

# 目 录

登录鉴权

设备统计数据

设备管理

    修改设备

    新增设备

    设备类型

    删除设备

    设备详情

    设备信息

    单个设备信息

    设备或车辆转移

车辆管理

    修改车辆

    新增车辆

    车辆类型

    删除车辆

    车辆信息

    车辆详情

    单个车辆信息

实时数据

轨迹数据

远程开锁

    开锁指令

    查询锁状态

远程控制

    开锁密码信息

    修改锁密码

    查询修改静态密码指令执行状态

    开启/关闭动态密码

    查询开启/关闭动态密码指令执行状态

    查询设备授权卡

查询车辆拍摄的最新的照片

主动拍照

查询设备的联网状态

新主动拍照(只支持705C的CAMERA9\_4)

报表查询

状态曲线

开关锁报表

从机记录

报警消息报表

事件消息报表

温湿度曲线报表（新）

油耗报表

加油报告

异常油耗

油耗分析

查看报警/事件/开关锁报表的附件

日里程报告

远程指令集

子锁开锁

子锁列表

远程指令集查询

下发远程指令

下发自定义指令

指令记录查询

下发在途监管指令(BASE66)

JT701设备类型指令类型对象

JT707/JT709A/JT705/JT702/JT301设备类型指令类型对象

GP4000/GP6000/GP6000F设备类型指令类型对象

JT/T808设备类型指令类型对象

自定义类型说明

运单管理

启动/结束监管

车次派发

查询公司车次列表（未完成车次）

查询运单报告（所有）

删除运单

修改运单信息

查询单个运单详情

查询所有路线信息

查询单条路线具体信息

查询单条路线信息,包含分支路线

新增路线

修改路线

围栏管理

批量添加围栏

查询公司下所有围栏及围栏分组信息

修改围栏

批量删除围栏

查询公司下所有围栏分组信息

围栏绑定车辆

公司信息

视频监控

实时视频监控

停止视频监控

司机管理

新增司机

修改司机

删除司机

查询司机列表

# 登录鉴权

## 简要描述：

- 用户验证接口，获取系统Token

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/login>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/login>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FUserName	是	string	用户名
FPassword	是	string	密码 (MD5(32))

## 返回示例：

### 正确时返回：

```
{
```

```

"Result": 200,
"Message": "success",
"FObject": [{
  "FUserName": "admin",
  "FUserGUID": "63C399BD-2A0B-40E9-857D-6DEBF4908F92",
  "FTokenID": "DB7F20D9-84C3-44C3-B05A-06A9C392A189",
  "FExpireTime": "2020-07-30T08:00:00"
}]
}

```

### 错误时返回:

```

{
  "Result": 103,
  "Message": "userName or password is error",
  "FObject": null
}

```

### 返回参数说明:

参数名	类型	说明
FUserName	String	用户名
FUserGUID	String	用户唯一标识
FTokenID	String	身份唯一标识（后面请求接口均需要以此作为身份验证）
FExpireTime	DateTime	账户过期时间（UTC时间，转成北京时间需要+8小时）

### 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误

### 请求示例:

#### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/login";

```

```

//请求参数 ,json格式参数,建议用对象传入
String body = "{UserName:\"admin\",FPassword:\"E10ADC3949BA59ABBE56E057F20F883E\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

### C#(.NET):

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/login";
//请求参数 ,json格式参数,建议用对象传入
string body = "{UserName:\"admin\",FPassword:\"E10ADC3949BA59ABBE56E057F20F883E\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";

```

```
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/login'
data = {
    'FUserName': 'admin',
    'FPassword': 'E10ADC3949BA59ABBE56E057F20F883E'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
return page;
```

# 设备统计数据

## 简要描述：

- 获取设备统计信息，包含子公司设备

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2022-06-06	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Monitor>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Monitor>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAssetStatusData)

## 返回示例：

### 正确时返回：

```
{
```

```

"Result": 200,
"Message": "check token success",
"FObject": {
  "FTotalCount": 212,
  "FOnlineCount": 6,
  "FOfflineCount": 36,
  "FInactiveCount": 165,
  "FMotionCount": 0,
  "FStationaryCount": 6,
  "FIdlingCount": 0,
  "FExpiringCount": 0,
  "FExpiredCount": 5,
  "FUnLockCount": 7,
  "FLockCount": 31,
  "FDoorOpenCount": 0,
  "FDoorCloseCount": 1,
  "FAlarmCount": 1,
  "FLowBatteryCount": 17
}
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FTotalCount	Int	设备总数量，（总数量=离线数量+在线数量+未激活数量+已过期数量）
FOfflineCount	Int	离线设备数量（设备最后接收数据的时间 + 当前公司设置的设备离线时长 < 当前时间，满足上面这个条件则表示设备离线）
FOnlineCount	Int	在线设备数量（非离线设备，则表示在线）
FIdlingCount	Int	怠速中设备数量（设备ACC状态为开，且速度为0，则表示怠速中，此状态主要是在车载设备中用到）

FStationaryCount	Int	停止中设备数量（设备速度为0，则表示停止中）
FMotionCount	Int	运行中设备数量（设备速度大于0，则表示运行中）
FInactiveCount	Int	未激活设备数量（没有联网过的设备数量）
FExpiringCount	Int	即将过期设备数量（过期时间在1个月内）
FExpiredCount	Int	已过期设备数量
FUnLockCount	Int	已开锁数量(当前锁状态为开的统计)
FLockCount	Int	已上锁数量（当前锁状态为关的统计）
FAlarmCount	Int	当前设备报警数量（当前有设备上传报警数据的统计）
FLowBatteryCount	Int	当前低电设备数量（低电阈值小于20%）
FDoorOpenCount	Int	开门数量(当前设备门状态为开的统计，没有门状态设备，可忽略此字段)
FDoorCloseCount	Int	关门数量(当前设备门状态为关的统计，没有门状态设备，可忽略此字段)

## 备注:

- 更多返回错误代码如下：
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 , json格式参数, 建议用对象传入
String body = "{FAction: \"QueryAssetStatusData\", FTOKENID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
```

```

conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryAssetStatusData\",FTokenID:\"3acef045-d302-4
032-b40a-d9ee6c1519cd\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串

```

```
return reader.ReadToEnd();  
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Monitor'  
data = {  
    'FAction': 'QueryAssetStatusData',  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd'  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 设备管理

[修改设备](#)

---

[新增设备](#)

---

[设备类型](#)

---

[删除设备](#)

---

[设备详情](#)

---

[设备信息](#)

---

# 修改设备

## 简要描述：

- 修改设备信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateAdminAsset)
FT_AdminAsset	是	Object	{“FGUID”:设备唯一标识,“FVehicleGUID”:车辆唯一标识 (不传则表示解绑) ,“FAssetID”:设备ID,“FAssetTypeID”:设备类型,“FSIMNumber”:SIM卡号,“FIMEI”:IMEI

			号,"FCustomerID":设备编号,"FDescription":描述,"FAgentGUID":公司唯一标识}
--	--	--	---

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
-----	----	----

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误
- 107: 有已存在的设备号

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"UpdateAdminAsset\",FTokenID:\"3acef045-d302-4032-
```

