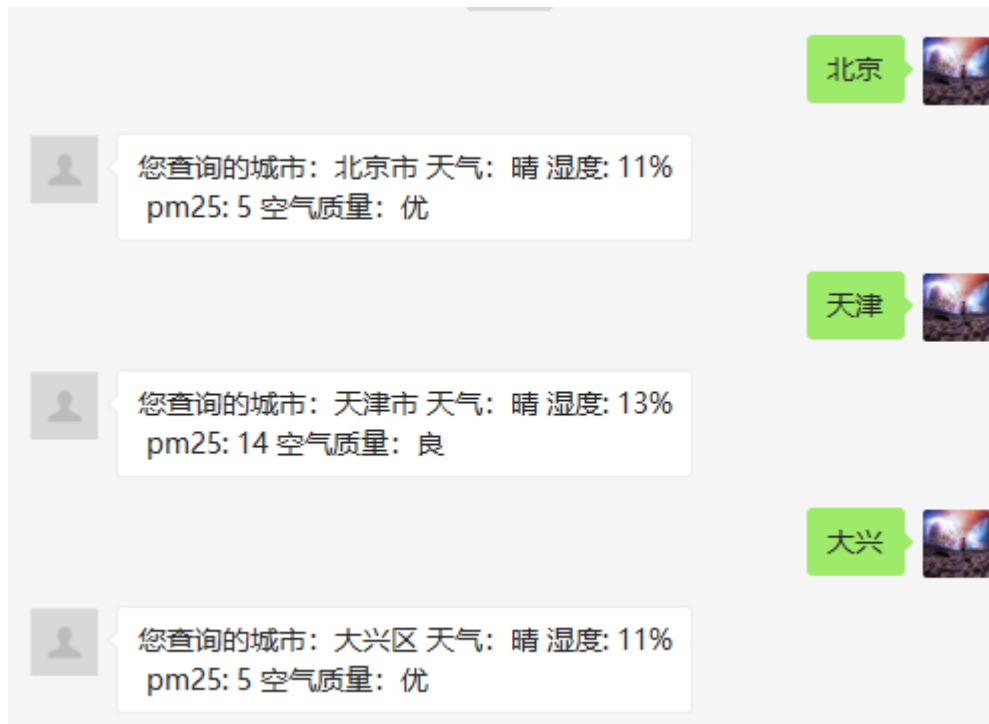
//通过 citycode 找到天气信息
$url='http://www.redpanda233.xyz/api/weathers/read/id/'. $citycode;
//json_decode 解析了 json 格式，组成字典，方便使用
$html = json_decode(file_get_contents($url));

$time =time();
$msgType ='text';

if($html->status==200) //200 表示服务器正常处理了请求
    $content = '您查询的城市：'. $html->cityInfo->city.' 天气：'. $html->data->forecast[0]->type.' 湿度：'. $html->data->shidu.' pm25:'. $html->data->pm25.' 空气质量：'. $html->data->quality;
else
    $content ='没有这个城市的编码';

```

结果如图：



这里返回的比较简单可以自行去配置页面修改下面这段代码：

```
if($html->status==200) //200 表示服务器正常处理了请求
    $content = ' 您 查 询 的 城 市 : '.$html->cityInfo->city.' 天 气 :
    '.$html->data->forecast[0]->type.' 湿 度 : '.$html->data->shidu.' pm25:
    '.$html->data->pm25.' 空气质量: '.$html->data->quality;
```

这里是返回的 json，对照着改一下上面代码就 ok 了：

