|  |
| --- |
| **ChinaWeatherAPI产品使用说明书** |
|  |
| 中国气象局公共气象服务中心 |
| 2015/7/15 |

版本号：2.6.8

|  |
| --- |
|  |

更改记录

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 内容 |
| 2.3.9 | 2014/3/18 | 更改parseWarning接口输入参数的描述。 |
| 2.3.9 | 2014/4/17 | 更改uploadScene接口输入参数的描述。 |
| 2.4.1 | 2014/4.28 | 更改推送接口方法 |
| 2.4.2 | 2014/6/11 | 增加getWeatherFactInfo接口返回的数据。 |
| 2.4.4 | 2014/7/3 | 更改uploadComment接口输入参数的描述 |
| 2.4.8 | 2014/9/5 | 修正getWeathers接口异常bug |
| 2.4.9 | 2014/9/9 | 增加getWeather2接口 |
| 2.4.9 | 2014/9/18 | 更改getAirQualityInfo接口输出参数的描述 |
| 2.5.0 | 2014/9/24 | 修改推送功能，以应用到多客户端调用。 |
| 2.5.1 | 2014/9/26 | 增加调查问卷上传接口uploadQuestionnaire |
| 2.5.2 | 2014/10/15 | 修改推送功能，支持多应用共享长连接。去掉setPushAdd接口。 |
| 2.5.3 | 2014/10/17 | 修改Notifier接口中id类型为int |
| 2.5.3 | 2014/10/20 | 更新天气现象编码表 |
| 2.5.4 | 2014/10/27 | 增加getWeathers2接口 |
| 2.5.5 | 2014/12/5 | 修正getScene接口获取图片路径不正确的bug |
| 2.5.6 | 2014/12/15 | 修正getPortrait接口获取图片路径不正确的bug |
| 2.5.7 | 2014/12/24 | 增加接口getAds，修正上传图片、声音文件大小限制 |
| 2.5.8 | 2014/12/19 | 增加uploadCrowdWeather接口，修改getScene接口返回数据 |
| 2.5.9 | 2015/1/7 | Weather增加getAirQualityForecast接口 |
| 2.6.0 | 2015/1/22 | CommonAPI增加getNotice接口 |
| 2.6.1 | 2015/1/29 | 1. bindTenc等绑定腾讯微博修改为绑定QQ， 2. 去掉Qzone类接口， 3. 分享到腾讯微博接口修改为share2TencWeibo 4. 第三方登录接口参数type=1修改为绑定QQ |
| 2.6.2 | 2015/2/6 | 1. 增加CrowdWeatherAPI 2. uploadCrowdWeather接口修改为uploadMyWeather 3. 修改getNotice接口 |
| 2.6.3 | 2015/2/10 | 1. 增加VideoAPI  2. 修改getMyWeather接口参数类型  3. 修改uploadMyWeather接口描述 |
| 2.6.4 | 2015/3/11 | UserAPI增加addScore、getScoreHistory接口 |
| 2.6.5 | 2015/3/27 | 1. 去掉uploadMyWeather用户登录限制 2. 修改getStatWeather、getCrowdWeather接口及描述 |
| 2.6.6 | 2015/3/31 | 1.修改getCrowdWeather接口返回数据 |
|  | 2015/4/1 | 1.修改getCrowdWeather接口返回数据 |
| 2.6.7 | 2015/4/15 | 1. CommonAPI增加setDailyPush接口 |
| 2.6.8 | 2015/5/6 | 1. 去掉分享功能 2. 修改thirdPartyLogin接口 3. 修改addSharedInfo接口描述 4. 修改TrafficStatsService产生NullPointerException的bug |

目录

[一、 产品概述 5](#_Toc418677042)

[二、 使用说明 5](#_Toc418677043)

[三、 调用规范 6](#_Toc418677044)

[获取AppKey 6](#_Toc418677045)

[运行环境 6](#_Toc418677046)

[导入jar 6](#_Toc418677047)

[配置AndroidManifest.xml 6](#_Toc418677048)

[四、 使用方法 8](#_Toc418677049)

[五、 服务内容 8](#_Toc418677050)

[5.1. 天气相关接口（WeatherAPI） 8](#_Toc418677056)

[5.1.1. getGeo 8](#_Toc418677057)

[5.1.2. getGeoByIp 10](#_Toc418677058)

[5.1.3. searchCity 10](#_Toc418677059)

[5.1.4. searchArea 13](#_Toc418677060)

[5.1.5. getAlmanac 15](#_Toc418677061)

[5.1.6. getHotCity 16](#_Toc418677062)

[5.1.7. getHistoryWeather 17](#_Toc418677063)

[5.1.8. getNewsList 18](#_Toc418677064)

[5.1.9. getNewsContent 20](#_Toc418677065)

[5.1.10. getWeather 21](#_Toc418677066)

[5.1.11. getWeathers 21](#_Toc418677067)

[5.1.12. getWeather2 22](#_Toc418677068)

[5.1.13. getWeathers2 22](#_Toc418677069)

[5.1.14. Weather 23](#_Toc418677070)

[5.1.15. getWeatherMedia 38](#_Toc418677071)

[5.1.16. WeatherMedia 39](#_Toc418677072)

[5.1.17. parseWeather 43](#_Toc418677073)

[5.1.18. parseWindForce 43](#_Toc418677074)

[5.1.19. parseWindDirection 44](#_Toc418677075)

[5.1.20. parseAQI 44](#_Toc418677076)

[5.1.21. parseWarning 45](#_Toc418677077)

[5.2. 用户相关接口（UserAPI） 46](#_Toc418677078)

[5.2.1. register 46](#_Toc418677079)

[5.2.2. login 47](#_Toc418677080)

[5.2.3. logout 48](#_Toc418677081)

[5.2.4. isExpired 48](#_Toc418677082)

[5.2.5. changePassword 48](#_Toc418677083)

[5.2.6. getUserInfo 49](#_Toc418677084)

[5.2.7. updateUserInfo 50](#_Toc418677085)

[5.2.8. bindingEmail 51](#_Toc418677086)

[5.2.9. uploadPortrait 51](#_Toc418677087)

[5.2.10. uploadLocalPortrait 52](#_Toc418677088)

[5.2.11. getPortrait 53](#_Toc418677089)

[5.2.12. bookCity 54](#_Toc418677090)

[5.2.13. unBookCity 55](#_Toc418677091)

[5.2.14. bookIndex 55](#_Toc418677092)

[5.2.15. unBookIndex 56](#_Toc418677093)

[5.2.16. thirdPartyLogin 57](#_Toc418677094)

[5.2.17. addScore 58](#_Toc418677095)

[5.2.18. getScoreHistory 59](#_Toc418677096)

[5.3. 增值服务相关接口（VASAPI） 60](#_Toc418677097)

[5.3.1. getVAS 60](#_Toc418677098)

[5.3.2. subscribeVAS 63](#_Toc418677099)

[5.3.3. unsubscribeVAS 64](#_Toc418677100)

[5.3.4. subscribedVAS 64](#_Toc418677101)

[5.4. 实景天气相关接口（SceneAPI） 65](#_Toc418677102)

[5.4.1. uploadScene 65](#_Toc418677103)

[5.4.2. getScene 67](#_Toc418677104)

[5.4.3. uploadComment 70](#_Toc418677105)

[5.4.4. getComment 72](#_Toc418677106)

[5.4.5. getComments 74](#_Toc418677107)

[5.4.6. uploadLaud 75](#_Toc418677108)

[5.4.7. inspectLauded 76](#_Toc418677109)

[5.5. 真人天气秀相关接口（ShowAPI） 77](#_Toc418677110)

[5.5.1. getShowTypes 77](#_Toc418677111)

[5.5.2. getShowList 78](#_Toc418677112)

[5.5.3. getShowDetail 79](#_Toc418677113)

[5.5.4. updateShow 80](#_Toc418677114)

[5.6. 众筹天气相关接口（CrowdWeatherAPI） 81](#_Toc418677115)

[5.6.1. uploadMyWeather 81](#_Toc418677116)

[5.6.2. getMyWeather 82](#_Toc418677117)

[5.6.3. deleteMyWeather 83](#_Toc418677118)

[5.6.4. getStatWeather 84](#_Toc418677119)

[5.6.5. getCrowdWeather 85](#_Toc418677120)

[5.7. 微视频相关接口（VideoAPI） 86](#_Toc418677121)

[5.7.1. uploadVideo 86](#_Toc418677122)

[5.7.2. getNewVideos 88](#_Toc418677123)

[5.7.3. getVideos 89](#_Toc418677124)

[5.7.4. getMyVideos 90](#_Toc418677125)

[5.7.5. getAnchors 92](#_Toc418677126)

[5.7.6. lottery 93](#_Toc418677127)

[5.7.7. uploadWish 93](#_Toc418677128)

[5.8. 普通接口（CommonAPI） 94](#_Toc418677129)

[5.8.1. reportError 94](#_Toc418677130)

[5.8.2. uploadFeedBack 95](#_Toc418677131)

[5.8.3. updateVersion 96](#_Toc418677132)

[5.8.4. updatePlugin 97](#_Toc418677133)

[5.8.5. getSplash 98](#_Toc418677134)

[5.8.6. getAds 99](#_Toc418677135)

[5.8.7. uploadQuestionnaire 101](#_Toc418677136)

[5.8.8. getNotice 101](#_Toc418677137)

[5.8.9. setDailyPushTime 102](#_Toc418677138)

[5.9. 信息统计 103](#_Toc418677139)

[5.9.1. StatAgent 103](#_Toc418677140)

[5.10. 推送服务 109](#_Toc418677141)

[5.10.1. PushAgent 109](#_Toc418677142)

[5.10.2. Notifier 111](#_Toc418677143)

[六、 图片下载 112](#_Toc418677144)

[6.1. 原始图像下载 112](#_Toc418677146)

[6.2. 固定分辨率图像下载 112](#_Toc418677147)

[6.3. 固定分辨率宽度图像下载 112](#_Toc418677148)

[6.4. 固定分辨率高度图像下载 112](#_Toc418677149)

[七、 音频流下载 113](#_Toc418677150)

[7.1. 原始音频流下载 113](#_Toc418677152)

[八、 编码表 113](#_Toc418677153)

[**天气现象编码表** 113](#_Toc418677154)

[**精细化预报天气现象编码表** 114](#_Toc418677155)

[**风力风向编码表** 115](#_Toc418677156)

[**预警信号类别编码表** 115](#_Toc418677157)

[**预警信号级别编码表** 115](#_Toc418677158)

[九、 数据更新频率表 116](#_Toc418677159)

# 产品概述

ChinaWeatherAPI接口(简称”API”接口) 是中国气象局推出的气象服务的手机客户端API，供手机客户端快速开发使用。

# 使用说明

接口为HTTP请求的封装，提供内容包括GEO、天气实况、预报、预警、资讯、增值服务、推送服务、用户注册登录、集成第三方分享、用户信息收集等内容。

# 调用规范

## 获取AppKey

请联系中国气象局获取相关AppKey

## 运行环境

本接口需要android sdk 1.6及以上版本

## 导入jar

下载最新版本的jar包，Eclipse用户右键您的工程根目录，选择Properties —>Java Build Path —>libraries，然后点击Add External JARs…选择指向jar的路径，点击 OK，即导入成功。

## 配置AndroidManifest.xml

* 添加应用程序的AppKey

需要先添加应用程序获得的AppId，将AppId添加到AndroidManifest.xml的meta-data里。

例如：<meta-data android:name="WEATHER\_APPKEY" android:value="unoernsswfd139xk"/>

* 添加应用程序的osVersion

将osVersion添加到AndroidManifest.xml的meta-data里。

例如：<meta-data android:name="OSVERSION" android:value="android"/>

osVersion用于区分不同操作系统

* 添加应用程序的softwareSrc

将softwareSrc添加到AndroidManifest.xml的meta-data里。

例如：<meta-data android:name="SOFTWARESRC" android:value="ChinaWeather"/>

softwareSrc用于区分手机发放渠道

* 添加应用程序的versionType

将osVersion添加到AndroidManifest.xml的meta-data里。

例如：<meta-data android:name="VERSIONTYPE" android:value="general"/>

versionType 为客户端版本类型，区分定制版和省级版，省级版为：general，定制版如联想定制客户端为：lenovo。

* 添加应用程序的category

将category添加到AndroidManifest.xml的meta-data里。

例如：<meta-data android:name="CATEGORY" android:value="mobile"/>

category为客户端设备类型，共四种，mobile（移动终端）, smartTV（智能电视）, PC（个人PC），watch（手表）

* 添加service

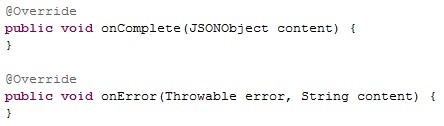
<service android:name="cn.com.weather.util.TrafficStatsService" />

* 添加权限android.permission.INTERNET
* 添加权限android.permission.ACCESS\_WIFI\_STATE
* 添加权限android.permission.ACCESS\_NETWORK\_STATE
* 添加权限android.permission.WRITE\_EXTERNAL\_STORAGE
* 添加权限android.permission.READ\_PHONE\_STATE
* 添加权限android.permission.GET\_TASKS

# 使用方法

1.所有带有context参数的接口（特殊接口有标注）:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| context | Context | true | android上下文对象 |

2.根据接口返回类型重写AsyncResponseHandler中的onComplete、onError方法，例如：getGeo接口，接口返回类型为JSONObject，则重写

# 服务内容



## 天气相关接口（WeatherAPI）

### getGeo

**接口描述：** 根据经纬度获取城市信息

**接口返回类型：**JSONObject

**输入示例：**

String longitude = "114.3937409";

String latitude = "30.513565";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| longitude | String | true | GPS经度 |
| latitude | String | true | GPS纬度 |

**输出示例：**

{

"msg":"success",

"status":"0",

"geo":{

"province":"北京",

"id":"101010100014",

"level":"4",

"district":"北京",

"station":"东花市",

"nation":"中国",

"city":"东城区"

}

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| geo | JSONObject | 获取数据内容 |
| msg | String | 请求信息描述  success表示成功  40001：未找到数据或者数据错误  40002：鉴权失败  40003：经纬度参数错误  40004：服务器内部错误 |
| status | String | 请求状态  -1表示系统繁忙  0表示请求数据成功  1表示请求数据失败 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | String | 12位区域ID |
| level | String | 城市级别 |
| station | String | 乡镇/旅游景点 |
| city | String | 城市 |
| district | String | 市 |
| province | String | 省 |
| nation | String | 国家 |

### getGeoByIp

**接口描述：**根据IP获取城市ID

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"ip":"123.108.212.157",

"cityId":"101010100"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| ip | String | ip地址 |
| cityId | String | 城市ID |

### searchCity

**接口描述：**城市搜索

**接口返回类型：**JSONObject

**输入示例：**

String nameZh = "";

String nameEn = "";

String areaId = "";

String postCode = "";

String telCode = "";