```

b40a-d9ee6c1519cd\",FT_AdminAsset:\"){\"FGUID\":\"5baf6ccb-478e-4a13-91a6-6805e
f93b7a7\",\"FVehicleGUID\":\"9b32070e-6cf5-4995-8c60-4d250e6256f8\",\"FAssetID\":\"
645342344\",\"FAssetTypeID\":701,\"FSIMNumber\":\"\",\"FIMEI\":\"\",\"FCustomerID\":\"\",
FDescription\":\"\",\"FAgentGUID\":\"661e7209-c77e-4a54-afb8-6d956f962bdf\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"UpdateAdminAsset\",FTokenID:\"3acef045-d302-4032-
b40a-d9ee6c1519cd\",FT_AdminAsset:\"){\"FGUID\":\"5baf6ccb-478e-4a13-91a6-6805e
f93b7a7\",\"FVehicleGUID\":\"9b32070e-6cf5-4995-8c60-4d250e6256f8\",\"FAssetID\":\"
645342344\",\"FAssetTypeID\":701,\"FSIMNumber\":\"\",\"FIMEI\":\"\",\"FCustomerID\":\"\",
FDescription\":\"\",\"FAgentGUID\":\"661e7209-c77e-4a54-afb8-6d956f962bdf\"}\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get

```

```

request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'UpdateAdminAsset',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FT_AdminAsset': '{"FGUID":"5baf6ccb-478e-4a13-91a6-6805ef93b7a7","FVehicleGUID":"9b32070e-6cf5-4995-8c60-4d250e6256f8","FAssetID":"645342344","FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":"","FAgentGUID":"661e7209-c77e-4a54-afb8-6d956f962bdf"}'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.Loads(page)
return page;

```

# 新增设备

## 简要描述：

- 修改设备

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (AddAdminAssetBatch)
FAgentGUID	是	string	公司唯一标识
			[{"FVehicleName": "车辆牌号", "FAssetID": "设备

FT_AdminAssetList	是	Object	ID,"FAssetTypeID":设备类型,"FSIMNumber":SIM卡号,"FIMEI":IMEI号,"FCustomerID":设备编号,"FDescription":描述,"FAgentGUID":公司唯一标识, ...]
-------------------	---	--------	--

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
-----	----	----

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误
- 106: 有设备号为空
- 107: 有已存在的设备ID
- 108: 有车辆已绑定设备
- 109: 有错误的设备类型

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
String body = "{FAction:\\"UpdateAdminAsset\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAgentGUID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FT_AdminAssetList:\\"[{"FVehicleName":"4678979","FAssetID":"645342344","FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":""},{
"FVehicleName":"7984798","FAssetID":"645342344","FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":""}]"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction:\\"UpdateAdminAsset\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAgentGUID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FT_

```

```

AdminAssetList:\":[{"FVehicleName":"4678979","FAssetID":"645342344","FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":""},{
"FVehicleName":"7984798","FAssetID":"645342344","FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":""}]];
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
//返回的是json字符串
return reader.ReadToEnd();
}

```

## Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
'FAction': 'UpdateAdminAsset',
'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
'FAgentGUID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
'FT_AdminAssetList': '[{"FVehicleName":"3453","FAssetID":"645342344",
"FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":""},{"FVehicleName":"6454","FAssetID":"645342344","FAssetTypeID":701,"FSIMNumber":"","FIMEI":"","FCustomerID":"","FDescription":""}]'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)

```

return page;

# 设备类型

## 简要描述：

- 获取所有设备类型

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminAssetType)
FAgentGUID	是	string	公司唯一标识

## 返回示例：

### 正确时返回：

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FAssetTypeID": 1120,
      "FOfflineTime": 86400,
      "FAssetTypeName": "AnyTrack",
      "FSystemAssetTypeName": "AnyTrack",
      "FClassify": 6,
      "FGateway": "47.112.122.222",
      "FPort": 10120
    },
    {
      "FAssetTypeID": 1150,
      "FOfflineTime": 86400,
      "FAssetTypeName": "AOVX-A/G",
      "FSystemAssetTypeName": "AOVX-A/G",
      "FClassify": 9,
      "FGateway": "47.112.122.222",
      "FPort": 6608
    }
  ]
}

```

#### 错误时返回:

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

#### 返回参数说明:

参数名	类型	说明
FAssetTypeID	Int	设备类型
FOfflineTime	String	设备离线时长（秒）
FAssetTypeName	String	设备类型名称（自定义）
FSystemAssetTypeName	String	设备类型名称

FClassify	Int	设备分类（1：电子锁；2：GP系列；3：部标；4：集装箱监控系列；5：带胎温胎压系列产品；6：其他（可当做基础定位设备）；7：蓝牙锁；8：铅封锁；9：追货宝系列 10：砂浆罐系列）
FGateway	String	网关IP
FPort	Int	网关端口

## 备注:

- 更多返回错误代码如下：
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAdminAssetType\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAgentGUID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
```

```

BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{FAction: \"QueryAdminAssetType\", FTokenID: \"3acef045-d302-40
32-b40a-d9ee6c1519cd\", FAgentGUID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}
";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminAssetType',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAgentGUID': '3acef045-d302-4032-b40a-d9ee6c1519cd'
}
data = parse.urlencode(data).encode('utf-8')

```

```
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 删除设备

## 简要描述：

- 删除设备

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (DeleteAdminAsset)
FGUIDs	是	string	设备唯一标识,多个设备以英文逗号隔开

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

### 返回参数说明:

参数名	类型	说明
-----	----	----

### 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

### 请求示例:

#### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"DeleteAdminAsset\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FGUIDs:\\"5f2d6752-2ca7-4c4e-909f-7030cefbb72a,5f2d6752-2ca7-4c4e-909f-7030cefbb72a\\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
```

```

indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"DeleteAdminAsset\",FTokenID:\"3acef045-d302-4032-
b40a-d9ee6c1519cd\",FGUIDs:\"5f2d6752-2ca7-4c4e-909f-7030cefbb72a,5f2d6752-
2ca7-4c4e-909f-7030cefbb72a\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/>";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

}

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'DeleteAdminAsset',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': '5f2d6752-2ca7-4c4e-909f-7030cefbb72a,5f2d6752-2ca7-4c4e-909f-7030cefbb72a'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 设备详情

## 简要描述：

- 获取单个设备详情信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminAssetByFGUID)
FGUID	是	string	设备唯一标识

## 返回示例:

### 正确时返回:

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "b8ac068d-b9c2-428f-ac52-5df88d069e5e",
      "FAssetID": "2690929719",
      "FAssetTypeID": 200,
      "FDescription": "",
      "FSIMNumber": "89860410102170005799",
      "FIMEI": "89860410102170005799",
      "FAgentGUID": "042032ac-6382-43b6-974f-7ec7f2c0fdfb",
      "FMac": "00E02A360860",
      "FAgentName": "柯力二部",
      "FCreateTime": "2022-05-31T02:08:11.873",
      "FVehicleGUID": "d30bd042-f69a-4f81-ad64-bac303c70980",
      "FVehicleName": "2690929719",
      "FCustomerID": "",
      "FAssetTypeName": "JT200KS"
    }
  ]
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FAssetTypeName	String	设备类型名称
FDescription	String	设备描述

FSIMNumber	String	SIM卡号
FIMEI	String	IMEI
FAgentGUID	String	设备所属公司唯一标识
FAgentName	String	设备所属公司名
FCreateTime	DateTime	注册时间(UTC时间)
FCustomerID	DateTime	设备编号
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车牌号
FMac	String	Mac

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAdminAssetByFGUID\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUID: \"5f2d6752-2ca7-4c4e-909f-7030cefbb72a\"}";

URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
```

```

conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryAdminAssetByFGUID\", FTokenID: \"3acef045-d302
-4032-b40a-d9ee6c1519cd\", FGUID: \"5f2d6752-2ca7-4c4e-909f-7030cefb72a\"}";

Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminAssetByFGUID',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUID': '5f2d6752-2ca7-4c4e-909f-7030cefbb72a'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                 r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 设备信息

