

## 附件 2

### 基于位置的天气实况服务接口使用说明

根据《基于位置的天气实况服务策略（V1.0）》，国家气象信息中心基于“天擎”开发了基于位置的天气实况服务接口（简称“位置服务接口”），各级气象部门可通过调用位置服务接口获取统一的天气实况信息，确保全国各类气象服务系统提供的同一时刻、同一地点的实况信息一致。

位置服务接口使用说明如下：

#### 1. 适用范围

位置服务接口部署在公共云上，支持全国互联网的气象数据服务系统基于用户位置获取天气实况数据，包括网站、APP、小程序等。

#### 2. 使用流程

在中国气象数据网(<http://data.cma.cn>)上申请 API 账户，审核通过后，参考调用示例（附录 1）在数据服务系统的开发程序中，调用位置服务接口，获取天气实况数据。

#### 3. 接口描述

##### （1）接口功能

位置服务接口支持两种应用场景：“点”的应用场景和“线”的应用场景。根据点或线的经纬度信息，获取气温、相对湿度、风速、风向、天气现象、能见度、总云量、海表温度以及降水等气象要素数据。要素清单详见表 1。

表 1 位置服务接口要素清单

要素	服务要素代码	单位	精度
温度	TEM	℃	0.1
相对湿度	RHU	%	1
风速	WINS	m/s	0.1
风向	WIND	°	1
天气现象	WEA	—	—
能见度	VIS	m	50
总云量	TCDC	%	1
海表温度	SST	℃	0.1
1 小时累计降水（10 分钟）	PRE_1H	mm	0.1
3 小时累计降水（10 分钟）	PRE_3H	mm	0.1
6 小时累计降水（10 分钟）	PRE_6H	mm	0.1
12 小时累计降水（10 分钟）	PRE_12H	mm	0.1
24 小时累计降水（10 分钟）	PRE_24H	mm	0.1

## （2）接口形式

位置服务接口采用 RESTful 风格发布。

### ● 接口地址

<https://music.data.cma.cn/lbs/api?>

### ● 接口参数

位置服务接口参数见表 2，所有参数均为必选。

表 2 位置服务接口参数表

参数代码	中文名	赋值格式	赋值示例
elements	要素代码	1 个或多个，多个以逗号(,)分隔	TEM, RHU, WINS, WIND, WEA, VIS, TCDC, SST, PRE_1H, PRE_3H, PRE_6H, PRE_12H, PRE_24H
interface Id	接口 ID	固定值	getWeatherLBS

参数代码	中文名	赋值格式	赋值示例
lat	纬度	1 个或多个, 多个以逗号(,)分隔; 单位: 度, 范围: 北纬 15-60 度	一个点: Lat=39.95&lon=116.33 表示纬度和经度为(39.95, 116.33)的点; 一条线:
lon	经度	1 个或多个, 多个以逗号(,)分隔; 单位: 度, 范围: 东经 70-140 度	lat=39.95, 40.05, 40.15, 40.25&lon=116.33, 116.43, 116.53, 116.63 表示纬度和经度为 (39.95, 116.33) (40.05, 116.43) (40.15, 116.53) (40.25, 116.63)的线;
timestamp	时间戳	当前时间 (北京时), 用数字表示, 即自 1970 年 1 月 1 日 (00:00:00 GMT) 以来的毫秒数	<u>1601007132000 表示</u> <u>2020-09-25 12:12:12 (北京时)</u>
apikey	API 账户	字符串 (申请时分配)	WeatherCom
sign	认证标签	用户程序生成: 在检索参数字符串基础上, 补充 “&pwd=账号密码”, 然后调用 MD5 加密后生成。其中账户密码是申请时分配的。	检索参数: elements=TEM, RHU, WINS, WIND, WEA, VIS, T CDC, SST, PRE_1H, PRE_3H, PRE_6H, PRE_12H , PRE_24H&interfaceId=getSurfEleInLocationByTime&lat=3.51&lon=73.33&timestamp=1601026069531&apikey=AxEkluely201 MD5 加密前信息: elements=TEM, RHU, WINS, WIND, WEA, VIS, T CDC, SST, PRE_1H, PRE_3H, PRE_6H, PRE_12H , PRE_24H&interfaceId=getSurfEleInLocationByTime&lat=3.51&lon=73.33&timestamp=1601026069531&apikey=WeatherCom &pwd=WeatherComPwd MD5 加密后信息: 936D7978111D4E801FBA8917C5763B0