String keyWord = "hejian";

int start = 0;

int limit = 10;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| nameZh | string | false | 城市中文名称 |
| nameEn | string | false | 城市英文名称 |
| areaId | string | false | 区域编号 |
| postCode | string | false | 邮编 |
| telCode | string | false | 区号 |
| keyWord | string | false | 关键字，如果传递了该关键字，则不使用上面的4个参数进行过滤，又程序自动判断采用哪个字段进行过滤（如果全为数字，则使用areaId、postCode、telCode分别进行筛选，如果全为字母，则使用nameEn进行模糊匹配，如果有汉字，则使用nameZh进行模糊匹配） |
| start | int | true | 每页起始条数 |
| limit | int | true | 每页显示多少条 |

**输出示例：**

{

"data":{

"records":[

{

"aliasEn":"",

"aliasZh":"",

"areaId":101090714,

"continentEn":"asia",

"continentZh":"亚洲",

"districtEn":"cangzhou",

"districtZh":"沧州",

"elevation":13,

"latitude":38.446624,

"level":3,

"longitude":116.099518,

"nameEn":"hejian",

"nameZh":"河间",

"nationEn":"china",

"nationZh":"中国",

"postcode":"062450",

"provEn":"hebei",

"provZh":"河北",

"telCode":"0317",

"timezone":-8

},

{

"aliasEn":"",

"aliasZh":"",

"areaId":101271004,

"continentEn":"asia",

"continentZh":"亚洲",

"districtEn":"luzhou",

"districtZh":"泸州",

"elevation":248,

"latitude":28.811517,

"level":3,

"longitude":105.831575,

"nameEn":"hejiang",

"nameZh":"合江",

"nationEn":"china",

"nationZh":"中国",

"postcode":"646200",

"provEn":"sichuan",

"provZh":"四川",

"telCode":"0830",

"timezone":-8

}

],

"total":2684

},

"msg":"获取成功",

"status":"SUCCESS",

"timestamp":"2013-08-12 10:49:16"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | string | SUCCESS成功/FAILURE失败 |
| msg | string | 服务器返回信息 |
| timestamp | string | 时间，格式yyyy-MM-dd HH:mm:ss |
| data | JSONObject | 返回数据 |
| records | JSONArray | 查询记录的集合 |
| total | long | 总共条数 |

records属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| aliasEn | string | 用户ID。 |
| aliasZh | string | 图片标识。 |
| areaId | long | 9位城市ID |
| continentEn | string | 英文州名 |
| continentZh | string | 中文州名 |
| districtEn | string | 英文区名 |
| districtZh | string | 中文区名 |
| elevation | long | 海拔 |
| latitude | double | 纬度 |
| level | long | 城市级别 |
| longitude | double | 经度 |
| nameEn | string | 英文地名 |
| nameZh | string | 中文地名 |
| nationEn | string | 英文国家名 |
| nationZh | string | 中文国家名 |
| postcode | string | 邮编 |
| provEn | string | 英文省名 |
| provZh | string | 中文省名 |
| telCode | string | 电话区号 |
| timezone | long | 时区 |

### searchArea

**接口描述：**区域搜索

**接口返回类型：**JSONObject

**输入示例：**

String nameZh = "";

String nameEn = "";

String areaId = "";

String postCode = "";

String telCode = "";

String keyWord = "金庄镇";

int start = 0;

int limit = 10;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| nameZh | string | false | 城市中文名称 |
| nameEn | string | false | 城市英文名称 |
| areaId | string | false | 区域编号 |
| postCode | string | false | 邮编 |
| telCode | string | false | 区号 |
| keyWord | string | false | 关键字，如果传递了该关键字，则不使用上面的4个参数进行过滤，又程序自动判断采用哪个字段进行过滤（如果全为数字，则使用areaId、postCode、telCode分别进行筛选，如果全为字母，则使用nameEn进行模糊匹配，如果有汉字，则使用nameZh进行模糊匹配） |
| start | int | true | 每页起始条数 |
| limit | int | true | 每页显示多少条 |

**输出示例：**

{

"timestamp":"2014-07-25 16:04:41",

"data":{

"total":1,

"records":[

{

"telCode":"0537",

"districtZh":"济宁",

"nameEn":"jinzhuangzhen",

"countyZh":"泗水县",

"postcode":"273200",

"areaId":101120708009,

"countyEn":"sishuixian",

"timezone":"8",

"nameZh":"金庄镇",

"nationZh":"中国",

"nationEn":"china",

"districtEn":"jining",

"continentEn":"asia",

"provZh":"山东",

"longitude":117.173,

"continentZh":"亚洲",

"latitude":35.6357,

"provEn":"shandong"

}

]

},

"status":"SUCCESS",

"msg":"获取成功"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | string | SUCCESS成功/FAILURE失败 |
| msg | string | 服务器返回信息 |
| timestamp | string | 时间，格式yyyy-MM-dd HH:mm:ss |
| data | JSONObject | 返回数据 |
| records | JSONArray | 查询记录的集合 |
| total | long | 总共条数 |

records属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| aliasEn | string | 用户ID。 |
| aliasZh | string | 图片标识。 |
| areaId | long | 12位城市ID |
| continentEn | string | 英文州名 |
| continentZh | string | 中文州名 |
| countyEn | string | 英文县名 |
| countyZh | string | 中文县名 |
| districtEn | string | 英文区名 |
| districtZh | string | 中文区名 |
| elevation | long | 海拔 |
| latitude | double | 纬度 |
| level | long | 城市级别 |
| longitude | double | 经度 |
| nameEn | string | 英文地名 |
| nameZh | string | 中文地名 |
| nationEn | string | 英文国家名 |
| nationZh | string | 中文国家名 |
| postcode | string | 邮编 |
| provEn | string | 英文省名 |
| provZh | string | 中文省名 |
| telCode | string | 电话区号 |
| timezone | long | 时区 |

### getAlmanac

**接口描述：**获取某日黄历信息

**接口返回类型：**JSONObject

**输入示例：**

String year = "2013";

String month = "5";

String day = "20";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| year | String | true | 年 |
| month | String | true | 月 |
| day | String | true | 日 |

**输出示例：**

{

"a":"四月初",

"b":"壬辰年乙巳月丁亥日",

"c":"龙",

"d":"诸事不宜",

"e":"诸事不宜",

"f":"冲蛇(巳亥相冲)",

"g":"母仓 四相 天德合 月德合 不将 玉堂",

"h":"游祸 天贼 重日",

"i":"仓库床外西北",

"j":"丁不剃头亥不嫁娶",

"k":"女士蝠宿星",

"l":"女星造作损婆娘，兄弟相嫌似虎狼，埋葬生灾逢鬼怪，颠邪疾病主瘟惶，为事遭官财失散，泻利留连不可当，开门放水用此日，全家财散主离乡。"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| a | String | 农历日期 |
| b | String | 天干地支 |
| c | String | 生肖 |
| d | String | 宜 |
| e | String | 忌 |
| f | String | 冲 |
| g | String | 吉神移趋 |
| h | String | 凶神移趋 |
| i | String | 胎神方位 |
| j | String | 彭祖百忌 |
| k | String | 星宿名称 |
| l | String | 星宿解释 |

### getHotCity

**接口描述：**获取热点城市

**接口返回类型：**JSONArray

**输入示例：无**

**输入参数：无**

**输出示例：**

[

[

"101010100",

"北京",

"BeiJing"

]

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
|  | JSONArray | 城市ID,城市中文名,城市英文名 |

### getHistoryWeather

**接口描述：**获取城市历史气侯数据

**接口返回类型：**JSONObject

**输入示例：**

String cityId = "101240106";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | String | true | 城市ID |
| language | String | true | 获取信息的语言 |

**输出示例：**

{

q:{

q1:

q2:21|26.3|40.5|79|119.6|236.3|459.2|297.7|116.3|132.5|43.4|16.3,

q3:0|0|0|0.4|1.8|4|26.5|41|4.2|0.3|0|0,

q4:-3.7|-0.7|5.8|14.2|19.9|24.4|26.2|24.9|20|13.1|4.6|-1.5,

q5:12.9|19.8|26.4|33|36.8|39.2|41.9|36.1|34.4|29.3|22|19.5,

q6:-18.3|-16|-15|-3.2|2.6|9.8|16.6|11.4|4.3|-3.5|-10.6|-15.6,

q7:1.8|5|11.6|20.3|26|30.2|30.9|29.7|25.8|19.1|10.1|3.7,

q8:-8.4|-5.6|0.4|7.9|13.6|18.8|22|20.8|14.8|7.9|0|-5.8,

q9:北京位于华北平原西北边缘，市中心位于北纬39度，东经116度，四周被河北省围着，东南和天津市相接。全市面积一万六千多平方公里，辖12区6县，人口1100余万。北京为暖温带半湿润大陆性季风气候，夏季炎热多雨，冬季寒冷干燥，春、秋短促，年平均气温10-12摄氏度。 北京是世界历史文化名城和古都之一。早在七十万年前，北京周口店地区就出现了原始人群部落&ldquo;北京人&rdquo;。而北京建城也已有两千多年的历史，最初见于记载的名字为&ldquo;蓟&rdquo;。公元前1045年北京成为蓟、燕等诸侯国的都城；公元前221年秦始皇统一中国以来，北京一直是中国北方重镇和地方中心；自公元938年以来，北京又先后成为辽陪都、金上都、元大都、明清国都。1949年10月1日正式定为中华人民共和国首都。 北京具有丰富的旅游资源，对外开放的旅游景点达200多处，有世界上最大的皇宫紫禁城、祭天神庙天坛、皇家花园北海、皇家园林颐和园，还有八达岭、慕田峪、司马台长城以及世界上最大的四合院恭王府等各胜古迹。全市共有文物古迹7309项，其中国家文物保护单位42个，市级文物保护单位222个。北京的市树为国槐和侧柏，市花为月季和菊花。另外，北京出产的象牙雕刻、玉器雕刻、景泰蓝、地毯等传统手工艺品驰誉世界。

}

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| q | JSONObject | 气候信息 |
| q1 | String | 月累计降水量(mm) |
| q2 | String | 月最大降水量(mm) |
| q3 | String | 月最小降水量(mm) |
| q4 | String | 月平均温度(℃) |
| q5 | String | 月极端最高温度(℃) |
| q6 | String | 月极端最低温度(℃) |
| q7 | String | 月平均最高温度(℃) |
| q8 | String | 月平均最低温度(℃) |
| q9 | String | 气候背景分析说明 |

### getNewsList

**接口描述：**获取资讯列表

**接口返回类型：**JSONObject

**输入示例：**

String category = "news";

String pageNum = "2";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| category | String | true | 资讯类别的英文名称 |
| pageNum | String | true | 页号 |

**输出示例：**

资讯列表：

{

"l1":[

{

"n1":"1689634",

"n2":"大暴雨席卷广西北海 市民积水路段拉网抓鱼",

"n3":"2012-08-07 11:21:19",

"n4":"",

"n5":"受“苏拉”残余云系影响，5日晚到6日白天广西北海普降暴雨、大暴雨。由于降雨强度大、持续时间长，北海市区各主要路段出现了很深的积水，内涝严重，全市各主要路段积水深度普遍在30至50厘米之间。有市民在积水路段拉网抓鱼。"

}

],

"l2":"2",

"l3":"20"

}

视频列表：

[

{

"v1":"新闻联播天气预报",

"v2":" CCTV-1《新闻联播》后播出的天气预报，内容包括48小时全国趋势预报、直辖市、省会城市、港澳48小时预报、部分城市24小时预报。",

"v3":" http://www.weather.com.cn/m2/i/lianbo.jpg",

"v4":" http://pv.mywtv.cn/android/lianbo\_h264.mp4"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| l1 | JSONArray | 资讯列表 |
| n1 | String | 新闻ID |
| n2 | String | 新闻标题 |
| n3 | String | 发布时间 |
| n4 | String | 标题图片 |
| n5 | String | 内容简介 |
| l2 | String | 当前页数 |
| l3 | String | 总页数 |
| v1 | String | 视频标题 |
| v2 | String | 视频简介 |
| v3 | String | 标题图片地址 |
| v4 | String | 视频地址 |

### getNewsContent

**接口描述：**获取资讯内容

**接口返回类型：**JSONObject

**输入示例：**

String category = "news";

String id = "1689634";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| category | String | true | 资讯类别的英文名称 |
| id | String | true | 资讯ID |

**输出示例：**

{

"c1":"未来三天华北黄淮等地明显降雨",

"c2":"未来三天华北黄淮等地明显降雨",

"c3":"",

"c4":"2012-07-06 10:59:31",

"c5":"预计未来三天西北地区东南部、四川盆地北部、黄淮中东部、华北南部等地将有大到暴雨，局地大暴雨；青藏高原中东部、内蒙古中东部、华北等地有阵雨或雷阵雨。需注意防范短时强降水可能带来的不利影响。",

"c6":"<P>中国天气网讯 今天（6日）开始黄淮等地的强降雨逐渐减弱。预计未来三天西北地区东南部、四川盆地北部、黄淮中东部、华北南部等地将有大到暴雨，局地大暴雨；青藏高原中东部、内蒙古中东部、华北等地有阵雨或雷阵雨。需注意防范短时强降水可能带来的不利影响。</P>…",

"c7":"中国天气网",

"c8":"gxw"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| c1 | String | 文章标题 |
| c2 | String | 文章长标题 |
| c3 | String | 标题图片 |
| c4 | String | 发布时间 |
| c5 | String | 内容简介 |
| c6 | String | 文章内容包含HTML标签 |
| c7 | String | 文章来源 |
| c8 | String | 文章作者 |

### getWeather

**接口描述：**获取城市的 时间、城市信息、实况、预报、指数、预警、广告、天气解读八方面数据

**接口返回类型：**Weather，异常返回null

**输入示例：**

String cityId = "101240106";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | String | true | 9位城市ID |
| language | String | true | 获取信息的语言 |

**输出示例：**参考Weather对象

**输出参数：**参考Weather对象

### getWeathers

**接口描述：**获取多个城市天气

**接口返回类型：**List<Weather>

**输入示例：**

List<String> cityIds = new ArrayList<String>();

cityIds.add("101010100");

cityIds.add("101020100");

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityIds | List<String> | true | 9位城市ID集合 |
| language | String | true | 获取信息的语言 |

**输出示例：**参考Weather对象

**输出参数：**参考Weather对象

### getWeather2

**接口描述：**获取城市的 时间、城市信息、实况、预报、指数、预警、广告、天气解读八方面数据

**接口返回类型：**Weather，异常返回null

**输入示例：**

String cityId = "101240106";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | String | true | 9/12位城市ID |
| language | String | true | 获取信息的语言 |

**输出示例：**参考Weather对象

**输出参数：**参考Weather对象，getWeather2()和getWeather返回的数据有所区别，位于方法getWeatherForecastInfo()、getHourlyFineForecast()

### getWeathers2

**接口描述：**获取多个城市天气

**接口返回类型：**List<Weather>

**输入示例：**

List<String> cityIds = new ArrayList<String>();

cityIds.add("101010100");

cityIds.add("101020100");

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityIds | List<String> | true | 9/12位城市ID集合 |
| language | String | true | 获取信息的语言 |