## 简要描述：

- 获取设备基础信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminAssetList)

## 返回示例：

### 正确时返回：

```
{
```

```

"Result": 200,
"Message": "check token success",
"FObject": [
  {
    "FGUID": "10f47be0-fe9c-4cc2-ae69-50a14bbfa55c",
    "FAssetID": "7560717002",
    "FAssetTypeID": 701,
    "FDescription": "",
    "FSIMNumber": "343343434",
    "FAgentGUID": "25a48ca5-a844-4732-8af4-afa2a9d5c6dc",
    "FMac": "",
    "FAgentName": "测试",
    "FGroupGUID": null,
    "FGroupName": null,
    "FCreateTime": "2020-05-17T12:10:44.143",
    "FFactorySimNo": null
  },
  {
    "FGUID": "983809d8-a894-43f8-8b52-37bab146fb13",
    "FAssetID": "868120231481844",
    "FAssetTypeID": 1010,
    "FDescription": "",
    "FSIMNumber": "868120231481844",
    "FAgentGUID": "4bfca45f-807d-45f4-bf4c-4ebabbf857ca",
    "FMac": "",
    "FAgentName": "Concox测试",
    "FGroupGUID": null,
    "FGroupName": null,
    "FCreateTime": "2020-07-31T02:30:37.433",
    "FFactorySimNo": null
  }
]
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

--	--	--

参数名	类型	说明
FGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FDescription	String	设备描述
FSIMNumber	String	SIM卡号
FAgentGUID	String	设备所属公司唯一标识
FAgentName	String	设备所属公司名
FGroupGUID	String	设备所属分组唯一标识
FGroupName	String	设备所属分组名
FCreateTime	DateTime	注册时间(UTC时间)
FExpireTime	DateTime	设备到期时间(UTC时间)
FFactorySimNo	String	出厂SIM卡号
FMac	String	Mac

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAdminAssetList\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
```

```

// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryAdminAssetList\", FTokenID: \"3acef045-d302-40
32-b40a-d9ee6c1519cd\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

```

```
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminAssetList',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 单个设备信息

## 简要描述：

- 获取单个设备基础信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2022-06-06	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminAssetByAssetId)
FAssetID	是	string	设备ID /Mac

## 返回示例：

### 正确时返回：

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "b8ac068d-b9c2-428f-ac52-5df88d069e5e",
      "FAssetID": "2690929719",
      "FAssetTypeID": 200,
      "FDescription": "",
      "FSIMNumber": "89860410102170005799",
      "FAgentGUID": "042032ac-6382-43b6-974f-7ec7f2c0fdfb",
      "FMac": "",
      "FAgentName": "柯力二部",
      "FCreateTime": "2022-05-31T02:08:11.873",
      "FExpireTime": null,
      "FVehicleGUID": "d30bd042-f69a-4f81-ad64-bac303c70980",
      "FVehicleName": "2690929719"
    }
  ]
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FDescription	String	设备描述
FSIMNumber	String	SIM卡号
FAgentGUID	String	设备所属公司唯一标识

FAgentName	String	设备所属公司名
FCreateTime	DateTime	注册时间(UTC时间)
FExpireTime	DateTime	设备到期时间(UTC时间)
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车牌号
FMac	String	Mac

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAdminAssetByAssetId\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetID: \"2690929760\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
```

```

pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{FAction:\\"QueryAdminAssetByAssetId\\",FTokenID:\\"3acef045-d3
02-4032-b40a-d9ee6c1519cd\\",FAssetID:\\"2690929760\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminAssetByAssetId',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAssetID': '2690929719'
}

```

```
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 设备或车辆转移

## 简要描述：

- 将设备或者车辆转移到另一个公司

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (TransferAssetByFGUIDs)
FNewAgentGUID	是	string	公司唯一标识
FGUIDs	是	string	设备或者车辆唯一标识，多个以英文逗号隔开
FSelectType	是	int	0: 按终端查 1:按车辆查； ; 默认0按终端

FStatus	是	int	是否转移绑定的车辆/设备（1：转移；0：不转移）
---------	---	-----	--------------------------

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "succeed",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
-----	----	----

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"TransferAssetByFGUIDs\",\"FGUIDs\":\"08c75b42-e80a-41ef-9c38-0bc4fd28cbe1,d2204050-7314-491a-a77f-87bce7ee9593\",\"FNewAgentGUID\"
```

```

:"0d1cada4-430c-4823-ba96-5ab4913d4019", "FStatus":0, "FSelectType":1, "FToken
ID":"ec51d691-d9a7-4f74-95a8-1856be2ae741", "FTimeDifferent":28800, "FLanguag
e":"1"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{\"FAction\":\"TransferAssetByFGUIDs\", \"FGUIDs\":\"08c75b42-e80a-4
1ef-9c38-0bc4fd28cbe1,d2204050-7314-491a-a77f-87bce7ee9593\", \"FNewAgentGUID
\":\"0d1cada4-430c-4823-ba96-5ab4913d4019\", \"FStatus\":0, \"FSelectType\":1, \"FToken
ID\":\"ec51d691-d9a7-4f74-95a8-1856be2ae741\", \"FTimeDifferent\":28800, \"FLanguag
e\":\"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";

```

```

request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'TransferAssetByFGUIDs',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': '08c75b42-e80a-41ef-9c38-0bc4fd28cbe1,d2204050-7314-491a-a77
f-87bce7ee9593'
    'FNewAgentGUID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FStatus': 0,
    'FSelectType': 0
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.Loads(page)
return page;

```

## 车辆管理

[修改车辆](#)

---

[新增车辆](#)

---

[车辆类型](#)

---

[删除车辆](#)

---

[车辆详情](#)

---

[车辆信息](#)

---

# 修改车辆

## 简要描述：

- 修改车辆信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateAdminVehicle)
			{“FGUID”: 车辆唯一标识,“FVehicleName”:车箱号,“FVIN”:车架号,“FEngineNumber”:发动机号,“FVehicleTypeID”:车辆类型,“FInsuDate”:保险日

FT_AdminVehicle	是	Object	期,"FPurchaseDate":购车日期,"FDescription":描述,"FAgentGUID":公司唯一标识,"FVehicleImgType":图片类型,非必填,"FColorRGB":颜色,非必填,"FImage":Base64字符的图片文件,非必填}
-----------------	---	--------	--

## FVehicleImgType 说明



## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
-----	----	----

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误
- 107: 有已存在的车箱号

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"UpdateAdminVehicle\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FT_AdminVehicle:\\"{"FGUID": "5f2d6752-2ca7-4c4e-909f-7030cefbb72a",FVehicleName:\\"43564565343\\",FVIN:\\"",FEngineNumber:\\"",FVehicleTypeID:\\"1\\",FInsuDate:\\"2023-4-6 00:00:00\\",FPurchaseDate:\\"2023-4-6 00:00:00\\",FDescription:\\"4435\\",FAgentGUID:\\"661e7209-c77e-4a54-afb8-6d956f962bdf\\"}"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C#:

```
//请求路径
```

```

string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"UpdateAdminVehicle\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FT_AdminVehicle:\\"{"FGUID": \"5f2d6752-2ca7-4c4e-909f-7030cefbb72a\", \"FVehicleName\": \"43564565343\", \"FVIN\": \"\", \"FEngineNumber\": \"\", \"FVehicleTypeID\": 1, \"FInsuDate\": \"2023-4-6 00:00:00\", \"FPurchaseDate\": \"2023-4-6 00:00:00\", \"FDescription\": \"4435\", \"FAgentGUID\": \"661e7209-c77e-4a54-afb8-6d956f962bdf\"}\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'UpdateAdminVehicle',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FT_AdminVehicle': '{"FGUID": "5f2d6752-2ca7-4c4e-909f-7030cefbb72a",
"FVehicleName": "43564565343", "FVIN": "", "FEngineNumber": "", "FVehicleTypeID":
1, "FInsuDate": "2023-4-6 00:00:00", "FPurchaseDate": "2023-4-6 00:00:00", "FDes
cription": "4435", "FAgentGUID": "661e7209-c77e-4a54-afb8-6d956f962bdf"}'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)