```
{
  "time": "2018-12-05 19:18:52",
  "cityInfo": {
    "city": "北京市",
    "cityId": "101010100",
    "parent": "北
    京",
    "updateTime": "19:05",
    "date": "20181205",
    "message": "Success !",
    "status": 200,
    "data": {
      "shidu": "67%",
      "pm25": 54,
      "pm10": 130,
      "quality": "良",
      "wendu": "0",
      "ganmao": "极少数敏感人群应减少户外活动",
      "yesterday": {
        "date": "04",
        "sunrise": "07:18",
        "high": "高温 4.0\u00b0C",
        "low": "低温 -6.0\u00b0C",
        "sunset": "16:50",
        "aqi": 81,
        "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",
        "forecast": [
          {
            "date": "05",
            "sunrise": "07:19",
            "high": "高温 0.0\u00b0C",
            "low": "低温 -3.0\u00b0C",
            "sunset": "16:49",
            "aqi": 95,
            "ymd": "2018-12-05",
            "week": "星期三",
            "fx": "南风",
            "fl": "<3级",
            "type": "阴",
            "notice": "不要被阴云遮住好心情",
            "date": "06",
            "sunrise": "07:20",
            "high": "高温 -2.0\u00b0C",
            "low": "低温 -9.0\u00b0C",
            "sunset": "16:49",
            "aqi": 54,
            "ymd": "2018-12-06",
            "week": "星期四",
            "fx": "无持续风向",
            "fl": "3-4级",
            "type": "晴",
            "notice": "愿你拥有比阳光明媚的心情",
            "date": "07",
            "sunrise": "07:21",
            "high": "高温 -4.0\u00b0C",
            "low": "低温 -10.0\u00b0C",
            "sunset": "16:49",
            "aqi": 29,
            "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",
            "date": "08",
            "sunrise": "07:22",
            "high": "高温 -4.0\u00b0C",
            "low": "低温 -10.0\u00b0C",
            "sunset": "16:49",
            "aqi": 28,
            "ymd": "2018-12-08",
            "week": "星期六",
            "fx": "西北风",
            "fl": "<3级",
            "type": "晴",
            "notice": "愿你拥有比阳光明媚的心情",
            "date": "09",
            "sunrise": "07:23",
            "high": "高温 -2.0\u00b0C",
            "low": "低温 -9.0\u00b0C",
            "sunset": "16:49",
            "aqi": 54,
            "ymd": "2018-12-09",
            "week": "星期日",
            "fx": "南风",
            "fl": "<3级",
            "type": "晴",
            "notice": "愿你拥有比阳光明媚的心情"
          }
        ]
      }
    }
  }
}
```

附录一：（爬虫）

```
# coding = utf-8
import requests
import time #计时用的
import pymysql
"""
连接数据库
返回: cursor,conn
"""
def connect_db():
    conn = pymysql.connect(host='localhost',
                           port=3306,
                           user='sql123_207_168_',
                           passwd='xiangjianyu123',
                           db='sql123_207_168_',
                           charset='utf8')

    # 使用 cursor()方法获取操作游标
    cursor = conn.cursor(cursor=pymysql.cursors.DictCursor)

    return cursor, conn
"""
得到天气信息的 json 内容
输入: citycode - 城市的 citycode
返回: r.text - json 格式的文件
"""
def get_info(citycode='101010100'):
    url = "http://t.weather.sojson.com/api/weather/city/" + citycode
    r = requests.get(url)

    return r.text
"""
更新天气信息内容
"""
def update_info():
    #连接数据库
    cursor, conn= connect_db()

    #去数据库查询所有的 citycode
    cursor.execute('select weather_code from ins_county')

    listcodes = cursor.fetchall() # 得到多行记录
    conn.commit()

    #更新数据库的 weather_info
    for citycode in listcodes:
        time.sleep(0.3) # 设置时间间隔为 0.3 秒, 因为这个接口阈值为每分钟 300 次, 也就是每
        0.2 秒一次, 这里保险点用 0.3 秒
        info = get_info(citycode['weather_code'])
        # 将字符串数据存入数据库
```

```

        sql = "UPDATE ins_county SET weather_info = '%s' WHERE weather_code
= %s" %(info, citycode['weather_code'])

        try: # 执行 SQL 语句
            cursor.execute(sql)
            # 提交到数据库执行
            conn.commit()
            print("{} 写入成功".format(citycode['weather_code']))
        except:
            # 发生错误时回滚
            conn.rollback()
    if __name__ == '__main__':
        update_info()

```

附录二：(model)

```

<?php
namespace app\api\model;

use think\Model;
use think\Db;

class Weathers extends Model
{
    //通过城市 code 取得天气信息
    public function getWeathers($id = 1)
    {
        $res = Db::name('ins_county')->where('weather_code',
$id)->column('weather_info');
        return $res;
    }

    //通过城市名称取得城市 code
    public function getCityCode($city = 1)
    {
        $res = Db::name('ins_county')->where('county_name',
$city)->column('weather_code');
        return $res;
    }
}

```

附录三：(controller)