**输出示例：**参考Weather对象

**输出参数：**参考Weather对象，getWeathers2()和getWeathers返回的数据有所区别，位于方法getWeatherForecastInfo()、getHourlyFineForecast()

### Weather

#### getTimeInfo

**接口描述：**获取时间信息

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：**

int day = 0;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| day | int | true | 为0时返回所有7天的值，为1-7的值时，返回相应天数的时间信息 |

**输出示例：**

[

{

"t1":"20120306",

"t2":"二月十四",

"t3":" 壬辰年癸卯月丙寅日",

"t4":"星期二",

"t5":" 8",

"t6":"惊蜇",

"t7":"离三八妇女节还有两天",

"t8":" 九九第四天",

"t9":"双鱼座",

"t10":"龙",

"t11":"祈福、开光、 塑绘|作灶、掘井、动土。",

"t12":"火曜日",

"t13":"4|6"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| t1 | String | 阳历日期 |
| t2 | String | 阴历日期 |
| t3 | String | 黄历(天干地支) |
| t4 | String | 星期 |
| t5 | String | 时区 |
| t6 | String | 节气 |
| t7 | String | 节日信息 |
| t8 | String | 农历的说法 |
| t9 | String | 星座 |
| t10 | String | 生肖 |
| t11 | String | 黄历宜忌,中间用竖线分割，前为宜后为忌。 |
| t12 | String | 日韩关于星期的说法 |
| t13 | String | 汽车限行尾号，用|分割 |

#### getCityInfo

**接口描述：**获取城市信息

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"c1":"101091003",

"c2":"linzhang",

"c3":"临漳",

"c4":"handan",

"c5":"邯郸",

"c6":"hebei",

"c7":"河北",

"c8":"china",

"c9":"中国",

"c10":"3",

"c11":"0310",

"c12":"056600",

"c13":"114.554",

"c14":"36.258",

"c15":"66"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| c1 | String | 城市九位ID号 |
| c2 | String | 站点对应英文名称 |
| c3 | String | 站点对应中文名称 |
| c4 | String | 地市对应英文名称 |
| c5 | String | 地市对应中文名称 |
| c6 | String | 省对应英文名 |
| c7 | String | 省对应中文名称 |
| c8 | String | 国家对应英文名称 |
| c9 | String | 国家对应中文名称 |
| c10 | String | 站点级别(1省会2地级市3县级) |
| c11 | String | 电话区号 |
| c12 | String | 邮编 |
| c13 | String | 经度 |
| c14 | String | 纬度 |
| c15 | String | 海拔高度（单位m）没有小数位 |

#### getWeatherFactInfo

**接口描述：**获取天气实况，实况天气信息(每个字段都放有过去24小时和当前的数据。按时间正序排列，中间用 | 分割,最后为当前的实况。(如果没有某时间数据用 ? 表示)

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"l1":"8|9|10|10|10|11|12|11|8|8|6|5|4|3|2|-1|-2|0|1|0|2|4|6|8|8",

"l2":"60|80|50|50|56|60|72|71|58|58|60|50|40|39|32|28|25|40|41|40|52|54|36|38|38",

"l3":"6|8|5|5|6|1|2|1|5|5|6|5|4|9|2|2|2|4|4|4|5|5|3|3|3",

"l4":"6|8|5|5|6|1|2|1|5|5|6|5|4|9|2|2|2|4|4|4|5|5|3|3|3",

"l5":"01|01|01|01|02|02|03|01|01|01|01|01|01|01|01|01|01|01|01|01|01|01|01|01|01",

"l6":"0|0|0|0|1|1|3|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0",

"l7":"11:00",

"l8":"http://data.weather.com.cn/livebag/day/day08.jpg",

"l9":"17000|19600|17000|15200|14800|15200|17700|20900|16300|11500|9400|6800|5900|5300|3900|3300|2700|2700|2300|2700|2500|2100|3200|3600|3600",

"l10":"1015|1018|1018|1016|1015|1018|1016|1013|1019|1014|1015|1016|1015|1019|1018|1020|1015|1016|1019|1020|1016|1015|1015|1016|1018",

"l11":"1.9|2.3|1.7|1.8|1.9|2.5|2.4|2.5|1.7|1.9|1.6|1.2|0.9|1.5|1.7|0.8|0.1|0.0|0.6|0.7|0.9|0.0|1.0|0.5|0.7",

"l12":"17.4|18.3|18.5|19.3|19.3|20.4|20.4|20.4|19.3|17.3|16.2|15.2|14|13.8|12.4|11.3|10.5|10.6|10|10|9.9|11.7|15|18.7|19.6"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| l1 | String | 过去24小时每1小时温度当前的温度(摄氏度) |
| l2 | String | 过去24小时每1小时湿度和当前湿度) |
| l3 | String | 过去24小时每1小时风力和当前风力（单位是级） |
| l4 | String | 过去24小时每1小时风向编号 |
| l5 | String | 过去24小时实况天气现象编号和当前天气现象编号 |
| l6 | String | 过去24小时每1小时降水量和当前降水量 |
| l7 | String | 实况发布时间 |
| l8 | String | 实景天气背景URL |
| l9 | String | 能见度 |
| l10 | String | 气压 |
| l11 | String | 风速 |
| l12 | String | 体感温度 |

#### getWeatherForecastInfo

**接口描述：**获取天气预报

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：**

int day = 0;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| day | int | true | 为0时返回所有7天的值，为1-7的值时，返回相应天数的天气预报。（getWeather2接口返回的数据为15天，即day取值为0-15，0为返回所有15天的数据） |

**输出示例：**

[

{

"fa":"01",

"fb":"01",

"fc":"11",

"fd":"0",

"fe":"4",

"ff":"4",

"fg":"1",

"fh":"0",

"fi":"06:44|18:21"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| fa | String | 第一天白天天气现象编号 |
| fb | String | 第一天晚上天气现象编号 |
| fc | String | 第一天白天天气温度(摄氏度) |
| fd | String | 第一天晚上天气温度(摄氏度) |
| fe | String | 第一天白天风向编号 |
| ff | String | 第一天晚上风向编号 |
| fg | String | 第一天白天风力编号 |
| fh | String | 第一天晚上风力编号 |
| fi | String | 第一天日出日落时间(中间用|分割) |

#### getIndexInfo

**接口描述：**获取指数信息

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：**

String[] index = {"pk","ag","yh","bp"};

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| index | String[] | true | 由指数简称组成的数组 |
| language | String | true | 语言类型 |

**输出示例：**

[

{

"i4":[

{

"ic":"您将感到有些冷，可以适当开启制暖空调调节室内温度，以免着凉感冒。",

"ib":"开启制暖空调",

"ia":"540000000000"

}

],

"i3":"",

"i2":"空调开启指数",

"i1":"ac"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| i1 | String | 指数简称 |
| i2 | String | 指数的中英文名称 |
| i3 | String | 冠名后的名称 |
| i4 | JSONArray | 指数的七天预报数据 |

i4

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| ia | String | 指数编号 |
| ib | String | 指数级别 |
| ic | String | 指数说明 |

#### getWarningInfo

**接口描述：**获取预警信息

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"w1":"河北省",

"w2":"邯郸市",

"w3":"临漳县",

"w4":"14",

"w5":"道路结冰",

"w6":"02",

"w7":"黄色",

"w8":"2012-03-0610:30",

"w9":"今天下午到夜间，张家口、承德、唐山北部、秦皇岛北部有对 交通有影响的道路结冰。请防范。",

"w10":"201203060202054511森林火险橙色"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| w1 | String | 预警发布单位的省份名称 |
| w2 | String | 预警发布单位的市的名称 |
| w3 | String | 预警发布单位的县级的名称 |
| w4 | String | 预警类别对应图标编号 |
| w5 | String | 预警类别名称 |
| w6 | String | 预警级别对应图标编号 |
| w7 | String | 预警级别名称 |
| w8 | String | 预警发布时间 |
| w9 | String | 预警发布内容 |
| w10 | String | 预警信息ID |

#### getAdInfo

**接口描述：**获取广告信息

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：**

String[] adId = {"ad1","ad2"}

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| adId | String[] | true | 由广告ID组成的数组 |

**输出示例：**

[

{

"a1":"ad1",

"a2":"img",

"a3":"http://www.weather.com.cn/m2/i/logo\_header\_out.gif",

"a4":"",

"a5":"中国天气网",

"a6":"中国天气通",

"a7":"http://www.weather.com.cn",

"a8":"",

"a9":"TopRolling",

"a10":"LeftRolling"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| a1 | String | 对应metadata 的ad\_switch中的"id" |
| a2 | String | 广告的显示类型 |
| a3 | String | 本地访问广告url,即广告的资源链接地址，可为图片或页面 |
| a4 | String | 异地访问广告url。如为空，表示内容同a3 |
| a5 | String | 本地访问广告的text |
| a6 | String | 异地访问广告的text。如为空，表示内容同a5 |
| a7 | String | 本地访问广告的跳转链接url, 可支持图片或文字广告的链接 |
| a8 | String | 异地访问广告的跳转链接url,。如为空，表示内容同a7 |
| a9 | String | 本地访问广告的特效设置 |
| a10 | String | 异地访问广告的特效设置。如为空，表示内容同a9 |

#### getWeatherText

**接口描述：**获取天气解读信息

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"r1":"广州市区21日白天阴天到多云。萝岗观测站录得早晨最低气温19.8℃，白天最高气温23.6℃；能见度6～7公里；相对湿度70～90%。"。

"r2":"21日早晨，我省部分市县仍有轻雾，白天全省阴天到多云为主，中部、西北部有分散小雨，珠三角和粤东部分市县有轻微灰霾。早晨全省最低气温分布：西北部偏北11～15℃，比昨天下降4～6℃，东南部偏东市县14～16℃，西南部市县19～23℃，其余市县16～21℃。白天最高温度分布：西北部偏北16～20℃，东南部20～23℃，其余地区23～27℃。粤东海面东北风7～8级阵风9级，粤中海面和川岛附近海面偏东风6～7级阵风8级。"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| r1 | String | 标准版天气解读信息 |
| r2 | String | 专业版天气解读信息，为空时和标准版一样 |

#### getFineForecast

**接口描述：**获取精细化预报信息

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

[

{

"ja":"03",

"jb":"26",

"jc":"24",

"jd":"2",

"je":"1",

"jf":"1",

"jg":"0～1",

"jh":"201211300000",

"ji":"201211300300"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| ja | String | 该时间段内天气现象 |
| jb | String | 该时间段内最高温度 |
| jc | String | 该时间段内最低温度 |
| jd | String | 该时间段内风向编号 |
| je | String | 该时间段内风力编号 |
| jf | String | 区别降雨，降雪，无降雨降雪(0：无降雨和降雪， 1：降雨， 2：降雪) |
| jg | String | 该时间段内降水或降雪区间(单位是mm) |
| jh | String | 该时间段内预报开始时间 |
| ji | String | 该时间段内预报结束时间 |

#### getHourlyFineForecast

**接口描述：**获取逐时精细化预报信息(现为各城市逐3小时，后会变为逐小时。)（getWeather2接口返回的数据无此信息）

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：同getFineForecast**

**输出参数：同 getFineForecast**

#### getHourlyFineForecast2

**接口描述：**获取逐时精细化预报信息

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

[

{

"ja":"00",

"jb":"-4.3",

"jc":"8",

"jd":"0.6",

"je":"37",

"jf":"201402101000",

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| ja | String | 该时间段天气现象编号 |
| jb | String | 该时间段温度 |
| jc | String | 该时间段风向编号 |
| jd | String | 该时间段风速 |
| je | String | 该时间段相对湿度 |
| jf | String | 预报时间 |

#### getAirQualityInfo

**接口描述：**获取城市信息，空气质量信息(每个字段都放有过去24小时和当前的数据。按时间正序排列，中间用 | 分割,最后为当前的实况。(如果没有某时间数据用 ? 表示)

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"k1":"109|118|131|138|144|139|135|137|139|145|146|140|126|105|82|37|16|8|7|6|8|9|10|13|13",

"k2":"133|133|133|134|135|135|136|136|136|136|135|134|132|129|127|124|119|114|109|104|99|94|90|86|86",

"k3":"175|175|176|176|177|178|179|180|179|179|178|176|173|170|167|162|156|149|142|136|130|123|118|113|113",

"k4":"201303130800",

"k5":"3|4|5|5|5|6|7|6|6|5|4|3|3|3|3|3|3|2|2|3|2|2|2|2|2",

"k6":"5|5|5|5|5|5|5|5|5|5|5|5|5|4|4|4|4|4|4|4|4|4|4|4|4",

"k7":"52|53|61|71|71|84|85|75|75|56|56|60|66|70|73|70|67|62|61|62|62|53|58|59|59",

"k8":"62|63|63|64|64|68|69|69|69|67|66|64|63|63|62|62|62|63|63|64|64|65|65|65|65",

"k9":"70|92|104|123|123|136|122|108|108|89|93|93|95|102|103|86|83|92|103|120|127|106|110|98|98",

"k10":"88|90|93|96|96|102|104|105|105|102|101|98|96|95|94|93|93|94|95|98|100|101|102|104|104",

"k11":"0.691|0.767|0.8|0.85|0.85|0.945|0.918|0.808|0.808|0.645|0.636|0.658|0.717|0.825|0.858|0.917|0.883|0.87|0.882|0.945|0.909|0.8|0.811|0.9|0.9",

"k12":"0.618|0.635|0.654|0.674|0.674|0.716|0.728|0.733|0.733|0.73|0.725|0.718|0.712|0.712|0.714|0.719|0.726|0.732|0.744|0.762|0.779|0.79|0.799|0.81|0.81",

"k13":"7|9|10|13|13|28|38|56|56|81|76|62|41|26|17|13|10|10|10|11|12|15|12|9|9",

"k14":"53|53|53|53|53|50|48|55|55|84|84|84|84|84|84|84|84|84|84|84|84|84|84|84|84",

"k15":"13|13|12|11|11|14|18|23|23|40|49|55|57|57|54|49|41|33|24|18|14|12|11|11|11",

"k16":"41|41|41|41|41|41|41|41|41|42|49|55|57|58|58|58|58|58|58|58|58|58|58|58|58",

"k17":"可吸入颗粒物"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| k1 | String | PM2.5近一小时 (μg/m³) |
| k2 | String | PM2.5近24小时 (μg/m³) |
| k3 | String | 近24小时空气质量指数(AQI) |
| k4 | String | 数据发布日期 |
| k5 | String | 二氧化硫1小时平均 |
| k6 | String | 二氧化硫24小时滑动平均 |
| k7 | String | 二氧化氮1小时平均 |
| k8 | String | 二氧化氮24小时滑动平均 |
| k9 | String | 颗粒物（粒径小于等于10μm）1小时平均 |
| k10 | String | 颗粒物（粒径小于等于10μm）24小时滑动平均 |
| k11 | String | 一氧化碳1小时平均 |
| k12 | String | 一氧化碳24小时滑动平均 |
| k13 | String | 臭氧1小时平均 |
| k14 | String | 臭氧日最大1小时平均 |
| k15 | String | 臭氧8小时滑动平均 |
| k16 | String | 臭氧日最大8小时滑动平均 |
| k17 | String | 首要污染物 |