```

```
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 新增车辆

## 简要描述：

- 新增车辆

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (AddAdminVehicleBatch)
FAgentGUID	是	string	公司唯一标识
			[{"FVehicleName":车箱号,"FVIN":车架

FT_AdminVehicleList	是	Object	号,"FEngineNumber":发动机号,"FVehicleTypeID":车辆类型,"FInsuDate":保险日期,"FPurchaseDate":购车日期,"FDescription":描述,"FAssetID":设备号,"FAssetTypeID":设备类型ID,"FVehicleImgType":图片类型,非必填,"FColorRGB":颜色,非必填,"FImage":Base64字符的图片文件,非必填}}
---------------------	---	--------	--

## FVehicleImgType 说明



## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
{
  "Result": 107,
  "Message": "4564844 , 654564",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
-----	----	----

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误
- 106: 有为空的车箱号
- 107: 有已存在的车箱号

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"AddAdminVehicleBatch\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAgentGUID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FT_AdminVehicleList:\\"[{\\"FVehicleName\\":\"43564565343\\",\"FVIN\\":\"\\",\"FEngineNumber\\":\"\\",\"FVehicleTypeID\\":1,\"FInsuDate\\":\"2023-4-6 00:00:00\\",\"FPurchaseDate\\":\"2023-4-6 00:00:00\\",\"FDescription\\":\"4435\"}]\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
```

```

        result += line;
    }
    //返回的是json字符串
    return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction:\\"AddAdminVehicleBatch\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAgentGUID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FT_AdminVehicleList:\\"[{\\"FVehicleName\\":\\"43564565343\\",\\"FVIN\\":\\"\\",\\"FEngineNumber\\":\\"\\",\\"FVehicleTypeID\\":1,\\"FInsuDate\\":\\"2023-4-6 00:00:00\\",\\"FPurchaseDate\\":\\"2023-4-6 00:00:00\\",\\"FDescription\\":\\"4435\\"}]}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'AddAdminVehicleBatch',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAgentGUID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FT_AdminVehicleList': '[{"FVehicleName":"43564565343","FVIN":"","FEngineNumber":"","FVehicleTypeID":1,"FInsuDate":"2023-4-6 00:00:00","FPurchaseDate":"2023-4-6 00:00:00","FDescription":"4435"}]'
}
data = parse.urlencode(data).encode('utf-8')

```

```
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 车辆类型

## 简要描述：

- 获取所有车辆类型

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminVehicleType)
FGUID	是	string	车辆唯一标识

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FVehicleTypeID": 1,
      "FVehicleTypeName": "货车",
      "FEnTypeName": "Truck"
    },
    {
      "FVehicleTypeID": 2,
      "FVehicleTypeName": "小型客车",
      "FEnTypeName": "Minibus"
    },
    {
      "FVehicleTypeID": 3,
      "FVehicleTypeName": "出租车",
      "FEnTypeName": "Taxi"
    },
    {
      "FVehicleTypeID": 4,
      "FVehicleTypeName": "大型客车",
      "FEnTypeName": "Bus"
    },
    {
      "FVehicleTypeID": 5,
      "FVehicleTypeName": "集装箱",
      "FEnTypeName": "Container"
    },
    {
      "FVehicleTypeID": 6,
      "FVehicleTypeName": "冷箱车",
      "FEnTypeName": "Truck-R"
    },
    {
      "FVehicleTypeID": 7,
      "FVehicleTypeName": "厢式货车",
      "FEnTypeName": "VAN"
    },
    {
      "FVehicleTypeID": 8,
      "FVehicleTypeName": "拖车",
      "FEnTypeName": "Trailer"
    },
    {
      "FVehicleTypeID": 9,
```

```

        "FVehicleTypeName": "小汽车",
        "FEnTypeName": "Car"
    },
    {
        "FVehicleTypeID": 10,
        "FVehicleTypeName": "摩托车",
        "FEnTypeName": "Motorcycle"
    },
    {
        "FVehicleTypeID": 11,
        "FVehicleTypeName": "冷库",
        "FEnTypeName": "Cold Store"
    }
]
}

```

### 错误时返回:

```

{
    "Result": 102,
    "Message": "Action is error",
    "FObject": []
}

```

### 返回参数说明:

参数名	类型	说明
FEnTypeName	String	车辆类型名称（英文）
FVehicleTypeName	String	车辆类型名称（中文）
FVehicleTypeID	Int	车辆类型

### 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

### 请求示例:

#### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAdminVehicleType\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryAdminVehicleType\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";

```

```
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminVehicleType',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 删除车辆

## 简要描述：

- 删除车辆

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (DeleteAdminVehicle)
FGUIDs	是	string	车辆唯一标识,多个车辆以英文逗号隔开

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"DeleteAdminVehicle\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FGUIDS:\\"5f2d6752-2ca7-4c4e-909f-7030cefbb72a,5f2d6752-2ca7-4c4e-909f-7030cefbb72a\\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
```

```

indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"DeleteAdminVehicle\\",FTokenID:\\"3acef045-d302-403
2-b40a-d9ee6c1519cd\\",FGUIDs:\\"5f2d6752-2ca7-4c4e-909f-7030cefbb72a,5f2d675
2-2ca7-4c4e-909f-7030cefbb72a\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

}

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'DeleteAdminVehicle',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': '5f2d6752-2ca7-4c4e-909f-7030cefbb72a,5f2d6752-2ca7-4c4e-909f-7030cefbb72a'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 车辆信息

## 简要描述：

- 获取车辆基础信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminVehicleList)

