环境数据接口文档
V1.0

目录

1.对接方式

本协议统一使用 **POST 方法**，通过开放 API 接口方式进行数据接入，接入 API 地址如下：

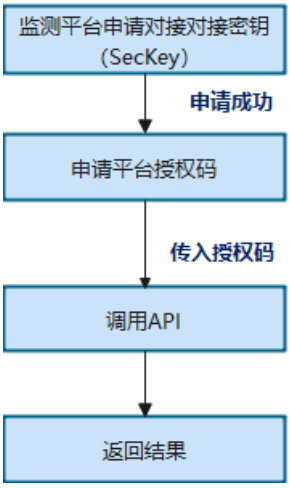
- http(s):端口号//BaseURL/openapi/版本号/接口地址
- BaseURL：数据平台开放 API 地址或者域名

  测试地址为: http://112.64.195.226:8899/api/
- 版本号：当前使用 api 的版本，如 v1，v2 等

  当前为:v1
- 接口地址：具体接入 API 方法

  例如 http://112.64.195.226:8899/api/openapi/v1/auth/getToken

考虑到平台数据接入安全性，第三方平台在接入数据平台时需要先进行授权码，在获取到授权码后方可进行后续接口操作。



（图）接入流程

- 加密方式

平台数据默认采用加密算法 AES 加密方式,加密模式为 CBC,补码方式为 PKCS5Padding，偏移量为 16，采用平台提供的 appSec 作为数据的加/解密密钥

2.对接内容

序号	接入内容	接入要求
1	认证接口	根据 appId 和 appSec 获取 token。
2	环境数据推送	将环境数据推送到平台。

3.API 参考

序号	API 地址	接口说明
1	/auth/getToken	认证接口
2	/environment/saveEnvData	获取实名制数据

3.1.Token 获取接口：

接口地址

/auth/getToken

请求参数(JSON 格式)：

字段名	必填	类型	说明
appId	是	String	厂商 ID（平台提供）
signature	是	String	使用 appSec 作为密钥进行 AES 加密，加密内容为:appId+时间戳，appSec 联系管理员提供。

请求示例：

```
{
  "appId": " testid ",
  "signature": "6bdc8cbf2aba5a0303e97321c7f1bd1047badf7239f8b0ad15492037ffdc3e24"
}
```

加密参考代码(Java 版本)

```
// 密钥
public static String key =””;

private static String charset = "utf-8";
// 偏移量
private static int offset = 16;

// 加密器类型:加密算法为 AES,加密模式为 CBC,补码方式为 PKCS5Padding
private static String transformation = "AES/CBC/PKCS5Padding";

// 算法类型：用于指定生成 AES 的密钥
private static String algorithm = "AES";
```

```
/**
 * 加密
 *
 * @param content 需要加密的内容
 * @param key     加密密码
 * @return
 */
public static String encrypt(String content, String key) {
    try {
        //构造密钥
        SecretKeySpec skey = new SecretKeySpec(key.getBytes(),
algorithm);
        //创建初始向量 iv 用于指定密钥偏移量(可自行指定但必须为 128 位)，因为 AES 是
分组加密，下一组的 iv 就用上一组加密的密文来充当
        IvParameterSpec iv = new IvParameterSpec(key.getBytes(), 0,
offset);
        //创建 AES 加密器
        Cipher cipher = Cipher.getInstance(transformation);
        byte[] byteContent = content.getBytes(charset);
        //使用加密器的加密模式
        cipher.init(Cipher.ENCRYPT_MODE, skey, iv);
        // 加密
        byte[] result = cipher.doFinal(byteContent);
        //使用 BASE64 对加密后的二进制数组进行编码
        return new Base64().encodeAsString(result);
    } catch (Exception e) {
        logger.info("", e);
    }
    return null;
}
```

返回参数说明

参数名	类型	说明
success	boolean	是否成功
message	String	备注信息
code	Integer	200 成功，其他失败
timestamp	long	时间戳
result	Object	主体信息，本接口该参数即为 token 字符串
requestId	String	

返回示例：


```
{
  "success": true,
  "message": "success",
  "code": 200,
  "timestamp": 1651712726863,
  "result": "c281448e-4778-450d-86d0-cbbf06ca1e45",
  "requestId": "d501089a-3b42-452c-b940-08ef15d280f7"
}
```


3.2.环境数据推送接口：

接口地址

`/environment/saveEnvData`

请求参数(JSON 格式)：

没有某个字段可以传空字符串，但是字段需要传过来。

字段名	必填	类型	说明
token	是	String	Token，通过认证接口获取
deviceNo	是	String	设备编号
superviseNumber	是	String	监督编号(报监编号)
deviceDesc	是	String	设备描述
pm10	是	Double	Pm10，单位 mg/m³
pm25	是	Double	Pm2.5，单位 mg/m³
tsp	否	Double	总悬浮颗粒物，单位 mg/m³

noise	是	Double	噪音，单位 db
temp	是	Double	温度，单位℃
windDirection	是	String	风向
windSpeed	是	Double	风速，单位 m/s
recordTime	是	String	数据采集时间（YYYY-MM-DD HH:ii:ss 格式）
airPressure	否	Double	大气压
co	否	Double	一氧化碳，单位 mg/m³
h2s	否	Double	硫化氢
humidity	否	Double	湿度
nh3	否	Double	氨气
no2	否	Double	二氧化氮，单位 mg/m³
so2	否	Double	二氧化硫，单位 mg/m³
voc	否	Double	挥发性有机物

请求示例：


```
{
    "deviceNo": "20210318DG819G9801",
    "deviceDesc": "设备 52",
    "superviseNumber": " GD22003",
    "pm10": 58,
    "pm25": 19.3,
    "temp": 18.9,
    "noise": 43,
    "windDirection": "东",
    "windSpeed": 51,
    "airPressure": "",
    "co": "",
    "h2s": "",
    "humidity": "",
    "nh3": "",
    "no2": "",
    "so2": "",
    "tsp": 60,
    "voc": "",
    "recordTime": "2022-05-04 04:14:53",
    "token": "d3bb124e-db4f-4688-8833-b7df51ab09a1"
}
```


返回参数说明

参数名	类型	说明
success	boolean	是否成功
message	String	备注信息
code	Integer	200 成功，其他失败
timestamp	long	时间戳
result	Object	主体信息

requestId	String	

返回示例：


```
{
    "success": true,
    "message": "环境数据保存成功",
    "code": 200,
    "timestamp": 1651647486869,
    "result": null,
    "requestId": "419dd8ae-5c3f-4270-a389-8519c1739b8d"
}
```