#### getAirQualityForecast

**接口描述：**获取空气质量3天预报，第一天数据为逐6小时，第二、三天为逐12小时

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

[

{

"k0":"201501060800",

"k1":"16.0",

"k2":"30.0",

"k3":"73.0",

"k4":"63.0",

"k5":"24.0",

"k6":"64.0"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| k0 | String | 预报数据起报时间 |
| k1 | String | SO2浓度(μg/m³) |
| k2 | String | NO2浓度(μg/m³) |
| k3 | String | PM10浓度(μg/m³) |
| k4 | String | CO浓度(μg/m³) |
| k5 | String | O3浓度(μg/m³) |
| k6 | String | PM2.5浓度(μg/m³) |

#### getPastTemperature

**接口描述：**获取昨天实况分析出的最高温和最低温，晚上更新数据会包含第一天的最高温。

**接口返回类型：**JSONArray，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

[

{

"pd":"18",

"pj":"20130904",

"pc":"22"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| pd | String | 最低温度 |
| pj | String | 日期 |
| pc | String | 最高温度 |

#### getFactTime

**接口描述：**获取天气实况发布时间，格式为HH:mm

**接口返回类型：**String，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**11:00

**输出参数：无**

#### getForecastTime

**接口描述：**获取天气预报发布时间，格式为yyyyMMddHHmm

**接口返回类型：**String，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**201203061100

**输出参数：无**

#### getFineForecastTime

**接口描述：**获取精细化预报发布时间，格式为yyyyMMddHHmmss

**接口返回类型：**String，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**20121119080000

**输出参数：无**

#### getHourlyFineForecastTime

**接口描述：**获取逐时精细化预报发布时间，格式为yyyyMMddHHmmss

**接口返回类型：**String，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**20121119080000

**输出参数：无**

#### getAirQualityInfoTime

**接口描述：**获取空气质量信息发布时间，格式为yyyyMMddHHmm

**接口返回类型：**String，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**201303130800

**输出参数：无**

### getWeatherMedia

**接口描述：**获取城市的城市信息、雷达、云图、视频四方面数据内容

**接口返回类型：**WeatherMedia对象，异常返回null

**输入示例：**

String cityId = "101240106";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | String | true | 城市ID |
| language | String | true | 获取信息的语言 |

**输出示例：**参考WeatherMedia对象

**输出参数：**参考WeatherMedia对象

### WeatherMedia

#### getCityInfo

**接口描述：**获取城市信息

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"c1":"101091003",

"c2":"linzhang",

"c3":"临漳",

"c4":"handan",

"c5":"邯郸",

"c6":"hebei",

"c7":"河北",

"c8":"china",

"c9":"中国",

"c10":"3",

"c11":"0310",

"c12":"056600",

"c13":"114.554",

"c14":"36.258",

"c15":"66"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| c1 | String | 城市九位ID号 |
| c2 | String | 站点对应英文名称 |
| c3 | String | 站点对应中文名称 |
| c4 | String | 地市对应英文名称 |
| c5 | String | 地市对应中文名称 |
| c6 | String | 省对应英文名 |
| c7 | String | 省对应中文名称 |
| c8 | String | 国家对应英文名称 |
| c9 | String | 国家对应中文名称 |
| c10 | String | 站点级别(1省会2地级市3县级) |
| c11 | String | 电话区号 |
| c12 | String | 邮编 |
| c13 | String | 经度 |
| c14 | String | 纬度 |
| c15 | String | 海拔高度（单位m）没有小数位 |

#### getRadarInfo

**接口描述：**获取雷达信息

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"r1":"河北\_石家庄基本反射率",

"r2":"http://i.weather.com.cn/i/product/pic/",

"r3":"png",

"r4":"l|m|s",

"r5":"sevp\_aoc\_rdcp\_sldas\_ebref\_az9311\_l88\_pi\_",

"r6":[

[

"20120319221000000",

"2012-03-2006:20:00"

],

[

"20120319222000000",

"2012-03-2006:10:00"

]

]

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| r1 | String | 名称 |
| r2 | String | 对应的URL地址前缀 |
| r3 | String | 文件类型 |
| r4 | String | 图片三种级别对应目录 大中小 用竖线分割 |
| r5 | String | 文件名称前缀 |
| r6 | JSONArray | 雷达图片信息列表(由文件名称、发布时间组成) |

#### getSatelliteImageryInfo

**接口描述：**获取卫星云图信息

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"s1":"中国大陆区域彩色云图",

"s2":"http://i.weather.com.cn/i/product/pic/",

"s3":"jpg",

"s4":"l|m|s",

"s5":"sevp\_nsmc\_wxcl\_asc\_e99\_achn\_lno\_py\_",

"s6":[

[

"20120320093000000",

"2012-03-2017:30:00"

],

[

"20120320090000000",

"2012-03-2017:00:00"

]

]

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| s1 | String | 名称 |
| s2 | String | 对应的URL地址前缀 |
| s3 | String | 文件类型 |
| s4 | String | 图片三种级别对应目录 大中小 用竖线分割 |
| s5 | String | 文件名称前缀 |
| s6 | JSONArray | 雷达图片信息列表(由文件名称、发布时间组成) |

#### getVideoInfo

**接口描述：**获取视频信息

**接口返回类型：**JSONObject，异常或不存在返回null

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"v1":"mp4|3gp|wmv",

"v2":"320X480|480X800",

"v3":"32X32|120X150",

"v4":"png",

"v5":"http://data.weather.com.cn/video/",

"v6":"http://data.weather.com.cn/video/pic/",

"v7":[

{

"va":"新闻联播",

"vb":"xwlb",

"vc":[

[

"03-21新闻联播",

"201203/retssts",

"2012-03-2117:30:00"

]

]

}

]

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| v1 | String | 视频支持格式， 用竖线分割 |
| v2 | String | 视频支持分辨率， 用竖线分割 |
| v3 | String | 缩略图支持尺寸， 用竖线分割 |
| v4 | String | 缩略图支持格式 |
| v5 | String | 视频地址前缀 |
| v6 | String | 视频缩略图地址前缀 |
| v7 | JSONArray | 视频的分类列表信息 |
| va | String | 分类名称 |
| vb | String | 分类存放目录 |
| vc | JSONArray | 分类下视频的列表（由视频名称、视频文件存放目录、视频发布时间组成） |

### parseWeather

**接口描述：**将天气现象编号转换为中文或英文（繁体暂不支持）

**接口返回类型：**String，多个用“|”隔开，不存在返回“null”,异常返回“error”

**输入示例：**

String numStr = "01|01|01|01|02|02|03|01|01|01|01|01";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| language | String | true | 获取信息的语言 |
| numStr | String | true | 编号，多个用“|”隔开 |

**输出示例：**

多云|多云|多云|多云|阴|阴|阵雨|多云|多云|多云|多云|多云

**输出参数：无**

### parseWindForce

**接口描述：**将风力编号转换为中文或英文（繁体暂不支持）

**接口返回类型：**String，多个用“|”隔开，不存在返回“null”,异常返回“error”

**输入示例：**

String numStr = "6|8|5|5|6|1|2|1|5|5|6";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| language | String | true | 获取信息的语言 |
| numStr | String | true | 编号，多个用“|”隔开 |

**输出示例：**

8-9级|10-11级|7-8级|7-8级|8-9级|3-4级|4-5级|3-4级|7-8级|7-8级|8-9级

**输出参数：无**

### parseWindDirection

**接口描述：**将风向编号转换为中文或英文（繁体暂不支持）

**接口返回类型：**String，多个用“|”隔开，不存在返回“null”,异常返回“error”

**输入示例：**

String numStr = "01|01|01|01|02|02|03|01|01|01|01|01";

String language = Constants.Language.ZH\_CN;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| language | String | true | 获取信息的语言 |
| numStr | String | true | 编号，多个用“|”隔开 |

**输出示例：**

多云|多云|多云|多云|阴|阴|阵雨|多云|多云|多云|多云|多云

**输出参数：无**

### parseAQI

**接口描述：**根据AQI数值获取空气质量指数（AQI）分级相关信息

**接口返回类型：**JSONArray，异常或不存在返回null，超出范围的编号返回空JSONObject

**输入示例：**

String num = "50|100|160";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| num | String | true | 编号，多个用“|”隔开 |

**输出示例：**

[

{

"level":"一级",

"measure":"各类人群可正常活动。",

"effect":"空气质量令人满意，基本无空气污染。",

"level\_in":"优",

"color":"绿色"

"rgb":"#00E400"

}

]

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| level | String | 级别 |
| measure | String | 建议采取的措施 |
| effect | String | 对健康的影响 |
| level\_in | String | 级别说明 |
| color | String | 颜色的中文名称 |
| rgb | String | 颜色的16进制 |

### parseWarning

**接口描述：**获取预警信号标准说明防御指南

**接口返回类型：**String，不存在返回null, 预警信号标准说明和预警信号防御指南，用“|”隔开

**输入示例：**

String id = " 1303";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| id | String | true | 预警信息中w4+w6 |

**输出示例：**

2小时内发生雷电活动的可能性非常大，或者已经有强烈的雷电活动发生，且可能持续，出现雷电灾害事故的可能性非常大。|1、政府及相关部门按照职责做好防雷应急抢险工作；<br>2、人员应当尽量躲入有防雷设施的建筑物或者汽车内，并关好门窗；<br>3、切勿接触天线、水管、铁丝网、金属门窗、建筑物外墙，远离电线等带电设备和其他类似金属装置；<br>4、尽量不要使用无防雷装置或者防雷装置不完备的电视、电话等电器；<br>5、密切注意雷电预警信息的发布。

**输出参数：无**

## 用户相关接口（UserAPI）

### register

**接口描述：**用户注册

**接口返回类型：**JSONObject

**输入示例：**

String username = "abc";

String password = "1235";

String cellphone = "15510861086";

String email = "chinaWeather@163.com";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| username | String | true | 用户名 |
| password | String | true | 密码 |
| cellphone | String | false | 电话 |
| email | String | false | 邮箱 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"注册成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### login

**接口描述：**用户登录

**接口返回类型：**JSONObject

**输入示例：**

String username = "abc";

String password = "1235";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| username | String | true | 用户名 |
| password | String | true | 密码 |

**输出示例：**

{

"expireTime":"2012-11-29",

"data":{

"vip":0,

"groupId":[

56

],

"userName":"test1",

"uId":"1350903144872195",

" belongProvince":10101,

" verifyEmail":false

},

"msg":"登录成功.",

"status":"SUCCESS"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| expireTime | String | 过期时间 |
| data | JSONObject | 用户信息 |
| vip | long | 用户VIP等级 |
| groupId | JSONArray | 用户所属组ID |
| userName | String | 用户名 |
| uId | String | 用户ID |
| belongProvince | String | 用户所属省份 |
| verifyEmail | boolean | 是否已经绑定邮箱 |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### logout

**接口描述：**注销用户，用于天气通用户和第三方登录用户

**接口返回类型：**无

**输入示例：无**

**输入参数：无**

**输出示例：无**

**输出参数：无**

### isExpired

**接口描述：**登录是否过期，用于天气通用户和第三方登录用户

**接口返回类型：**boolean

**输入示例：无**

**输入参数：无**

**输出示例：无**

**输出参数：无**

### changePassword

**接口描述：**用户修改密码（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

String oldPassword = "1235";

String newPassword = "12345";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| oldPassword | String | true | 旧密码 |
| newPassword | String | true | 新密码 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"密码修改成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### getUserInfo

**接口描述：**获取用户信息（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"status":"SUCCESS",

"data":{

"nickName":"lilith",

"email":"1012090873@qq.com",

"bookedIndex":[

"ac",

"pk",

"ag"

],

"bookedCityId":[

"101020100"

],

"sex":"male",

"cellphone":"010-53123131",

"birthday":"1987-11-19"

}

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| data | JSONObject | 用户信息 |
| status | String | SUCCESS成功/FAILURE失败 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| sex | String | 性别 |
| cellphone | String | 电话 |
| birthday | String | 生日 |
| nickName | String | 昵称 |
| email | String | 邮箱 |
| bookedCityId | JSONArray | 订阅的城市ID |
| bookedIndex | JSONArray | 订阅的指数 |

### updateUserInfo

**接口描述：**修改用户信息（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONObject userInfo = new JSONObject();

userInfo.put("birthday", "1987-11-19");

userInfo.put("cellphone", "010-53123131");

userInfo.put("nickName", "lilith");

userInfo.put("email", "chinaWeather@163.com");

userInfo.put("sex", Constants.Sex.MALE);

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| userInfo | JSONObject | true | 用户信息 |
| birthday | String | false | 生日 |
| cellphone | String | false | 电话 |
| sex | String | false | 性别 |
| nickName | String | false | 昵称 |
| email | String | false | 邮箱 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"用户信息更新成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### bindingEmail

**接口描述：**绑定用户邮箱（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

String email = "chinaWeather@163.com";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| email | String | true | 邮箱 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"验证邮件已发出，请登录您的邮箱验证。"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### uploadPortrait

**接口描述：**用户头像上传（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

File file = new File(dir);

String imageSuffix = Constants.ImageSuffix.JPG;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| file | File | true | 图片字节流(图片<50kb) |
| imageSuffix | String | true | 图像文件后缀名，目前支持jpg、jpeg、png三种格式，请确保图片的字节流编码符合所设置后缀的格式。 |

**输出示例：**

{

"imageName":"f4638a79a5b444ddb2be98b4c7fa31dc",

"imageSuffix":"jpg",

"result":1

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | long | 是否获取成功，1表示成功，0表示失败 |
| imageName | String | 文件名，可以通过文件名从服务器下载图片 |
| imageSuffix | String | 文件后缀 |
| detail | String | 本次获取结果的详细信息，如果result的结果是不成功，该属性中会包含错误信息。 |

### uploadLocalPortrait

**接口描述：**用户默认头像上传（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

String imageTag = "portrait1";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| imageTag | String | true | 图片标识。 |

**输出示例：**

{

" imageTag":" portrait1",

"result":1

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | long | 是否获取成功，1表示成功，0表示失败 |
| detail | String | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| imageTag | String | 图片标识。 |

### getPortrait

**接口描述：**获取用户头像（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

String uid = null;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| uid | String | false | 获取指定用户的头像，uid为空或null则获取登录用户的头像。 |

**输出示例：**

{

"imageName":"http://myspace0%s.weather.com.cn/ 96508828e3bc4b25a23c0847e8e195bd.jpg ",

"imageSuffix":"jpg",

"result":1,

"uid":"1365738248300649",

"updateTime":1368689445000,

"mode":"upload"

}

或

{

"imageTag":" portrait1",

"result":1,

"uid":"1365738248300649",

"updateTime":1368689445000,

"mode":"local"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | long | 是否获取成功，1表示成功，0表示失败 |
| uid | String | 用户ID |
| updateTime | long | 上传时间的时间戳。 |
| imageName | String | 文件地址，可以通过文件名从服务器下载图片。当mode为local时不存在 |
| imageSuffix | String | 图像文件后缀名，目前支持jpg、jpeg、png三种格式，请确保图片的字节流编码符合所设置后缀的格式。当mode为local时不存在 |
| imageTag | String | 默认头像标识。当mode为upload时不存在 |
| mode | String | 头像最后一次上传的方式，”local”或者”upload” |

### bookCity

**接口描述：**用户订阅城市天气，跟推送功能相关联（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONArray cityId = new JSONArray();

cityId.put("101010100");

cityId.put("101020100");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | JSONArray | true | 用户订阅的城市ID集合 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"订阅成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### unBookCity

**接口描述：**取消订阅城市天气（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONArray cityId = new JSONArray();

cityId.put("101010100");

cityId.put("101020100");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | JSONArray | true | 用户取消订阅的城市ID集合 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"取消订阅成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### bookIndex

**接口描述：**用户订阅指数（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONArray index = new JSONArray();

index.put("pk");

index.put("ag");

index.put("ac");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| index | JSONArray | true | 用户订阅的指数简称的集合 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"订阅成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### unBookIndex

**接口描述：**取消订阅指数（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONArray index = new JSONArray();

index.put("pk");

index.put("ag");

index.put("ac");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| index | JSONArray | true | 用户取消订阅的指数简称的集合 |

**输出示例：**

{

"status":"SUCCESS",

"msg":"取消订阅成功."