```

<?php
namespace app\api\controller;
header("Content-type: text/html; charset=utf-8");
use think\Controller;

```

```

class Weathers extends Controller
{
    public function read()
    {
        $id = input('id');
        $model = model('Weathers');
        $data = $model->getWeathers($id);
        $html = json_decode($data[0]);
        return json($html);
    }
    //得到城市 id
    public function getcitycode()
    {
        $id = input('name');
        $model = model('Weathers');
        $data = $model->getCityCode($id);
        if ($data) {
            $code = 200;
        } else {
            $code = 404;
        }
        $data = [
            'code' => $code,
            'data' => $data
        ];
        return json($data);
    }
}

```

附录四：（公众号配置接口）

```

<?php
    //获得参数 signature nonce token timestamp echostr
    $nonce      = $_GET['nonce'];
    $token      = 'haha';
    $timestamp  = $_GET['timestamp'];
    $echostr    = $_GET['echostr'];
    $signature  = $_GET['signature'];
    //形成数组，然后按字典序排序
    $array = array();
    $array = array($nonce, $timestamp, $token);
    sort($array);
    //拼接成字符串,sha1 加密，然后与 signature 进行校验
    $str = sha1( implode( $array ) );
    if( $str == $signature && $echostr ){

```

```

//第一次接入 weixin api 接口的时候
echo $echostr;
exit;
}
else{
//1.获取到微信推送过来 post 数据 (xml 格式)
$postArr = $GLOBALS['HTTP_RAW_POST_DATA'];
//2.处理消息类型, 并设置回复类型和内容
$postObj = simplexml_load_string( $postArr );
//判断该数据包是否是订阅的事件推送
if( strtolower( $postObj->MsgType) == 'event'){
//如果是关注 subscribe 事件
if( strtolower($postObj->Event == 'subscribe') ){
//回复用户消息(纯文本格式)
$user = $postObj->FromUserName;
$fromUser = $postObj->ToUserName;
$time = time();
$msgType = 'text';
$content = '欢迎关注我们的微信公众账号, 此公众号为测试公众号!
' . $postObj->FromUserName . '-' . $postObj->ToUserName;
$template = "<xml>
                <ToUserName><![CDATA[%s]]></ToUserName>

<FromUserName><![CDATA[%s]]></FromUserName>
                <CreateTime>%s</CreateTime>
                <MsgType><![CDATA[%s]]></MsgType>
                <Content><![CDATA[%s]]></Content>
            </xml>";
    $info = sprintf($template, $user, $fromUser, $time, $msgType,
$content);
    echo $info;
        }
    }
    else if(strtolower($postObj->MsgType) == 'text')
    {
$content = $postObj->Content;
$user=$postObj->FromUserName;
$fromUser=$postObj->ToUserName;
//通过城市名找到 citycode
$url_citycode =
'http://www.redpanda233.xyz/api/weathers/getcitycode/name/' . $content;
$html2 = json_decode(file_get_contents($url_citycode));
$citycode = $html2->data[0];

```

```
//通过 citycode 找到天气信息
$url='http://www.redpanda233.xyz/api/weathers/read/id/'.$citycode;
//json_decode 解析了 json 格式，组成字典，方便使用
$html = json_decode(file_get_contents($url));

$time =time();
$msgType ='text';

if($html->status==200) //200 表示服务器正常处理了请求
    $content =' 您 查 询 的 城 市 : ' . $html->cityInfo->city.' 天 气 : ' . $html->data->forecast[0]->type.' 湿 度 : ' . $html->data->shidu.' pm25: ' . $html->data->pm25.' 空气质量: ' . $html->data->quality;
else
    $content ='没有这个城市的编码';
    //$content =$html->cityInfo->city;

$template ="<xml>
    <ToUserName><![CDATA[%s]]></ToUserName>
    <FromUserName><![CDATA[%s]]></FromUserName>
    <CreateTime>%s</CreateTime>
    <MsgType><![CDATA[%s]]></MsgType>
    <Content><![CDATA[%s]]></Content>
</xml>";

$info = sprintf($template, $toUser, $fromUser, $time, $msgType, $content);
echo $info;
}

}
```