## 返回示例：

### 正确时返回：

```
{
```

```
"Result": 200,
"Message": "check token success",
"FObject": [
  {
    "FGUID": "4b0be13f-4703-4629-93d2-b51f632344dc",
    "FVehicleName": "视频机816",
    "FAssetGUID": "066cd9b2-b64b-44f5-9f19-b6e7dc974140",
    "FAssetID": "6165016816",
    "FAssetTypeID": 808,
    "FVehicleCode": "816",
    "FVehicleTypeID": 1,
    "FDescription": "",
    "FOperateType": "",
    "FVIN": "",
    "FAgentGUID": "903ba776-b329-474e-86b1-3a922c54f821",
    "FAgentName": "部标测试",
    "FGroupGUID": "0b887ae3-e190-42a2-81f3-81f6127c8aa1",
    "FGroupName": "视频机测试",
    "FMainDriverGUID": "778572c1-7ca4-44a9-ab2d-6c4f9cfed800",
    "FDriverName": "Jhon",
    "FDriverPhoneNumber": "18362985485",
    "FCreateTime": "2020-07-20T06:13:37.613",
    "FColorRGB": null,
    "FVehicleImgType": 0,
    "FImage": null
  },
  {
    "FGUID": "9154a616-038e-40da-91b4-87b7faccc15f",
    "FVehicleName": "粤B45678",
    "FAssetGUID": null,
    "FAssetID": null,
    "FAssetTypeID": null,
    "FVehicleCode": "",
    "FVehicleTypeID": 1,
    "FDescription": "粤B45678",
    "FOperateType": "1",
    "FVIN": "",
    "FAgentGUID": "4b3dd67a-00af-4118-87bf-fa9583a83b25",
    "FAgentName": "Joint测试",
    "FGroupGUID": "b595f14b-e8a7-49b3-b26c-4687ebfcb33f",
    "FGroupName": "JT702/JT705",
    "FMainDriverGUID": "00000000-0000-0000-0000-000000000000",
    "FDriverName": null,
    "FDriverPhoneNumber": null,
    "FCreateTime": "2020-08-04T09:27:21.127",
    "FColorRGB": null,
  }
]
```

```

    "FVehicleImgType": 0,
    "FImage": null
  }
}
]
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FGUID	String	车辆唯一标识
FVehicleName	String	车辆名
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FVehicleCode	String	车辆编码
FVehicleTypeID	Int	车辆类型
FDescription	String	描述
FOperateType	String	运营类型（1：自营；2：租赁）
FVIN	String	车架号
FAgentGUID	String	设备所属公司唯一标识
FAgentName	String	设备所属公司名
FGroupGUID	String	设备所属分组唯一标识
FGroupName	String	设备所属分组名

FMainDriverGUID	String	主驾司机唯一标识
FDriverName	String	主驾司机姓名
FDriverPhoneNumber	String	主驾司机电话
FCreateTime	DateTime	注册时间(UTC时间)
FColorRGB	String	颜色
FVehicleImgType	Int	图标图片类型
FImage	String	车辆图片url

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAdminVehicleList\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
```

```

pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryAdminVehicleList\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminVehicleList',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd'
}

```

```
    }  
    data = parse.urlencode(data).encode('utf-8')  
    headers = {  
        'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
            r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
        'Connection': 'keep-alive'  
    }  
    req = request.Request(url, headers=headers, data=data)  
    page = request.urlopen(req).read()  
    page = page.decode('utf-8')  
    # json_array = json.loads(page)  
    return page;
```

# 车辆详情

## 简要描述：

- 获取单个车辆详情信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2023-04-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminVehicleByFGUID)
FGUID	是	string	车辆唯一标识

## 返回示例：

### 正确时返回：

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "b8ac068d-b9c2-428f-ac52-5df88d069e5e",
      "FAssetID": "2690929719",
      "FAssetTypeID": 200,
      "FDescription": "",
      "FVehicleCode": "89860410102170005799",
      "FVIN": "89860410102170005799",
      "FAgentGUID": "042032ac-6382-43b6-974f-7ec7f2c0fdfb",
      "FAgentName": "柯力二部",
      "FCreateTime": "2022-05-31T02:08:11.873",
      "FVehicleGUID": "d30bd042-f69a-4f81-ad64-bac303c70980",
      "FVehicleName": "2690929719",
      "FVehicleTypeID": "1",
      "FEngineNumber": "64688",
      "FInsuDate": "2023-04-07",
      "FPurchaseDate": "2023-04-07",
      "FColorRGB": null,
      "FVehicleImgType": 0,
      "FImage": null
    }
  ]
}

```

#### 错误时返回:

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

#### 返回参数说明:

参数名	类型	说明
FGUID	String	车辆唯一标识
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID

FAssetTypeID	Int	设备类型
FVehicleCode	String	车辆编号
FDescription	String	描述
FSIMNumber	String	SIM卡号
FVIN	String	车架号
FEngineNumber	String	发动机号
FAgentGUID	String	公司唯一标识
FAgentName	String	所属公司名
FCreateTime	DateTime	注册时间(UTC时间)
FInsuDate	String	保险日期
FPurchaseDate	String	购车日期
FVehicleTypeID	Int	车牌类型
FColorRGB	String	颜色
FVehicleImgType	Int	图标图片类型
FImage	String	车辆图片url

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryAdminVehicleByFGUID\",FTokenID:\"3acef045-d3
```

```

02-4032-b40a-d9ee6c1519cd\",FGUID:\"5f2d6752-2ca7-4c4e-909f-7030cefbb72a\"}
";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryAdminVehicleByFGUID\",FTokenID:\"3acef045-d3
02-4032-b40a-d9ee6c1519cd\",FGUID:\"5f2d6752-2ca7-4c4e-909f-7030cefbb72a\"}
";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";

```

```
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminVehicleByFGUID',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUID': '5f2d6752-2ca7-4c4e-909f-7030cefbb72a'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 单个车辆信息

## 简要描述：

- 获取单个车辆基础信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2022-06-06	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminVehicleByVehicleName)
FVehicleName	是	string	车牌号

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "5b24bab0-829d-452b-898f-cc492bc9287b",
      "FVehicleName": "2690929760",
      "FAssetGUID": "0a1ecd86-1b0d-4b51-af37-e41dbf71d46a",
      "FAssetID": "2690929760",
      "FAssetTypeID": 200,
      "FVehicleCode": "",
      "FVehicleTypeID": 1,
      "FDescription": "",
      "FOperateType": "1",
      "FVIN": "",
      "FAgentGUID": "042032ac-6382-43b6-974f-7ec7f2c0fdfb",
      "FAgentName": "柯力二部",
      "FGroupGUID": "d1026011-6479-4b74-8a98-aaa42ce2cedc",
      "FGroupName": "砂浆罐",
      "FMainDriverGUID": "00000000-0000-0000-0000-000000000000",
      "FDriverName": null,
      "FDriverPhoneNumber": null,
      "FCreateTime": "2022-06-01T14:12:24.523"
    }
  ]
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
FGUID	String	车辆唯一标识
FVehicleName	String	车辆名
FAssetGUID	String	设备唯一标识

FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FVehicleCode	String	车辆编码
FVehicleTypeID	Int	车辆类型
FDescription	String	描述
FOperateType	String	运营类型（1：自营；2：租赁）
FVIN	String	车架号
FAgentGUID	String	设备所属公司唯一标识
FAgentName	String	设备所属公司名
FGroupGUID	String	设备所属分组唯一标识
FGroupName	String	设备所属分组名
FMainDriverGUID	String	主驾司机唯一标识
FDriverName	String	主驾司机姓名
FDriverPhoneNumber	String	主驾司机电话
FCreateTime	DateTime	注册时间(UTC时间)

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryAdminVehicleByVehicleName\",FTokenID:\"3acef
```

```

045-d302-4032-b40a-d9ee6c1519cd\",FVehicleName:\"2690929760\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryAdminVehicleByVehicleName\",FTokenID:\"3acef
045-d302-4032-b40a-d9ee6c1519cd\",FVehicleName:\"2690929760\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;