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### thirdPartyLogin

**接口描述：**第三方登录

**接口返回类型：**JSONObject

**输入示例：**

int type = 1;

String openId = "DF8E45D953552E252FD5691836E3E358";

String accessToken = "E1643A21718D419BDF0F36FE71385CA8";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| type | int | true | 用户类型，1为腾讯QQ、2为新浪微博、3为微信 |
| openId | String | true | 第三方授权返回的openId,新浪为uid |
| accessToken | String | true | 第三方授权返回的accessToken |

**输出示例：**

{

"expireTime":"2012-11-29",

"data":{

"vip":0,

"groupId":[

56

],

"uId":"1350903144872195",

" belongProvince":10101,

" verifyEmail":false

},

"msg":"登录成功.",

"status":"SUCCESS"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| expireTime | String | 过期时间 |
| data | JSONObject | 用户信息 |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

data属性：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| vip | long | 用户VIP等级 |
| groupId | JSONArray | 用户所属组ID |
| uId | String | 用户ID |
| belongProvince | String | 用户所属省份 |
| verifyEmail | boolean | 是否已经绑定邮箱 |

### addScore

**接口描述：**增加积分（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

int score = 10;

String comment = "测试加分";

String action = "测试";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| score | int | true | 积分,可为负数 |
| comment | String | true | 积分描述 |
| action | String | true | 积分动作 |

**输出示例：**

{

"timestamp": "2015-03-11 15:47:35",

"data": {

"id": 18

},

"status": "SUCCESS",

"msg": "add success!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| timestamp | String | 时间 |
| data | JSONObject | 积分信息 |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

data属性：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 积分记录ID |

### getScoreHistory

**接口描述：**获取用户的总积分和积分历史记录（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

long start = 0L;

long limit = 10L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| start | long | true | 分页参数，起始记录数 |
| limit | long | true | 分页参数，每页记录数 |

**输出示例：**

{

"timestamp": "2015-03-11 15:54:20",

"data": {

"history": [{

"score": 10,

"id": 18,

"action": "测试",

"createTime": 1426060055864,

"comment": "测试加分"

}],

"sum": 20

},

"status": "SUCCESS",

"msg": "query success!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| timestamp | String | 时间 |
| data | JSONObject | 积分信息 |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

data属性：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| sum | int | 总积分 |
| history | JSONArray | 积分历史记录 |

history属性：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 积分记录ID |
| score | int | 积分 |
| action | String | 积分动作 |
| comment | String | 积分描述 |
| createTime | long | 创建时间 |

## 增值服务相关接口（VASAPI）

### getVAS

**接口描述：**获取可订阅的增值业务服务

**接口返回类型：**JSONObject

**输入示例：**

String longitude = "116.4491166";

String latitude = "39.9024854";

String cityId = "101010100";

String provinceId = "10101";

JSONArray bookedCityId = new JSONArray();

bookedCityId.put("101010100");

bookedCityId.put("101010200");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| longitude | String | false | 经度 |
| latitude | String | false | 纬度 |
| cityId | String | false | 所在城市ID |
| provinceId | String | true | 省级ID（参数为空获取所有省份的服务），“全国精品推荐”为0 |
| bookedCityId | JSONArray | false | 定制的城市ID集合 |

**输出示例：**

{

"data":{

"special":[

{

"createTime":"2012-10-23 10:46:28",

"icon":"http://114.255.31.31/chinaweather/images/bj\_txwb.jpg",

"id":1,

"provinceIdOfUser":10101,

"serviceName":"导航服务",

"tip":{

"tipType":1,

"tipContent":"效果内容"

},

"category":1,

"available":false,

"info":{

"content":"该服务仅供vip7级以上用户使用，如有需要，请升级至vip7",

"url":"http://114.255.31.31/chinaweather/cash/login"

}

},

{

"createTime":"2012-10-23 10:46:28",

"icon":"http://114.255.31.31/chinaweather/images/bj\_qxpd.jpg",

"id":2,

"provinceIdOfUser":10101,

"serviceName":"导航服务",

"tip":{

"tipType":1,

"tipContent":"效果内容"

},

"category":2,

"data":{

"version":"2.1.5",

"package":"com.yryj.chinaweather"

},

"available":true,

"url":"http://3gplatform.weather.com.cn/operationConsole/resources/download/android/ChinaWeather\_demo.apk"

}

],

"vas":[

{

"available":true,

"category":1,

"createTime":"2013-05-16 19:22:19",

"icon":"http://114.255.31.31/chinaweather/images/bj\_qxpd.jpg",

"id":35,

"provinceIdOfUser":10101,

"serviceName":"气象频道",

"tip":{

"tipContent":"提示new",

"tipType":"1"

},

"url":"http://data.weather.com.cn/cnweather/templates/templates\_NewsList\_main.html?dataFile=http://data.weather.com.cn/cnweather/provdata/beijing/Video/data.html"

}

]

},

"msg":"获取成功",

"status":"SUCCESS",

"timestamp":"2013-05-17 13:46:32"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| data | JSONObject | 增值服务数据 |
| msg | String | 服务器返回信息 |
| status | String | SUCCESS登录成功/FAILURE登录失败 |
| timestamp | String | 时间 |
| special | JSONArray | 导航服务集合 |
| vas | JSONArray | 增值服务集合 |

special和vas服务属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| category | long | 类别,1为Html、2为app插件 |
| available | boolean | 是否可用,true为可用、false为不可用 |
| createTime | String | 服务创建时间(用来判断是否有更新) |
| icon | String | 服务图标地址 |
| id | long | 服务ID |
| provinceIdOfUser | String | 服务所属省份 |
| serviceName | String | 服务名称 |
| url | String | 服务地址 |
| info | JSONObject | 服务介绍 |
| data | JSONObject | app插件信息 |
| tip | JSONObject | 显示效果 |

info属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| content | String | 服务介绍文字内容 |
| url | String | 服务介绍地址/付费入口 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| version | String | app插件版本 |
| package | String | app插件包名 |

tip属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| tipType | String | 显示效果类型 |
| tipContent | String | 显示效果内容 |

**注意：**

当category:1时，即HTML：无data(app插件信息)字段。

当available:false时，即服务不可用：无url(服务地址)字段

当available:true时，即服务可用：无info (服务介绍)字段

### subscribeVAS

**接口描述：**订阅增值业务服务（需要用户登录）

**接口返回类型：**json，异常返回error

**输入示例：**

JSONArray serviceIds = new JSONArray();

serviceIds.put("21");

serviceIds.put("61");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| serviceIds | JSONArray | true | 服务ID |

**输出示例：**

{

"msg":"订阅失败",

"status":"FAILURE"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### unsubscribeVAS

**接口描述：**退订增值业务服务（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONArray serviceIds = new JSONArray();

serviceIds.put("21");

serviceIds.put("61");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| serviceIds | JSONArray | true | 服务ID |

**输出示例：**

{

"msg":"退订失败",

"status":"FAILURE"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| msg | String | 文字信息 |
| status | String | SUCCESS成功/FAILURE失败 |

### subscribedVAS

**接口描述：**获取已订阅的增值业务服务（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**同getVAS

**输出参数：**同getVAS

## 实景天气相关接口（SceneAPI）

### uploadScene

**接口描述：**实景图片上传

**接口返回类型：**JSONObject

**输入示例：**

File image = new File(dir\_img);

String imageSuffix = Constants.ImageSuffix.JPG;

File voice = new File(dir\_voice);

String voiceSuffix = "amr";

JSONObject extra\_json = new JSONObject();

extra\_json.put("cityId", "101010100");

JSONObject location = new JSONObject();

location.put("lat", "98.12121");

location.put("lon", "102.4111");

location.put("addr", "北京市朝阳区京密路");

JSONObject weather = new JSONObject();

weather.put("weatherP", "00");

weather.put("temp", "18");

weather.put("windF", "3");

weather.put("windD", "2");

extra\_json.put("location", location);

extra\_json.put("weather", weather);

extra\_json.put("myExtra", "123");

String extra = extra\_json.toString();

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| image | File | true | 图片字节流（图片<500kb、声音<50kb、声音时间长度<60s） |
| imageSuffix | String | true | 图像文件后缀名，目前支持jpg、jpeg、png三种格式，请确保图片的字节流编码符合所设置后缀的格式。 |
| voice | File | false | 音频字节流。如果本字段中有数据，也要保证”voiceSuffix”字段含有有效数据。 |
| voiceSuffix | String | false | 音频格式后缀。 |
| extra | String | true | 附加信息，客户端想跟随图片一起保存的信息。该信息会在客户端获得图片信息时同时获取到。 |

extra 的JSON 格式

{

"location":{

"lat":"98.12121",

"lon":"102.4111",

"addr":"北京市朝阳区"

},

"weather":{

"weatherP":"00",

"temp":"18",

"windF":"3",

"windD":"2"

},

"cityId":"101010200",

" myExtra":"123"

}

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| cityId | String | true | 城市ID |
| location | JSONObject | true | 位置信息 |
| weather | JSONObject | true | 天气信息 |
| myExtra | String | false | 自定义属性，可根据需要自定义key（myExtra） |

location属性：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| lat | String | true | 纬度 |
| lon | String | true | 经度 |
| addr | String | true | 地理位置 |

weather属性：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| weatherP | String | true | 天气现象编号 |
| temp | String | true | 温度（℃） |
| windF | String | true | 风力编号 |
| windD | String | true | 风向编号 |

**输出示例：**

{

"imageName":"f4638a79a5b444ddb2be98b4c7fa31dc",

"imageSuffix":"jpg",

"voiceName":" 466c0d43b96b443e919c5c1fbe1589ba",

"voiceSuffix":" amr",

"result":1

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | long | 是否获取成功，1表示成功，0表示失败 |
| detail | String | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| imageName | String | 图片文件名，可以通过文件名从服务器下载图片。 |
| imageSuffix | String | 图片文件后缀 |
| voiceName | String | 音频文件名，可通过文件名从服务器下载音频流。 |
| voiceSuffix | String | 音频格式后缀。 |

### getScene

**接口描述：**获取实景信息

**接口返回类型：**JSONObject

**输入示例：**

String uid = null;

int limit = 2;

String offset = null;

String provinceId = null;

String cityId = "101250102";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| uid | string | false | 用户ID。如果该参数有意义，查找该用户上传的实景天气图片。优先级：1 |
| provinceId | string | false | 省份ID。如果该参数有意义，查找该省份的实景天气图片。优先级：2 |
| cityId | string | false | 城市ID。如果该参数有意义，查找该城市的实景天气图片。优先级：3 |
| limit | int | true | 要获得的实景天气图片数量。 |
| offset | string | false | 图片的偏移量，该字段在第一次获取时不需要设置，在获得接下来的图片时需要设置，这个值应该成上一次请求返回结果中的 offset 字段的内容。 |

**输出示例：**

{

"offset":"1365738248300649!9223370665244431884!6f6e0164eb0e459a9d6d7f669c2c0999",

"result":1,

"scenes":[

{

"extra":{

"cityId":"101010200",

"location":{

"addr":"北京市朝阳区",

"lat":"98.12121",

"lon":"102.4111"

},

"weather":{

"temp":"18",

"weatherP":"00",

"windD":"2",

"windF":"3"

}

},

"imageName":"http://myspace03.weather.com.cn/74a4565863ee4939abd502964fb64574.jpg",

"imageSuffix":"jpg",

"imageWidth":1024,

"imageHeight":768,

"imageSize":608834,

"voiceName":"http://myspace01.weather.com.cn/voice/466c0d43b96b443e919c5c1fbe1589ba.amr",

"voiceSuffix":"amr",

"uid":"1365738248300649",

"uploadTime":1371612593318,

"commentQuantity":10,

"laudQuantity":13,

"memberInfo":{

"annualIncome":3000,

"birthday":1388505600000,

"businessScope":"it",

"cellphone":"18561105639",

"company":"eraymobile",

"companyAddr":"建外soho",

"degree":"high",

"hometown":101300802,

"icon":"http://myspace03.weather.com.cn/f04ca6cb4251491184e1cfe633ef0b60.JPG",

"id":38740,

"livingcity":101300802,

"major":"机械制造",

"nickName":"水儿昵称",

"occupational":"软件开发",

"position":"架构师",

"realName":"水儿",

"school":"重庆大学",

"sex":"男",

"trade":"it",

"uId":"1419507848479499",

"zipCode":"400400"

}

}

]

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | int | 是否获取成功，1表示成功，0表示失败 |
| detail | String | 本次获取结果的详细信息，如果result的结果是不成功，该属性中会包含错误信息 |
| offset | String | 下一次获取的偏移量，该值需要在连续获取图片时设置到请求参数中。在结果中，如果该值不存在，则代表接下来没有图片了。 |
| scenes | JSONArray | 实景图片信息的集合 |

scenes：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| uid | String | 用户ID |
| imageName | String | 图片文件名，可以通过文件名从服务器下载图片。 |
| imageSuffix | String | 图片文件后缀。 |
| imageWidth | int | 图片宽度。 |
| imageHeight | int | 图片高度。 |
| imageSize | int | 图片大小，单位：byte。 |
| voiceName | String | 音频文件名，可以通过文件名从服务器下载音频流。 |
| voiceSuffix | String | 音频文件后缀。 |
| uploadTime | long | 上传时间的毫秒值 |
| extra | String | 上传图片时附加的信息 |
| commentQuantity | long | 本图片被评论的总数。 |
| laudQuantity | long | 本图片被“赞”的总数。 |
| memberInfo | JSONObject | 上传本图片的用户信息 |

memberInfo：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| annualIncome | int | 年收入 |
| birthday | long | 生日 |
| businessScope | String | 经营范围 |
| cellphone | String | 联系电话 |
| company | String | 公司名称 |
| companyAddr | String | 公司地址 |
| degree | String | 学历 |
| hometown | long | 籍贯 |
| icon | String | 头像 |
| livingcity | long | 生活城市 |
| major | String | 专业 |
| nickName | String | 昵称 |
| occupationa | String | 职业 |
| position | String | 职位 |
| realName | String | 真实姓名 |
| school | String | 学校 |
| sex | String | 性别 |
| trade | String | 行业 |
| zipCode | int | 邮编 |
| uId | String | 用户ID |