### (3) 返回结果

接口返回结果为 json 格式, 参见示例 (见附录 2) 描述, 各要素项说明如表 3。

表 3 返回数据项清单

返回项	中文描述	值类型	备注
returnCode	返回码 (取值含义见表 4)	Int	调用状态
returnMessage	返回状态信息	String	
rowCount	行数	Int	
colCount	列数	Int	
requestTime	请求时间	String	
responseTime	响应时间	String	
takeTime	耗时 (秒)	Float	
fieldNames	要素名称	String	
fieldUnits	要素单位	String	
DATETIME	资料时间	String	要素数据, 按用户所选要素项返回
LAT	纬度	Float	
LON	经度	Float	
TEM	气温	String	
RHU	相对湿度	String	
WINS	风速	String	
WIND	风向	String	
WEA	天气现象	String	
VIS	能见度	String	
TCDC	总云量	String	
SST	海表温度	String	
PRE_1H	1 小时累计降水 (10 分钟)	String	
PRE_3H	3 小时累计降水 (10 分钟)	String	
PRE_6H	6 小时累计降水 (10 分钟)	String	

返回项	中文描述	值类型	备注
PRE_12H	12 小时累计降水（10 分钟）	String	
PRE_24H	24 小时累计降水（10 分钟）	String	

表 4 位置服务接口返回码清单

序号	返回码	描述信息
1.	0	成功
2.	-1	成功，没有获取到数据
3.	-1001	无效的 apikey
4.	-1002	apikey 验证失败
5.	-2001	接口错误
6.	-3001	参数赋值错误
7.	-3004	经纬度超过服务范围
8.	-3005	检索要素未提供服务
9.	-6001	数据库连接失败, 服务器无法连接
10.	-7001	数据库 SQL 操作错误
11.	-10001	其他异常错误

## 4. 调用示例

### （1）调用程序

以 JAVA 为例，调用程序详见附录 1。

### （2）返回结果

返回结果示例详见附录 2。

## 附录 1 位置服务接口调用程序示例

```
package cma.music.location;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
/** * 基于位置的实况数据公众服务示例
 * @author
 */
public class LbsDemo {
    /**
     * main方法,程序入口
     * @param args
     */
    public static void main(String[] args) {
        /** 服务地址 */
        String serviceUrl = "https://music.data.cma.cn/lbs/api";

        /** 1. 定义服务完整路径对象,并拼接基础路径*/
        StringBuffer url = new StringBuffer();
        url.append(serviceUrl).append("?");
        /** 2. 拼接检索参数 */
        /** 2.1 检索要素 */
        String eles =
"TEM,RHU,WINS,WIND,WEA,VIS,TCDC,SST,PRE_1H,PRE_3H,PRE_6H,PRE_12H,PRE_24H";
        url.append("elements=").append(eles).append("&");
        /** 2.2 接口ID */
        String interfaceId = "getSurfEleInLocationByTime";
        url.append("interfaceId=").append(interfaceId).append("&");
        /** 2.3 经纬度点或线*/
        String lat = "3.51";
        String lon = "73.33";
        //String lat = "3.51,3.61,3.71";
        //String lon = "73.33,73.66,74.65";
        url.append("lat=").append(lat).append("&");
        url.append("lon=").append(lon).append("&");
        /** 2.4 时间戳 */
        String timestamp = String.valueOf(System.currentTimeMillis());
        url.append("timestamp=").append(timestamp).append("&");
    }
}
```

```

    /* 2.5 apikey */
    String apikey = "AxEkluuey201";
    url.append("apikey=").append(apikey).append("&");
    /* 2.6 生成sign */
    String pwd = "*****"; //账号密码
    String paramStr = url.substring(url.indexOf("?")+1, url.length());
    paramStr += "pwd=" + pwd;
    String sign = getSign(paramStr);
    url.append("sign=").append(sign);
    /* 3. 调用接口 */
    StringBuffer result = new StringBuffer();
    int iRet = sendGet(url.toString(),result);
    if (iRet == 0) { //http连接正常
        System.out.print("HTTP Response Result:" + result);
    } else {
        System.out.print("HTTP Connect Exception Error:" + result);
    }
}
/**
 * 发送数据获取请求
 * @param url 请求url地址
 */
private static int sendGet(String url, StringBuffer result) {
    int ret = -1;
    BufferedReader in = null;
    try {
        URL realUrl = new URL(url);
        // 打开和URL之间的连接
        URLConnection connection = realUrl.openConnection();
        // 设置通用的请求属性
        connection.setRequestProperty("accept", "*/*");
        connection.setRequestProperty("connection", "Keep-Alive");
        connection.setRequestProperty("Accept-Charset", "utf-8");
        connection.setRequestProperty("user-agent", "Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1;SV1)");
        // 建立实际的连接
        connection.connect();
        // 定义 BufferedReader输入流来读取URL的响应
        in = new BufferedReader(new
InputStreamReader(connection.getInputStream(), "utf-8"));
        String line;
        while ((line = in.readLine()) != null) {
            result.append(line);
        }
    }
}