```

```
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminVehicleByVehicleName',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FVehicleName': '2690929760'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 实时数据

## 简要描述：

- 获取设备的实时定位数据

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/LBS>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/LBS>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryLBSMonitorListByFGUIDs)
FGUIDs	是	string	车辆/设备唯一标识，多个以逗号隔开
FType	是	int	1: 车辆 2: 设备

## 返回示例:

## 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FAssetID": "7500313618",
      "FAssetTypeID": 701,
      "FLongitude": 113.916965,
      "FLatitude": 22.58107,
      "FSpeed": 0,
      "FDirection": 0,
      "FMileage": 0,
      "FGPSTime": "2020-08-12T01:56:16Z",
      "FRecvTime": "2020-08-12T01:56:31.036Z",
      "FLocationType": 1,
      "FCellSignal": 8,
      "FGPSSignal": 5,
      "FMNC": 0,
      "FMCC": 460,
      "FLAC": 10342,
      "FCELLID": 4242,
      "FBattery": 100,
      "FACC": -1,
      "FLockRope": 0,
      "FLockStatus": 0,
      "FFuelCut": -1,
      "FDoor": -1,
      "FMotor": -1,
      "FFuelValue1": -1,
      "FFuelValue2": -1,
      "FFuelValue3": -1,
      "FTemperature1": -1000.0,
      "FTemperature2": -1000.0,
      "FTemperature3": -1000.0,
      "FTemperature4": -1000.0,
      "FTemperature5": -1000.0,
      "FTemperature6": -1000.0,
      "FHumidity1": 0,
      "FHumidity2": 0,
      "FHumidity3": 0,
      "FHumidity4": 0,
      "FHumidity5": 0,
      "FHumidity6": 0,
      "FExpandProto": null,
    }
  ]
}
```

```
“FOnline” : 1 ,
"SubAssets" : {
  "E0171A0107" : {
    "FAssetID" : "E0171A0107",
    "FDateTime" : "2020-08-12T01:53:10Z",
    "FEvent" : 6,
    "FHumidity" : 0,
    "FLockRope" : 0,
    "FLockStatus" : 0,
    "FLockTimes" : 1028,
    "FPower" : 68,
    "FRSSI" : 84,
    "FSensorType" : 4,
    "FTemperature" : -1000.0,
    "FVoltage" : 3.93
  },
  "1018330094" : {
    "FAssetID" : "1018330094",
    "FDateTime" : "2020-08-12T01:34:30Z",
    "FEvent" : -1,
    "FHumidity" : 71,
    "FLockRope" : 1,
    "FLockStatus" : 1,
    "FLockTimes" : -1,
    "FPower" : 98,
    "FRSSI" : 69,
    "FSensorType" : 1,
    "FTemperature" : 27.6,
    "FVoltage" : 3.04
  },
  "E0171A00FD" : {
    "FAssetID" : "E0171A00FD",
    "FDateTime" : "2020-08-11T11:07:35Z",
    "FEvent" : 5,
    "FHumidity" : 0,
    "FLockRope" : 0,
    "FLockStatus" : 0,
    "FLockTimes" : 17,
    "FPower" : 68,
    "FRSSI" : 53,
    "FSensorType" : 4,
    "FTemperature" : -1000.0,
    "FVoltage" : 3.93
  },
  "1018100169" : {
    "FAssetID" : "1018100169",
```

```

        "FDateTime": "2020-08-12T01:29:29Z",
        "FEvent": -1,
        "FHumidity": 71,
        "FLockRope": 1,
        "FLockStatus": 1,
        "FLockTimes": -1,
        "FPower": 98,
        "FRSSI": 87,
        "FSensorType": 1,
        "FTemperature": 28.200000000000003,
        "FVoltage": 3.04
    }
}
]
}

```

#### 错误时返回:

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

#### 返回参数说明:

参数名	类型	说明
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FLongitude	Decimal	经度
FLatitude	Decimal	纬度
FSpeed	Int	速度
FDirection	Int	方向
FMileage	Int	里程
FGPSTime	DateTime	GPS时间（UTC时间，转成北京时间需要+8小时）

FRecvTime	DateTime	数据接收时间（UTC时间，转成北京时间需要+8小时）
FLocationType	Int	定位类型 1:卫星定位 2:基站定位 3:小区码定位 4:手机定位 0:不定位
FCellSignal	Int	GSM值号值
FGPSSignal	Int	GPS值号值
FMNC	Int	小区码MNC
FMCC	Int	小区码MCC
FLAC	Int	小区码LAC
FCELLID	Int	小区码CID
FBattery	Int	电量
FACC	Int	引擎状态
FLockRope	Int	锁绳状态 (1: 拔出 0:插入 -1:无)
FLockStatus	Int	锁状态 (1: 开锁 0:关锁 -1:无)
FFuelCut	Int	断油电状态
FDoor	Int	车门状态
FMotor	Int	车辆发动机状态
FFuelValue1	Int	油位值1
FFuelValue2	Int	油位值2
FFuelValue3	Int	油位值3
FTemperature1	Decimal	温度值1
FTemperature2	Decimal	温度值2
FTemperature3	Decimal	温度值3
FTemperature4	Decimal	温度值4
FTemperature5	Decimal	温度值5
FTemperature6	Decimal	温度值6

FHumidity1	Decimal	湿度值1
FHumidity2	Decimal	湿度值2
FHumidity3	Decimal	湿度值3
FHumidity4	Decimal	湿度值4
FHumidity5	Decimal	湿度值5
FHumidity6	Decimal	湿度值6
FExpandProto	String	预留字段
SubAssets	String	从机信息
FOnline	Int	是否在网状态 1: 是 0: 否

### FExpandProto: 预留字段

参数名	类型	说明
FAwake	Int	唤醒源-1: 无 0:重启 1: RTC 2: 震动 3: 开盖/拆卸 4: 锁绳/条 5: 充电 6: 刷卡 7: Lora 8: VIP号码 9: 短信 10: 蓝牙 11: 插绳 12: 关盖 13: 模拟 14: 按键 15: 非VIP号码 16: VIP号码唤醒 17: 复位 18: 温湿度异常 19: 气压异常 20: 开机 21: 开关门 22: 光感
FPosture	Int	姿态系数
FDesc	String	其它拓展信息Json字符串

### FDesc:其它拓展信息Json字符串

例:

```
{ "fBackBattery": "92,4.08,0", "fChange": 0, "fNetworkType": 4, "fPosture": "x:-178;y:3;z:12169", "fRopeInfo": "{ \"batteryStatus\": 0, \"exist\": 0, \"mac\": \"000000000000\", \"ropeStatus\": 6, \"rssi\": 0 }", "fVoltage": 0.0 }
```

以下是拓展信息的所有字段，实际返回的Json字符串字段数量不会有这么多

参数名	类型	说明
fBackBattery	String	备用电池
fNetworkType	Int	网络类型 2: 2G 3:3G 4:4G 5:5G
fPosture	String	姿态 "x:1;y:-29;z:-2903"

fRopeInfo	String	锁条状态信息（Json字符串）
fVoltage	Float	电压值
fAngle	Int	倾斜度
fAcceleration	String	加速度
fLx	Float	光照度
fPressure	Float	气压
fVibrate	Int	震幅
fReportType	Int	数据类型 0：实时；1:补报；2：报警
fDoor1	Int	1:门1打开 0：门1关闭
fDoor2	Int	1:门2打开 0：门2关闭
fVolStatus	Int	电压状态（0：LED电压正常 1：LED高电压状态标记）
fVolValue	Double	LED电压值
fEleStatus	Int	电流状态 0：LED电流正常 1：LED高电流状态标记
fEleValue	Double	LED电流值
fLuminance	Int	亮度 0：表示关闭（节能一级），<=25 (节能二级) <=50 (节能三级) <=75 (节能四级) <=100 (节能五级)
fBackCover	Int	后盖状态：-1：无此状态；0：后盖关；1：后盖开
fMainBattery	Int	针对JT705C，主电池状态（1: 主电池拆卸,0:主电池已安装；-1：不显示此状态）

### fRopeInfo:其它拓展信息Json字符串

例：“fRopeInfo”:“{“batteryStatus”:0,“exist”:0,“mac”:“000000000000”,“ropeStatus”:6,“rssi”:0}”

参数名	类型	说明
batteryStatus	Int	低电状态 0：正常 1：低电 -1：-
exist	Int	在位状态 0：不在位 1：在位 -1：-
mac	String	锁条MAC

ropeStatus	Int	0:未知 1: 在线 2: 弯曲 3: 锯断 4: 丢失 5: 锁钩锯断 6: 锁杆锯断 7: 锁杆拔出 -1: -
rsssi	Int	RSSI