### uploadComment

**接口描述：**图片评论信息上传（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

String imageName = "e11d182f22314b5ea0caa34546351b4c";

String commentType = Constants.CommentType.VOICE;

String content = "第1条评论";

File voice = new File(dir\_voice);

String voiceSuffix = "amr";

String extra = "{'nickname':'ceshi','portrait':'http://'}";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| imageName | string | true | 图片标识。 |
| commentType | string | true | 评论类型，包括文字评论(“text”)和语音评论(“voice”)。 |
| content | string | false | 文字评论内容，如果评论类型为”text”，则本字段必填。 |
| voice | File | false | 语音评论的语音文件，如果评论类型为”voice”，则本字段必填。 |
| voiceSuffix | string | false | 语音评论的语音后缀名，如果评论类型为”voice”，则本字段必填。 |
| extra | string | true | 附加字段，内容为JSON格式，可传输评论用户昵称和头像地址。 |

**输出示例：**

{

"commentType":"text",

"content":"第1条评论",

"imageName":" e11d182f22314b5ea0caa34546351b4c",

"result":1,

"uid":"1365738248300649",

"uploadTime":1373959998380

"extra":"{'nickname':'ceshi','portrait':'http:// '}"

"commentStatistics":{

"commentQuantity":4,

"imageName":" e11d182f22314b5ea0caa34546351b4c"

}

}

或

{

"commentType":"voice",

"imageName":" e11d182f22314b5ea0caa34546351b4c ",

"result":1,

"uid":"1365738248300649",

"uploadTime":1373960329655,

"voiceName":"445fbc5709d24d6ebe83d695c0987bda",

"voiceSuffix":"amr"

"extra":"{'nickname':'ceshi','portrait':'http:// '}"

"commentStatistics":{

"commentQuantity":4,

"imageName":" e11d182f22314b5ea0caa34546351b4c "

}

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | int | 是否获取成功，1表示成功，0表示失败。 |
| detail | string | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| uid | string | 用户ID。 |
| imageName | string | 图片标识。 |
| commentType | string | 评论类型，包括文字评论(“text”)和语音评论(“voice”)。 |
| content | string | 文字评论内容。如果评论类型为”text”，则本字段存在。 |
| voice | string | 语音评论的语音字节流。如果评论类型为”voice”，则本字段存在。 |
| voiceSuffix | string | 语音评论的语音后缀名。如果评论类型为”voice”，则本字段存在。 |
| uploadTime | long | 评论上传时间。 |
| extra | string | 附加信息，客户端想跟随评论一起保存的信息。该信息会在客户端获得评论信息时同时获取到。该字段是一个 json 字符串。 |
| commentStatistics | JSONObject | 本次被评论图片的统计信息（只包含图片总评论数量） |

### getComment

**接口描述：**获取图片评论信息

**接口返回类型：**JSONObject

**输入示例：**

String imageName = "e11d182f22314b5ea0caa34546351b4c";

int limit = 2;

String offset = null;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| imageName | string | true | 图片名称。不包括图片后缀名。 |
| limit | int | true | 要获得的评论数量。 |
| offset | string | false | 下一次获取的偏移量，该值需要在连续获取评论时设置到请求参数中。在结果中，如果该值不存在，则代表接下来没有评论了。 |

**输出示例：**

{

"comments":[

{

"commentType":"voice",

"imageName":"3e35c0d558c84d969a8e8050685ecfd8",

"uid":"1365738248300649",

"uploadTime":9223370662894447000,

"voiceName":"445fbc5709d24d6ebe83d695c0987bda",

"voiceSuffix":"amr"

"extra":"{'nickname':'ceshi','portrait':'http:// '}"

},

{

"commentType":"text",

"content":"第1条评论",

"imageName":"3e35c0d558c84d969a8e8050685ecfd8",

"uid":"1365738248300649",

"uploadTime":9223370662894777000

"extra":"{'nickname':'ceshi','portrait':'http:// '}"

}

],

"offset":"3e35c0d558c84d969a8e8050685ecfd8!9223370662895238819",

"result":1

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | int | 是否获取成功，1表示成功，0表示失败 |
| detail | string | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| offset | string | 下一次获取的偏移量，该值需要在连续获取评论时设置到请求参数中。在结果中，如果该值不存在，则代表接下来没有评论了。 |
| comments | list | 评论信息的集合。每个comment的属性如下表所示。 |

comment 属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| uid | string | 用户ID。 |
| imageName | string | 图片标识。 |
| commentType | string | 评论类型，包括文字评论(“text”)和语音评论(“voice”)。 |
| content | string | 文字评论内容。如果评论类型为”text”，则本字段存在。 |
| voice | string | 语音评论的语音字节流。如果评论类型为”voice”，则本字段存在。 |
| voiceSuffix | string | 语音评论的语音后缀名。如果评论类型为”voice”，则本字段存在。 |
| uploadTime | long | 评论上传时间的毫秒值。 |
| extra | json | 评论上传时的附加字段 |

### getComments

**接口描述：**批量获得实景图片的评论统计信息

**接口返回类型：**JSONObject

**输入示例：**

JSONArray imageNames = new JSONArray();

imageNames.put("1db6f544eb764803bc8b8b8b98b6e3e7");

imageNames.put("f7fcd33e10b1440196d64d17a204c5ff");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| imageNames | JSONArray | true | 要查询的图片名称集合 |

**输出示例：**

{

"conclusions":[{

"commentQuantity":4,

"imageName":"1db6f544eb764803bc8b8b8b98b6e3e7",

"laudQuantity":2

},{

"commentQuantity":2,

"imageName":"f7fcd33e10b1440196d64d17a204c5ff",

"laudQuantity":1

}],

"result":1

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | int | 是否获取成功，1表示成功，0表示失败 |
| detail | string | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| conclusions | string | 查询结果。该集合中的每一项都包含图片名称及对应的评论统计信息。 |

conclusion 属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| imageName | string | 图片标识。 |
| commentQuantity | int | 该图片被评论的总数。 |
| laudQuantity | int | 该图片被“赞”的总数。 |

### uploadLaud

**接口描述：**图片“赞”信息上传（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

String imageName = "e11d182f22314b5ea0caa34546351b4c";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| imageName | string | true | 图片名称。不包括图片后缀名。 |

**输出示例：**

{

"result":1,

"laudStatistics":{

"imageName":"1db6f544eb764803bc8b8b8b98b6e3e7",

"laudQuantity":2

}

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | int | 是否获取成功，1表示成功，0表示失败 |
| detail | string | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| laudStatistics | JSONObject | 本次被“赞”图片的统计信息（只包含图片“赞”的数量）。 |

laudStatistics属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| imageName | string | 图片标识。 |
| laudQuantity | long | 图片被“赞”的总数。 |

### inspectLauded

**接口描述：**查询用户对一批图片“赞”的状态（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

JSONArray imageNames = new JSONArray();

imageNames.put("1db6f544eb764803bc8b8b8b98b6e3e7");

imageNames.put("f7fcd33e10b1440196d64d17a204c5ff");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| imageNames | JSONArray | true | 要查询的图片名称集合 |

**输出示例：**

{

"result":1,

"conclusions":[{

"hasLauded":1,

"imageName":"1db6f544eb764803bc8b8b8b98b6e3e7"

},{

"hasLauded":0,

"imageName":"f7fcd33e10b1440196d64d17a204c5ff"

}]

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| result | int | 是否获取成功，1表示成功，0表示失败 |
| detail | string | 本次获取结果的详细信息，如果result的结果表示不成功，该字段中会包含错误信息。 |
| conclusions | string | 查询结果。该集合中的每一项都包含图片和用户所对应的状态信息。 |

conclusion 属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| imageName | string | 图片标识。 |
| hasLauded | int | 用户对图片是否已经“赞”过。”1”代表已经“赞”过，”0”代表没有“赞”过。 |

## 真人天气秀相关接口（ShowAPI）

### getShowTypes

**接口描述：**获取真人秀类别列表

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"data":[

{

"category":1,

" categoryName":"v1"

},

{

" category":2,

"value":"v1"

}

],

"status":"SUCCESS",

"timestamp":"2013-08-22 17:21:44"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 真人秀类别列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| category | Integer | 类别 |
| categoryName | String | 类别名称 |

### getShowList

**接口描述：**获取符合条件的真人秀列表

**接口返回类型：**JSONObject

**输入示例：**

JSONArray type = new JSONArray();

type.put("1");

type.put("2");

Integer gender = 2;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| type | JSONArray | false | 真人秀类别列表，null为全部 |
| gender | Integer | false | 真人秀的性别，1-男，2-女,null为全部 |

**输出示例：**

{

"data":[

{

"category":1,

"detail":"detail content",

"downloadUrl":"http://myspace02.weather.com.cn/realShow/3ff92ca1fbb141039c8e47457ac94fe9.zip",

"downloads":123,

"gender":2,

"id":1,

"isRecommended":false,

"isSticked":true,

"link":"link content",

"name":"name test",

"overview":"over view",

"thumbnailUrl":"http://myspace01.weather.com.cn/realShow/bad2708e039d44cbab555744af3cc123.jpg",

"version":"1.0.1"

}

],

"status":"SUCCESS",

"timestamp":"2013-08-22 17:40:48"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 真人秀列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| category | String | 类别 |
| detail | String | 详情 |
| downloadUrl | String | 资源包下载地址 |
| downloads | String | 已下载量 |
| gender | Integer | 性别，1-男，2-女 |
| id | String | ID |
| isRecommended | String | 是否推荐，0：不推荐；1：推荐 |
| isSticked | String | 是否置顶， 0：不置顶；1：置顶 |
| link | String | 外部链接 |
| name | String | 名字 |
| overview | String | 简单说明 |
| thumbnailUrl | String | 缩略图下载地址 |
| version | String | 真人秀版本，格式：\*.\*.\*，如1.0.0 |

### getShowDetail

**接口描述：**获取指定的真人秀信息

**接口返回类型：**JSONObject

**输入示例：**

String id = "1";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| id | String | true | 真人秀id |

**输出示例：**

{

"data":"detail content",

"status":"SUCCESS",

"timestamp":"2013-08-22 17:10:46"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | String | 真人秀信息 |

### updateShow

**接口描述：**获取真人天气秀更新信息

**接口返回类型：**JSONObject

**输入示例：**

JSONObject object = new JSONObject();

object.put("id", 1);

object.put("version", "0.1.0");

JSONArray params = new JSONArray();

params.put(object);

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| params | JSONArray | true | 需要更新的真人秀列表 |
| id | String | true | 真人秀id |
| version | String | true | 真人秀的版本 |

**输出示例：**

{

"data":[

{

"downloadUrl":"http://myspace03.weather.com.cn/realShow/3ff92ca1fbb141039c8e47457ac94fe9.zip",

"id":1,

"version":"1.0.1"

}

],

"status":"SUCCESS",

"timestamp":"2013-08-22 17:17:01"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 真人天气秀更新信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| downloadUrl | String | 真人秀下载地址 |
| id | String | 真人秀ID |
| version | String | 真人秀最新版本 |

## 众筹天气相关接口（CrowdWeatherAPI）

### uploadMyWeather

**接口描述：**上传天气信息

**接口返回类型：**JSONObject

**输入示例：**

JSONObject weather = new JSONObject();

loc.put("code", "01");

loc.put("wd", 45);

loc.put("windSpeed", 5.24);

loc.put("temp", 16.5);

loc.put("pressure", 234);

loc.put("hum", 45.3);

File imgFile = new File(dir\_img);

RequestParams params = new RequestParams();

params.put("areaId", "10101");

params.put("lng", "116.323026");

params.put("lat", "39.946291");

params.put("position", "北京市海淀区颐和园路");

params.put("weather", weather.toString());

params.put("imgFile", imgFile);

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| params | RequestParams | true | 上传参数 |
| areaId | String | true | 5位、9/12位区域站点号 |
| lng | String | false | 客户端经度 |
| lat | String | false | 客户端纬度 |
| position | String | true | 位置信息 |
| code | String | true | 天气现象编号 |
| wd | Double | false | 风向，单位：° |
| windSpeed | Double | false | 风速，单位：m/s |
| temp | Double | false | 温度，单位：℃ |
| pressure | Double | false | 气压，单位：Pa |
| hum | Double | false | 湿度，单位：%rh |
| imgFile | File | false | 天气图片，图片<500kb |

**输出示例：**

{

"data": {

"id": 26310

},

"msg": "upload success",

"status": "SUCCESS",

"timestamp": "2014-12-29 12:14:56"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| id | long | 天气信息id |

### getMyWeather

**接口描述：**获取用户上传的天气（需要用户登录）

**接口返回类型：**JSONObject

**输入示例：**

long start = 0L;

long limit = 10L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| start | long | true | 起始条数,起始值为0 |
| limit | long | true | 查询条数 |

**输出示例：**

{

"timestamp": "2015-02-05 16:01:52",

"data": [{

"position": "北京市海淀区颐和园路",

"img":"http://myspace03.weather.com.cn/5f3f66b4ff00494eb6ec066f8abc0646.jpg",

"id": 1421926683999

}],

"status": "SUCCESS",

"msg": "get weather successfully"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 天气信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 天气消息ID |
| position | String | 地址信息 |
| img | String | 天气图片地址 |

### deleteMyWeather

**接口描述：**删除用户上传的天气

**接口返回类型：**JSONObject

**输入示例：**

long id = 1421926683999L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| id | long | true | 天气消息id |

**输出示例：**

{

"timestamp": "2015-02-05 16:14:06",

"status": "SUCCESS",

"msg": "delete sucess!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

### getStatWeather

**接口描述：**获取众筹天气统计信息

**接口返回类型：**JSONObject

**输入示例：**

String areaId = "10101";

Double offset = 2.5;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| areaId | String | true | 5位省ID、9/12位城市ID |
| offset | Double | true | 时间偏移量,单位小时 |

**输出示例：**

{

"timestamp": "2015-02-05 16:21:10",

"data": [{

"count": 2,

"code": "01"

}],

"status": "SUCCESS",

"msg": "get statistic successfully"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 统计信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| code | String | 天气现象编号 |
| count | Long | 人数 |

### getCrowdWeather

**接口描述：**获取众筹天气信息（包含不含图片天气信息）

**接口返回类型：**JSONObject

**输入示例：**

String areaId = "10101";

String code = "01";

Double offset = 2.5;

long start = 0;

long limit = 10;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| areaId | String | true | 5位省ID、9/12位城市ID |
| code | String | false | 天气现象编号，为null时获取所有天气现象信息 |
| offset | Double | true | 时间偏移量,单位小时 |
| start | long | true | 起始条数,起始值0 |
| limit | long | true | 查询条数 |