```

```

        ret = 0;
    } catch (Exception e) {
        result.append(e.getMessage());
        e.printStackTrace();
    } finally { // 使用finally块来关闭输入流
        try {
            if (in != null) {
                in.close();
            }
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
    return ret;
}

public static String getSign(String paramString) {
    String sign = "";
    //进行MD5运算
    MessageDigest md5;
    try {
        md5 = MessageDigest.getInstance("MD5");
        md5.reset();
        byte[] byteArray = md5.digest(paramString.getBytes("utf-8"));
        StringBuffer md5StrBuff = new StringBuffer();
        //将加密后的byte数组转换为十六进制的字符串, 否则的话生成的字符串会乱码
        for (int i = 0; i < byteArray.length; i++) {
            if (Integer.toHexString(0xFF & byteArray[i]).length() == 1){
                md5StrBuff.append("0").append(Integer.toHexString(0xFF &
byteArray[i]));
            } else {
                md5StrBuff.append(Integer.toHexString(0xFF &
byteArray[i]));
            }
        }
        sign = md5StrBuff.toString().toUpperCase();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }

    return sign;
}
}

```



## 附录 2 位置服务接口返回结果示例

```
{
  "returnCode": 0,
  "returnMessage": "query success",
  "rowCount": "1",
  "colCount": "8",
  "requestTime": "2019-09-20 02:57:48",
  "responseTime": "2019-09-20 02:57:48",
  "takeTime": "0.01",
  "fieldNames": "温度 降水量 相对湿度 风速 风向 天气现象",
  "fieldUnits": "℃ mm % m/s ° -",
  "DS": [{
    "DATETIME": "201909181020",
    "LAT": "3.51",
    "LON": "73.33",
    "TEM": "29",
    "RHU": "77",
    "WINS": "6.0",
    "WIND": "38.0",
    "VIS": "10.0",
    "WEA": "9999",
    "TCDC": "10.0",
    "SST": "10.0",
    "PRE_1H": "10.0",
    "PRE_3H": "10.0",
    "PRE_6H": "10.0",
    "PRE_12H": "10.0",
    "PRE_24H": "10.0"
  }, {
    "DATETIME": "201909181020",
    "LAT": "3.61",
    "LON": "74.33",
    "TEM": "29",
    "RHU": "77",
    "WINS": "6.0",
    .....
    "PRE_6H": "10.0",
    "PRE_12H": "10.0",
    "PRE_24H": "10.0"
  }, .....]
}
```