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/LBS";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryLBSMonitorListByFGUIDs\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUIDs: \"BE2A1E27-B36E-4301-B2C8-465A62AD215F, FE0BDE75-4A30-4B9C-AD2E-66D8A4F44F5E\", FType: \"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
```

```

while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/LBS";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"QueryLBSMonitorListByFGUIDs\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FGUIDs:\\"BE2A1E27-B36E-4301-B2C8-465A62AD215F,FE0BDE75-4A30-4B9C-AD2E-66D8A4F44F5E\\",FType:\\"1\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/LBS'
data = {
    'FAction': 'QueryLBSMonitorListByFGUIDs',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': 'BE2A1E27-B36E-4301-B2C8-465A62AD215F,FE0BDE75-4A30-4B9C-AD2E-66D8A4F44F5E',
    'FType':1,
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.

```

```
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
        'Connection': 'keep-alive'  
    }  
    req = request.Request(url, headers=headers, data=data)  
    page = request.urlopen(req).read()  
    page = page.decode('utf-8')  
    # json_array = json.loads(page)  
    return page;
```

# 轨迹数据

## 简要描述：

- 获取设备的实时定位数据

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/LBS>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/LBS>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryLBSTrackListByFGUID)
FGUID	是	string	车辆/设备唯一标识
FType	是	int	1: 车辆唯一标识 2:设备唯一标识
FAssetTypeID	是	int	设备类型ID
FStartTime	是	datetime	开始时间(UTC时间)

FEndTime	是	datetime	结束时间(UTC时间)
FLanguage	否	int	1: 中文 0: 英文 默认1 (主要是用于小区码解析)
FDateType	是	int	1: 按接收时间查; 0: 按GPS时间查

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "Lon": 114.062655140347,
      "Lat": 22.5711617062429,
      "Speed": 0,
      "Dir": 0,
      "Mil": 38,
      "LType": 3,
      "MCC": 460,
      "MNC": 0,
      "LAC": 9351,
      "CID": 9493077,
      "GT": "2020-12-27T16:13:47",
      "RT": "2020-12-27T16:14:06",
      "Bat": 45,
      "LS": 1,
      "Hum1": 63,
      "Hum2": 0,
      "Temp1": 27.799999237060547,
      "Temp2": -1000.0
    },
    {
      "Lon": 114.062655140347,
      "Lat": 22.5711617062429,
      "Speed": 0,
      "Dir": 0,
      "Mil": 38,
      "LType": 3,
      "MCC": 460,
      "MNC": 0,
      "LAC": 9351,
      "CID": 9493077,
```

```

"GT": "2020-12-27T16:43:48",
"RT": "2020-12-27T16:44:07",
"Bat": 45,
"LS": 1,
"Hum1": 63,
"Hum2": 0,
"Temp1": 27.799999237060547,
"Temp2": -1000.0
},
{
"Lon": 114.062655140347,
"Lat": 22.5711617062429,
"Speed": 0,
"Dir": 0,
"Mil": 38,
"LType": 3,
"MCC": 460,
"MNC": 0,
"LAC": 9351,
"CID": 9493077,
"GT": "2020-12-27T17:13:49",
"RT": "2020-12-27T17:14:08",
"Bat": 45,
"LS": 1,
"Hum1": 63,
"Hum2": 0,
"Temp1": 27.799999237060547,
"Temp2": -1000.0
},
{
"Lon": 114.06537285643,
"Lat": 22.57270175043,
"Speed": 0,
"Dir": 0,
"Mil": 38,
"LType": 3,
"MCC": 460,
"MNC": 0,
"LAC": 9351,
"CID": 176023831,
"GT": "2020-12-27T17:43:50",
"RT": "2020-12-27T17:44:09",
"Bat": 45,
"LS": 1,
"Hum1": 63,
"Hum2": 0,

```

```

"Temp1": 27.799999237060547,
"Temp2": -1000.0
},
{
  "Lon": 114.062655140347,
  "Lat": 22.5711617062429,
  "Speed": 0,
  "Dir": 0,
  "Mil": 38,
  "LType": 3,
  "MCC": 460,
  "MNC": 0,
  "LAC": 9351,
  "CID": 9493077,
  "GT": "2020-12-27T18:13:51",
  "RT": "2020-12-27T18:14:10",
  "Bat": 44,
  "LS": 1,
  "Hum1": 63,
  "Hum2": 0,
  "Temp1": 27.799999237060547,
  "Temp2": -1000.0
},
{
  "Lon": 114.064252,
  "Lat": 22.572598,
  "Speed": 0,
  "Dir": 0,
  "Mil": 38,
  "LType": 1,
  "MCC": 460,
  "MNC": 0,
  "LAC": 9351,
  "CID": 9493077,
  "GT": "2020-12-27T18:43:26",
  "RT": "2020-12-27T18:43:45",
  "Bat": 44,
  "LS": 1,
  "Hum1": 63,
  "Hum2": 0,
  "Temp1": 27.799999237060547,
  "Temp2": -1000.0
},
{
  "Lon": 114.063952788266,
  "Lat": 22.5727879731391,

```

```

"Speed": 0,
"Dir": 0,
"Mil": 38,
"LType": 3,
"MCC": 460,
"MNC": 0,
"LAC": 9351,
"CID": 176023834,
"GT": "2020-12-27T19:13:53",
"RT": "2020-12-27T19:14:13",
"Bat": 44,
"LS": 1,
"Hum1": 63,
"Hum2": 0,
"Temp1": 27.799999237060547,
"Temp2": -1000.0
},
{
"Lon": 114.062655140347,
"Lat": 22.5711617062429,
"Speed": 0,
"Dir": 0,
"Mil": 38,
"LType": 3,
"MCC": 460,
"MNC": 0,
"LAC": 9351,
"CID": 9493077,
"GT": "2020-12-27T19:43:54",
"RT": "2020-12-27T19:44:14",
"Bat": 44,
"LS": 1
},
{
"Lon": 114.062655140347,
"Lat": 22.5711617062429,
"Speed": 0,
"Dir": 0,
"Mil": 38,
"LType": 3,
"MCC": 460,
"MNC": 0,
"LAC": 9351,
"CID": 9493077,
"GT": "2020-12-27T20:13:55",
"RT": "2020-12-27T20:14:16",

```

```

        "Bat": 44,
        "LS": 1,
        "Hum1": 63,
        "Hum2": 0,
        "Temp1": 27.799999237060547,
        "Temp2": -1000.0
    },
    {
        "Lon": 114.063952788266,
        "Lat": 22.5727879731391,
        "Speed": 0,
        "Dir": 0,
        "Mil": 38,
        "LType": 3,
        "MCC": 460,
        "MNC": 0,
        "LAC": 9351,
        "CID": 176023834,
        "GT": "2020-12-27T20:43:56",
        "RT": "2020-12-27T20:44:16",
        "Bat": 44,
        "LS": 1,
        "Hum1": 63,
        "Hum2": 0,
        "Temp1": 27.799999237060547,
        "Temp2": -1000.0
    },
    {
        "Lon": 114.062655140347,
        "Lat": 22.5711617062429,
        "Speed": 0,
        "Dir": 0,
        "Mil": 38,
        "LType": 3,
        "MCC": 460,
        "MNC": 0,
        "LAC": 9351,
        "CID": 9493077,
        "GT": "2020-12-27T21:13:57",
        "RT": "2020-12-27T21:14:17",
        "Bat": 44,
        "LS": 1,
        "Hum1": 63,
        "Hum2": 0,
        "Temp1": 27.799999237060547,
        "Temp2": -1000.0
    }