**输出示例：**

{

"timestamp": "2015-04-01 17:32:38",

"data": {

"total": 65,

"records": [{

"position": "北京市海淀区南大街42号",

"img": "http://myspace03.weather.com.cn/a1b5b114450f4f2b9120a76c74b4ce92.jpg",

"createTime": 1427857324193,

"lng": 116.335824,

"code": "01",

"imageSuffix": "jpg",

"lat": 39.953259

}]

},

"status": "SUCCESS",

"msg": "get detail successfully"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONObject | 天气信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| total | int | 信息总条数 |
| records | JSONArray | 信息内容 |

records属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| position | String | 地址信息 |
| img | String | 天气图片地址 |
| imageSuffix | String | 图片文件后缀。 |
| createTime | Long | 天气信息上传时间 |
| code | String | 天气现象编号 |
| lng | Double | 经度 |
| lat | Double | 纬度 |

## 微视频相关接口（VideoAPI）

### uploadVideo

**接口描述：**上传微视频信息。用户登录后上传的视频可通过getMyVideos接口获取，否则不能获取。

**接口返回类型：**JSONObject

**输入示例：**

File file = new File(dir\_video);

File thumbnail = new File(dir\_img);

RequestParams params = new RequestParams();

params.put("title", "我的视频");

params.put("desc", "视频的详细描述");

params.put("category", "2");

params.put("file", file);

params.put("thumbnail", thumbnail);

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| params | RequestParams | true | 上传参数 |
| title | String | true | 视频的标题 |
| desc | String | false | 视频的详细描述 |
| category | String | true | 视频分类，字典值，组名：video\_category，1、天气信使（COURIER）；2、我的天气日记(JOURNAL)；3、天气主播祝福(BLESSING)； |
| file | File | true | 上传文件，一次只能上传一个文件，目前支持的视频格式：mp4, flv, rmvb, rm, mkv, avi, mpg, mpeg, 3gp |
| thumbnail | File | true | 视频文件的缩略图（封面），目前支持的格式有：jpg, jpeg, png, gif |

**输出示例：**

{

"timestamp": "2015-02-10 12:25:45",

"data": {

"id": 419,

"thumb": "http:\/\/video.tianqi.cn\/9b53f86c494f4108a7b62762a3b1f697.jpg",

"url": "http:\/\/video.tianqi.cn\/9b53f86c494f4108a7b62762a3b1f697.mp4"

},

"status": "SUCCESS",

"msg": "upload success!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONObject | 视频信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | String | 视频信息id |
| thumb | String | 视频封面地址 |
| url | String | 视频地址 |

### getNewVideos

**接口描述：**获取最新的主播视频列表

**接口返回类型：**JSONObject

**输入示例：**

long start = 0L;

long limit = 10L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| start | int | true | 分页参数，起始记录数 |
| limit | int | true | 分页参数，每页记录数 |

**输出示例：**

{

"timestamp": "2015-02-10 12:32:37",

"data": [{

"id": 419,

"createTime": 1423449963815,

"title": "视频的标题",

"category": {

"key": 1,

"disp": "天气信使"

},

"desc": "视频的详细描述",

"thumb": "http:\/\/video.tianqi.cn\/9b53f86c494f4108a7b62762a3b1f697.jpg",

"url": "http:\/\/video.tianqi.cn\/9b53f86c494f4108a7b62762a3b1f697.mp4"

}],

"status": "SUCCESS",

"msg": "success!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 主播视频信息列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 视频信息ID |
| createTime | String | 视频信息创建时间 |
| title | String | 视频信息标题 |
| desc | String | 视频信息描述 |
| thumb | String | 视频封面地址 |
| url | String | 视频地址 |
| category | JSONObject | 视频类别信息 |

category属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| key | int | 视频类别key |
| disp | String | 视频类别名 |

### getVideos

**接口描述：**获取主播微视频列表，列表按照时间顺序倒叙排列

**接口返回类型：**JSONObject

**输入示例：**

int category = 2;

long start = 0L;

long limit = 10L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| category | int | true | 视频的类别（参考uploadVideo） |
| start | long | true | 分页参数，起始记录数 |
| limit | long | true | 分页参数，每页记录数 |

**输出示例：**

{

"timestamp": "2015-02-10 12:45:23",

"data": [{

"id": 419,

"createTime": 1423449963815,

"title": "视频的标题",

"desc": "视频的详细描述",

"thumb": "http:\/\/video.tianqi.cn\/9b53f86c494f4108a7b62762a3b1f697.jpg",

"url": "http:\/\/video.tianqi.cn\/9b53f86c494f4108a7b62762a3b1f697.mp4"

}],

"status": "SUCCESS",

"msg": "success!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 主播视频信息列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 视频信息ID |
| createTime | String | 视频信息创建时间 |
| title | String | 视频信息标题 |
| desc | String | 视频信息描述 |
| thumb | String | 视频封面地址 |
| url | String | 视频地址 |

### getMyVideos

**接口描述：**获取用户上传的微视频列表，列表按照时间顺序倒叙排列,需用户登录

**接口返回类型：**JSONObject

**输入示例：**

int category = 2;

long start = 0L;

long limit = 10L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| category | int | true | 视频的类别（参考uploadVideo） |
| start | long | true | 分页参数，起始记录数 |
| limit | long | true | 分页参数，每页记录数 |

**输出示例：**

{

"timestamp": "2015-02-10 15:09:55",

"data": [{

"id": 419,

"createTime": 1423449963815,

"title": "视频的标题",

"desc": "视频的详细描述",

"userId": 1423448235223670,

"thumb": "http:\/\/video.tianqi.cn\/92ce87eb7fda49c79eca9aefac758b8d.jpg",

"url": "http:\/\/video.tianqi.cn\/92ce87eb7fda49c79eca9aefac758b8d.mp4"

}],

"status": "SUCCESS",

"msg": "success!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 主播视频信息列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 视频信息ID |
| createTime | String | 视频信息创建时间 |
| title | String | 视频信息标题 |
| desc | String | 视频信息描述 |
| thumb | String | 视频封面地址 |
| url | String | 视频地址 |
| userId | long | 上传此视频的用户id |

### getAnchors

**接口描述：**获取所有的主播列表

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"timestamp": "2015-02-10 12:42:01",

"data": [{

"id": 1,

"createTime": 1423449963815,

"schedule": "星期四 星期五",

"videoThumb": "http:\/\/video.tianqi.cn\/76bf1b17fe274357969e84b78938e983.jpg",

"name": "张小娴^^",

"comment": "你是谁呀？",

"portrait": "http:\/\/video.tianqi.cn\/d7b0aaa12a714d54bf2411c63317575d.jpg"

}],

"status": "SUCCESS",

"msg": "get anchor successfully!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 主播视频信息列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 主播ID |
| name | String | 主播名字 |
| portrait | String | 主播的头像url地址 |
| videoThumb | String | 主播默认的视频缩略图 |
| schedule | String | 主播的日程安排 |

### lottery

**接口描述：**抽奖，需用户登录

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"timestamp": "2015-02-10 12:49:46",

"data": {

"id": 10

},

"status": "SUCCESS",

"msg": "Hey guy, You are an lucky dog, congratulation!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONArray | 主播视频信息列表 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | long | 获奖的id，后续选主播、填写祝福语，提交祝福录制表单信息时，需要该参数 |

### uploadWish

**接口描述：**客户端将选定的主播、祝福人、祝福语以及对想对主播说的话等信息提交给后台

**接口返回类型：**JSONObject

**输入示例：**

long id = 10L;

long anchorId = 1L;

String from = "lilith";

String wish = "愿天下有情人终成圈鼠";

String toAnchor = "丹丹一定要给我录好了哦！";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| id | long | true | 获奖id |
| anchorId | long | true | 主播id |
| from | String | true | 祝愿人 |
| wish | String | true | 祝福语 |
| toAnchor | String | false | 祝愿人想对主播说的话 |

**输出示例：**

{

"timestamp": "2015-02-10 12:51:25",

"status": "SUCCESS",

"msg": "order successfully!"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

## 普通接口（CommonAPI）

### reportError

**接口描述：**上传错误日志

**接口返回类型：**JSONObject

**输入示例：**

String osVersion = "android\_4.0";

String softVersion = "2.0.4";

String metadataVersion = "7";

String provinceVersion = "10101";

String stageVersion = "pub";

String deviceInfo = "GT-9000|480\*800";

String traceInfo = "Error:systemRrror";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| osVersion | String | true | 操作系统版本（os\_version） |
| softVersion | String | true | 客户端版本 |
| metadataVersion | String | false | 元数据版本 |
| provinceVersion | String | false | 省级版ID |
| stageVersion | String | false | 专业版/大众版(MetaData) |
| deviceInfo | String | true | 设备信息(设备型号|分辨率) |
| traceInfo | String | true | 错误日志 |

**输出示例：**

{

"msg":"log upload success",

"status":"SUCCESS",

"timestamp":"2013-01-10 14:53:09"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

### uploadFeedBack

**接口描述：**发送反馈数据

**接口返回类型：**JSONObject

**输入示例：**

String content = "中国天气通接口测试";

String email = null;

String tel = null;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| content | String | true | 内容 |
| email | String | false | 邮箱 |
| tel | String | false | 电话 |

**输出示例：**

{

"msg":"upload success",

"status":"SUCCESS",

"timestamp":"2013-06-24 11:09:22"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

### updateVersion

**接口描述：**客户端版本升级

**接口返回类型：**JSONObject

**输入示例：无**

**输入参数：无**

**输出示例：**

{

"status":"SUCCESS",

"msg":"newversionpublished",

"data":{

"latestVersion":"1.0",

"url":"http://download.weather.com.cn/general/weather.apk"

},

"timestamp":"2012-09-03 18:03:25"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| data | JSONObject | 最新客户端信息 |
| status | String | SUCCESS需要更新/FAILURE不需要更新 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| latestVersion | String | 最新版本 |
| url | String | 最新版本下载地址 |

### updatePlugin

**接口描述：**插件服务升级

**接口返回类型：**JSONObject

**输入示例：**

String id = "215";

String version = "2.0.0";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| id | string | true | 插件服务的ID |
| version | string | true | 插件服务的版本 |

**输出示例：**

{

"msg":" new version published",

"status":"SUCCESS",

"data":{

"latestVersion":"1.0",

"url":"http://download.weather.com.cn/weather.apk"

},

"timestamp":"2013-08-09 14:10:35"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| data | JSONObject | 最新客户端信息 |
| status | string | SUCCESS需要更新/FAILURE不需要更新 |
| msg | string | 服务器返回信息 |
| timestamp | string | 服务器返回时间 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| latestVersion | string | 最新客户端版本 |
| url | string | 客户端下载地址 |

### getSplash

**接口描述：**获取闪屏界面

**接口返回类型：**JSONObject

**输入示例：**

String excludeId = "1";

String cityId = "101010101";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| excludeId | String | false | 上次下载的图片id，从结果中过滤数据，如果该字段为null，不过滤该id |
| cityId | String | true | 闪屏图片所属城市id，全国为0（定位失败时，使用全国） |

**输出示例：**

{

"data":{

"downloadUrl":"http://myspace01.weather.com.cn/3ca4c33ea9b14a889aaa5f9de9f5dc0c.jpg",

"id":1

},

"msg":"new splash",

"status":"SUCCESS",

"timestamp":"2013-08-21 17:01:50"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |
| data | JSONObject | 闪屏图片信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| downloadUrl | String | 图片下载地址 |
| id | String | 图片ID |
| name | String | 名称 |
| link | String | 外链地址 |

### getAds

**接口描述：**获取广告信息

**接口返回类型：**JSONObject

**输入示例：**

JSONArray adCodes = new JSONArray();

ids.put("aa");

ids.put("bb");

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| adCodes | JSONArray | true | 广告编码 |

**输出示例：**

{

"data":[

{

"adName":"你好",

"adType":1,

"adUrl":"http://myspace01.weather.com.cn/7ab7f8918a7b44e4bebc89444c7da2f6\_300\_1.jpg",

"adcode":"aa",

"appKey":"terywutyu",

"createTime":1419329312474,

"effects":2,

"endTime":1419329312474,

"id":22,

"sourceUrl":"www.weather.com",

"teladUrl":"www.weather.com"

},

{

"adName":"你好",

"adType":1,

"adUrl":"http://myspace01.weather.com.cn/7ab7f8918a7b44e4bebc89444c7da2f6\_300\_1.jpg",

"adcode":"bb",

"appKey":"terywutyu",

"createTime":1419329322862,

"effects":3,

"endTime":1419329322862,

"id":23,

"sourceUrl":"www.weather.com",

"teladUrl":"www.weather.com"

}

],

"status":"SUCCESS",

"timestamp":"2014-12-24 09:43:54"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息（status为FAILURE时返回的错误信息） |
| timestamp | String | 时间 |
| data | JSONObject | 广告信息 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | Long | 广告id |
| adcode | String | 广告编码 |
| appKey | String | 应用编码 |
| adName | String | 广告名称 |
| adType | Integer | 广告类型（1：图片广告；2：视频广告）暂时只支持图片 |
| adUrl | String | 本地访问地址，页面或者图片位置等 |
| teladUrl | String | 异地访问广告url，如为空，同adUrl |
| localText | String | 本地访问广告的说明 |
| telText | String | 异地访问广告的说明，如为空，同localText |
| sourceUrl | String | 本地访问广告的跳转链接，支持图片或文字的广告链接 |
| telSourceUrl | String | 异地访问广告的跳转链接，如为空，同sourceUrl |
| effects | Integer | 本地访问广告的特效设置（1,闪屏；2,提示new；3,高亮；4,置顶） |
| telEffects | Integer | 异地访问广告的特效设置，如为空，同effects（1,闪屏；2,提示new；3,高亮；4,置顶） |
| createTime | String | 创建时间 |
| endTime | String | 广告截止时间 |

### uploadQuestionnaire

**接口描述：**发送调查问卷数据

**接口返回类型：**JSONObject

**输入示例：**

String content = "中国天气通接口测试";

String email = 772109280@qq.com;

String cellPhone = 15210861969;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| content | String | true | 内容 |
| email | String | true | 邮箱 |
| cellPhone | String | true | 电话 |

**输出示例：**

{

"msg":"message upload success",

"status":"SUCCESS",

"timestamp":"2013-06-24 11:09:22"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

### getNotice

**接口描述：**接收通知

**接口返回类型：**JSONObject

**输入示例：**

long start = 0L;

long limit = 10L;

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| start | long | true | 起始条数,起始值0 |
| limit | long | true | 查询条数 |

**输出示例：**

{

"data": [{

"id": 13,

"title": "请关注天气情况",

"content": "文字描述",

"url": "http://app.weather.com.cn",

"img": "http://app.weather.com.cn/new.jpg",

"createTime": 1423131396132

}],

"msg": " get nocitce successfully ",

"status": "SUCCESS",

"timestamp": "2015-01-16 12:14:56"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

data属性

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| id | Long | 通知的id |
| title | String | 标题 |
| content | String | 内容 |
| url | String | 跳转URL |
| img | String | 图片地址 |
| createTime | Long | 通知生成时间 |

### setDailyPushTime

**接口描述：**设置早晚推送

**接口返回类型：**JSONObject

**输入示例：**

int t1 = 25200;

int t2 = -1;

String pushCityId = " 10101 ";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| t1 | int | true | 第一次推送时间，单位为秒，一天为60\*60\*24,25200为7:00，-1表示不启用 |
| t2 | int | true | 第二次推送时间 |
| pushCityId | String | true | 推送城市 |

**输出示例：**

{

"msg": " 设置成功 ",

"status": "SUCCESS",

"timestamp": "2015-01-16 12:14:56"

}

**输出参数：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| status | String | SUCCESS成功/FAILURE失败 |
| msg | String | 服务器返回信息 |
| timestamp | String | 时间 |

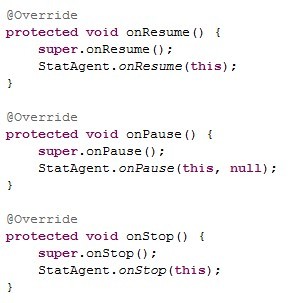
## 信息统计

### StatAgent

一、页面统计使用方法:

在每个Activity的onResume方法中调用 StatAgent.onResume(), onPause方法中调用StatAgent.onPause(),onStop方法中调用StatAgent.onStop ()

Activity页面：



注意：

1. 确保在所有的需要统计的activity中都调用 StatAgent.onResume()、StatAgent.onPause()和StatAgent.onStop()方法。
2. 注意如果您的Activity之间有继承或者控制关系请不要同时在父和子Activity中重复添加onPause、onResume和onStop方法，否则会造成重复统计(eg.使用TabHost、TabActivity、ActivityGroup时)。
3. 在<application>标签中,添加:

<service android:name="cn.com.weather.util.TrafficStatsService" />

1. 在省级本地服务内容页面StatAgent.onPause(this,serviceId)的参数serviceId为服务ID

二、自定义事件统计使用方法:

1.事件次数统计：onEvent

2.事件时长统计：事件开始时调用onEventBegin，事件结束时调用onEventEnd

注意：

1.事件次数统计和时长统计的eventId不能相同

#### onResume

**接口描述：**在每个Activity的onResume方法中调用

**接口返回类型：**无

**输入示例：无**

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| context | Activity | true | 当前Activity的引用,这里请不要将全局的application context传入 |

**输出示例：无**

**输出参数：无**

#### onPause

**接口描述：**在每个Activity的onPause方法中调用。serviceId用于区分用于区分同一Activity显示不同内容，例如：WebActivity（由webView构成的页面）。此种情况外serviceId为null

**接口返回类型：**无

**输入示例：无**

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| serviceId | String | false | serviceId用于区分同一Activity显示不同内容，例如：WebActivity,展示服务时serviceId为服务的ID; 非省级服务，即无服务ID的服务（如天气视频），其serviceId为服务名称的小写全拼（tianqishipin）。  除此之外的正常Activity页面serviceId为null |

**输出示例：无**

**输出参数：无**

#### onStop

**接口描述：**在每个Activity的onStop方法中调用

**接口返回类型：**无

**输入示例：无**

**输入参数：无**

**输出示例：无**

**输出参数：无**

#### onEvent

**接口描述：**自定义事件次数统计

**接口返回类型：**无

**输入示例：无**

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| eventId | String | true | 事件ID |

**输出示例：无**

**输出参数：无**

#### onEventBegin

**接口描述：**自定义事件时长统计，在事件开始时调用，eventId不能和onEvent中eventId相同

**接口返回类型：**无

**输入示例：无**

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| eventId | String | true | 事件ID |

**输出示例：无**

**输出参数：无**

#### onEventEnd

**接口描述：**自定义事件时长统计，在事件结束时调用，eventId必须和onEventBegin中eventId相同。

**接口返回类型：**无

**输入示例：无**

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| eventId | String | true | 事件ID |

**输出示例：无**

**输出参数：无**

#### setSettings

**接口描述：**用户的设置信息

**接口返回类型：**无

**输入示例：**

JSONObject settings = new JSONObject();

settings.put("receiveWarning", true);

settings.put("pushCityMsg", true);

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| settings | JSONObject | true | 用户设置的信息 |
| receiveWarning | boolean | true | 是否接收预警信息 |
| pushCityMsg | boolean | true | 是否接收推送城市消息。 |

**输出示例：无**

**输出参数：无**

#### addAdClickedInfo

**接口描述：**用户点击的广告，点击广告时调用

**接口返回类型：**无

**输入示例：**

String adId = "1";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| adId | String | true | 广告ID |

**输出示例：无**

**输出参数：无**

#### addSharedInfo

**接口描述：**用户分享的信息

**接口返回类型：**无

**输入示例：**

JSONObject sharedInfo = new JSONObject();

sharedInfo.put("channel", Constants.Channel.SMS);

sharedInfo.put("content", Constants.ShareType.Weather);

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| sharedInfo | JSONObject | true | 用户短信分享的信息 |
| channel | String | true | 分享渠道SMS（Channel的枚举类型） |
| content | String | true | 分享的类型（ShareType的枚举类型） |

Channel：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| SMS | 短信 |  |
| SINAWEIBO | 新浪微博 |  |
| TENCWEIBO | 腾讯微博 |  |
| RENRWEB | 人人网 |  |
| QZONE | QQ空间 |  |
| WEIXIN | 微信 |  |
| EMAIL | Email |  |

ShareType：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数含义** | **备注** |
| WEATHER | 天气 |  |
| WARNING | 预警（推送信息） |  |

**输出示例：无**

**输出参数：无**

#### addCalledPhone

**接口描述：**用户拨打的客户端中的电话，拨打电话时调用

**接口返回类型：**无

**输入示例：**

String phoneNum = "10086";

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| phoneNum | String | true | 电话号码 |

**输出示例：无**

**输出参数：无**

## 推送服务

### PushAgent

推送注意事项：

1.在<application>标签中,添加:

<service

android:name="com.smartapi.pn.client.NotificationService"

android:exported="true"

android:process=":pushservice" >

<intent-filter>

<action android:name="com.smartapi.pn.client.NotificationService" />

</intent-filter>

</service>

<service android:name="com.smartapi.pn.client.IntentService"/>

<receiver android:name="com.smartapi.pn.client.NotificationReceiver">

<intent-filter >

<action android:name="【应用包名】.SHOW\_NOTIFICATION"/>

</intent-filter>

</receiver>

<receiver android:name="com.smartapi.pn.client.ConnectivityReceiver">

<intent-filter >

<action android:name="android.net.conn.CONNECTIVITY\_CHANGE"/>

</intent-filter>

<intent-filter>

<action android:name="android.intent.action.PACKAGE\_REMOVED" />

<data android:scheme="package" />

</intent-filter>

</receiver>

#### getInstance

**接口描述：**获取PushAgent实例

**接口返回类型：**PushAgent对象

**输入示例：无**

**输入参数：无**

**输出示例：无**

**输出参数：无**

#### setNotifier

**接口描述：**设置推送回调接口

**接口返回类型：**无

**输入示例：无**

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **参数含义** |
| notifier | Notifier | true | Notifier的实现类 |

**输出示例：无**

**输出参数：无**

#### enable

**接口描述：**启动推送服务。

**接口返回类型：**无

**输入示例：无**

**输入参数：无**

**输出示例：无**

**输出参数：无**

### Notifier

#### notify

**接口描述：**推送回调接口

**接口返回类型：**无

**输入示例：无**

**输入参数：无**

**输出示例：无**

**输出参数：无**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **参数含义** |
| type | String | 消息类型，1普通消息，2带链接的消息 |
| id | Int | 消息ID |
| title | String | 消息标题 |
| content | String | 消息内容 |
| detail | String | 消息连接，和type联合使用 |

# 图片下载

图片下载适用于用户头像（GetPortraitAsyncTask）和实景天气（GetSceneAsyncTask）接口。



## 原始图像下载

举例：http://myspace01.weather.com.cn/ 96508828e3bc4b25a23c0847e8e195bd.jpg

访问方式：GET

说明：

取得上传的原始图像（没有经过任何处理）。

## 固定分辨率图像下载

举例：http://myspace01.weather.com.cn/ 96508828e3bc4b25a23c0847e8e195bd\_100x100.jpg

访问方式：GET

说明：

取得上传的图像（图像会根据给定的分辨率进行缩放）。

分辨率在fileName中设置，在文件名称和文件后缀之间添加分辨率信息，格式为“\_[weight]x[height]”，其中[weight]和[height]是两位到四位的数字。

## 固定分辨率宽度图像下载

举例：http://myspace01.weather.com.cn /96508828e3bc4b25a23c0847e8e195bd\_100\_0.jpg

访问方式：GET

说明：

取得上传的图像（图像宽度根据给定的数值进行缩放，高度根据宽度的缩放比例进行缩放）。

分辨率在fileName中设置，在文件名称和文件后缀之间添加宽度缩放策略，格式为“\_[weight]\_0”，其中[weight]是两位到四位的数字。

## 固定分辨率高度图像下载

举例：http://myspace01.weather.com.cn /96508828e3bc4b25a23c0847e8e195bd\_100\_1.jpg

访问方式：GET

说明：

取得上传的图像（图像高度根据给定的数值进行缩放，宽度根据高度的缩放比例进行缩放）。

分辨率在fileName中设置，在文件名称和文件后缀之间添加高度缩放策略，格式为“\_[height]\_1”，其中[height]是两位到四位的数字。

# 音频流下载



## 原始音频流下载

举例：http://myspace01.weather.com.cn/ voice/f8b2a988062c45ff97330c08d5976da0.amr

访问方式：GET

说明：

取得上传的原始音频流（没有经过任何处理）。

# 编码表

**天气现象编码表**

|  |  |  |
| --- | --- | --- |
| 编号 | 中文名称 | 英文名称 |
| 00 | 晴 | Sunny |
| 01 | 多云 | Cloudy |
| 02 | 阴 | Overcast |
| 03 | 阵雨 | Shower |
| 04 | 雷阵雨 | Thundershower |
| 05 | 雷阵雨伴有冰雹 | Thundershower with hail |
| 06 | 雨夹雪 | Sleet |
| 07 | 小雨 | Light rain |
| 08 | 中雨 | Moderate rain |
| 09 | 大雨 | Heavy rain |
| 10 | 暴雨 | Storm |
| 11 | 大暴雨 | Heavy storm |
| 12 | 特大暴雨 | Severe storm |
| 13 | 阵雪 | Snow flurry |
| 14 | 小雪 | Light snow |
| 15 | 中雪 | Moderate snow |
| 16 | 大雪 | Heavy snow |
| 17 | 暴雪 | Snowstorm |
| 18 | 雾 | Foggy |
| 19 | 冻雨 | Ice rain |
| 20 | 沙尘暴 | Duststorm |
| 21 | 小到中雨 | Light to moderate rain |
| 22 | 中到大雨 | Moderate to heavy rain |
| 23 | 大到暴雨 | Heavy rain to storm |
| 24 | 暴雨到大暴雨 | Storm to heavy storm |
| 25 | 大暴雨到特大暴雨 | Heavy to severe storm |
| 26 | 小到中雪 | Light to moderate snow |
| 27 | 中到大雪 | Moderate to heavy snow |
| 28 | 大到暴雪 | Heavy snow to snowstorm |
| 29 | 浮尘 | Dust |
| 30 | 扬沙 | Sand |
| 31 | 强沙尘暴 | Sandstorm |
| 32 | 浓雾 | Dense Foggy |
| 33 | 雪 | Snow |
| 49 | 强浓雾 |  |
| 53 | 霾 | Haze |
| 54 | 中度霾 |  |
| 55 | 重度霾 |  |
| 56 | 严重霾 |  |
| 57 | 大雾 |  |
| 58 | 特强浓雾 |  |
| 99 | 无 | Unknown |

**注**：天气现象图标文件名与天气现象编号对应，如“晴”的图标文件名00.jpg，图标扩展名统一为jpg

**精细化预报天气现象编码表**

|  |  |  |
| --- | --- | --- |
| 编号 | 中文名称 | 英文名称 |
| 1 | 晴 | fine |
| 2 | 少云 | partly cloudy |
| 3 | 多云 | cloudy |
| 4 | 阴天 | overcast |
| 5 | 小雨 | light rain |
| 6 | 中雨 | moderate rain |
| 7 | 大雨 | Heavy rain |
| 8 | 暴雨 | Torrential rain |
| 9 | 大暴雨 | Severe storm |
| 10 | 特大暴雨 | Extra torrential rain |
| 11 | 小雪 | Light snow |
| 12 | 中雪 | Moderate snow |
| 13 | 大雪 | Heavy |
| 14 | 暴雪 | snowstorm |
| 15 | 大暴雪 | Heavy snowstorm |
| 16 | 特大暴雪 | Heavy snow |
| 17 | 雨夹雪 | Rain and snow |

**风力风向编码表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 风向编号 | 中文名称 | 英文名称 | 风力编号 | 中文名称 | 英文名称 |
| 0 | 无持续风向 | No wind | **0** | 微风 | <10m/h |
| 1 | 东北风 | Northeast | **1** | 3-4级 | 10~17m/h |
| 2 | 东风 | East | **2** | 4-5级 | 17~25m/h |
| 3 | 东南风 | Southeast | **3** | 5-6级 | 25~34m/h |
| 4 | 南风 | South | **4** | 6-7级 | 34~43m/h |
| 5 | 西南风 | Southwest | **5** | 7-8级 | 43~54m/h |
| 6 | 西风 | West | **6** | 8-9级 | 54~65m/h |
| 7 | 西北风 | Northwest | **7** | 9-10级 | 65~77m/h |
| 8 | 北风 | North | **8** | 10-11级 | 77~89m/h |
| 9 | 旋转风 | Whirl wind | **9** | 11-12级 | 89~102m/h |

**注：**风力单位是mile

**预警信号类别编码表**

|  |  |
| --- | --- |
| **预警信号类别** | **编号** |
| 台风 | 01 |
| 暴雨 | 02 |
| 暴雪 | 03 |
| 寒潮 | 04 |
| 大风 | 05 |
| 沙尘暴 | 06 |
| 高温 | 07 |
| 干旱 | 08 |
| 雷电 | 09 |
| 冰雹 | 10 |
| 霜冻 | 11 |
| 大雾 | 12 |
| 霾 | 13 |
| 道路结冰 | 14 |

**预警信号级别编码表**

|  |  |
| --- | --- |
| **预警信号级别** | **编号** |
| 蓝色 | 01 |
| 黄色 | 02 |
| 橙色 | 03 |
| 红色 | 04 |

# 数据更新频率表

|  |  |
| --- | --- |
| **数据类型** | **更新频率** |
| 预警 | 不定期更新，为保证预警的时效性，建议程序启动后每隔10分钟进行一次轮循 |
| 实况 | 逐小时更新 |
| 常规预报(24小时) | 每天发布3次：  发布时间分别为：08时、11时、18时 |
| 3小时精细化预报 | 每天发布2次：  发布时间分别为：08时、20时 |
| 6小时精细化预报 | 每天发布3次：  发布时间分别为：08时、11时、18时 |
| 指数 | 每天发布3次：  发布时间分别为：08时、11时、18时 |