```

```

    },
    {
      "Lon": 114.062655140347,
      "Lat": 22.5711617062429,
      "Speed": 0,
      "Dir": 0,
      "Mil": 38,
      "LType": 3,
      "MCC": 460,
      "MNC": 0,
      "LAC": 9351,
      "CID": 9493077,
      "GT": "2020-12-27T21:43:58",
      "RT": "2020-12-27T21:44:19",
      "Bat": 44,
      "LS": 1
    },
    {
      "Lon": 114.062655140347,
      "Lat": 22.5711617062429,
      "Speed": 0,
      "Dir": 0,
      "Mil": 38,
      "LType": 3,
      "MCC": 460,
      "MNC": 0,
      "LAC": 9351,
      "CID": 9493077,
      "GT": "2020-12-27T22:13:59",
      "RT": "2020-12-27T22:14:20",
      "Bat": 44,
      "LS": 1,
      "Hum1": 63,
      "Hum2": 0,
      "Temp1": 27.799999237060547,
      "Temp2": -1000.0
    },
    {
      "Lon": 114.06580855643,
      "Lat": 22.57472835043,
      "Speed": 0,
      "Dir": 0,
      "Mil": 38,
      "LType": 3,
      "MCC": 460,
      "MNC": 0,
  
```

```

"LAC": 9351,
"CID": 176023821,
"GT": "2020-12-27T22:44:00",
"RT": "2020-12-27T22:44:21",
"Bat": 44,
"LS": 1,
"Hum1": 63,
"Hum2": 0,
"Temp1": 27.799999237060547,
"Temp2": -1000.0
},
{
  "Lon": 114.062655140347,
  "Lat": 22.5711617062429,
  "Speed": 0,
  "Dir": 0,
  "Mil": 38,
  "LType": 3,
  "MCC": 460,
  "MNC": 0,
  "LAC": 9351,
  "CID": 9493077,
  "GT": "2020-12-27T23:14:01",
  "RT": "2020-12-27T23:14:22",
  "Bat": 44,
  "LS": 1,
  "Hum1": 63,
  "Hum2": 0,
  "Temp1": 27.799999237060547,
  "Temp2": -1000.0
},
{
  "Lon": 0.000000,
  "Lat": 0.000000,
  "Speed": 0,
  "Dir": 0,
  "Mil": 38,
  "LType": 0,
  "MCC": 0,
  "MNC": 0,
  "LAC": 0,
  "CID": 0,
  "GT": "2020-12-27T23:25:50",
  "RT": "2020-12-27T23:27:00",
  "Bat": 44,
  "LS": 1,

```

```

        "Hum1": 63,
        "Hum2": 0,
        "Temp1": 27.799999237060547,
        "Temp2": -1000.0
    },
    {
        "Lon": 114.062655140347,
        "Lat": 22.5711617062429,
        "Speed": 0,
        "Dir": 0,
        "Mil": 38,
        "LType": 3,
        "MCC": 460,
        "MNC": 0,
        "LAC": 9351,
        "CID": 9493077,
        "GT": "2020-12-27T23:26:50",
        "RT": "2020-12-27T23:27:13",
        "Bat": 44,
        "LS": 1,
        "Hum1": 63,
        "Hum2": 0,
        "Temp1": 27.799999237060547,
        "Temp2": -1000.0
    }
]
}

```

### 错误时返回:

```

{
    "Result": 102,
    "Message": "Action is error",
    "FObject": []
}

```

### 返回参数说明:

参数名	类型	说明
Lon	Decimal	经度
Lat	Decimal	纬度
Speed	Int	速度

Dir	Int	方向
Mil	Int	里程
MCC	Int	移动信号国家码
MNC	Int	运营商代码
LAC	Int	位置区码
CID	Int	全球小区识别码
GT	DateTime	GPS时间(UTC时间)
RT	DateTime	数据接收时间(UTC时间)
LType	Int	定位类型 1:卫星定位 2:基站定位 3:小区码定位(同2,实际也称为基站定位) 4:手机定位 0:不定位
Bat	Int	电量
ACC	Int	引擎状态 (1: 开; 0:关)
LS	Int	锁状态 (1: 开; 0: 关)
LR	Int	锁绳状态 (1: 拔出; 0: 插入)
CS	Int	GSM信号值
GS	Int	GPS信号值
Awaken	Int	唤醒源 (0: 重启; 1: RTC; 2: 震动; 3: 开盖/拆卸; 4: 剪绳; 5: 充电; 6: 刷卡; 7: Loar;8:VIP号码; 9: 短信; 10: 蓝牙; 11: 插绳; 12: 关盖; 13: 模拟; 14: 按键; 15: 非VIP号码; 16: VIP号码; 17: 复位)
FV1	Int	油位值1
FV2	Int	油位值2
FV3	Int	油位值3
Hum1	Int	湿度值1
Hum2	Int	湿度值2
Temp1	Decimal	温度值1 (-1000表示无效数据)

Temp2	Decimal	温度值2（-1000表示无效数据）
-------	---------	-------------------

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/LBS";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryLBSTrackListByFGUID\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FGUID:\"BE2A1E27-B36E-4301-B2C8-465A62AD215F\",FType:\"1\",FStartTime:\"2020-08-05 16:00:00\",FEndTime:\"2020-08-12 15:59:59\",FAssetTypeID:\"701\",FLanguage:\"1\",FDateType:\"0\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
```

```

}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/LBS";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction:\\"QueryLBSTrackListByFGUID\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FGUID:\\"BE2A1E27-B36E-4301-B2C8-465A62AD215F\\",FType:\\"1\\",FStartTime:\\"2020-08-05 16:00:00\\",FEndTime:\\"2020-08-12 15:59:59\\",FAssetTypeID:\\"701\\",FLanguage:\\"1\\",FDateType:\\"0\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
//返回的是json字符串
return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/LBS'
data = {
'FAction': 'QueryLBSTrackListByFGUID',
'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
'FGUID': 'BE2A1E27-B36E-4301-B2C8-465A62AD215F',
'FType': 1,
'FStartTime': '2020-08-05 16:00:00',
'FEndTime': '2020-08-12 15:59:59',
'FAssetTypeID': '701',
'FLanguage': '1',
'FDateType': 0
}

```

```
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

## 远程开锁

### 开锁指令

---

### 查询锁状态

---

# 开锁指令

## 简要描述：

- 发送远程开锁指令

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (OpenLockControl)
FAssetGUID	是	string	设备唯一标识

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 113: 未在开锁区域, 禁止开锁
- 115: 未定位, 无法进行开锁
- 110: 没有开锁权限
- 108: 开锁指令发送异常
- 106: 设备不属于该用户
- 105: 系统异常
- 123: 设备不支持
- 111: 设备离线或休眠

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"OpenLockControl\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
```

```

conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{FAction:\\"OpenLockControl\\",FTokenID:\\"3acef045-d302-4032-b
40a-d9ee6c1519cd\\",FAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))

```

开锁指令

```
{  
    //返回的是json字符串  
    return reader.ReadToEnd();  
}
```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'  
data = {  
    'FAction': 'OpenLockControl',  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',  
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c'  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 查询锁状态

## 简要描述：

- 查询锁状态

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryLockStatus)
FAssetGUID	是	string	设备唯一标识

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "succeed",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 109: 开锁中 (每2S查询一次, 查询10次后, 若结果仍为 "Result": 109, 则解锁设备失败; 若结果为 "Result": 200, 则解锁设备成功)
- 111: 设备离线
- 116: 开锁密码错误
- 106: 设备不属于该用户
- 105: 系统异常
- 102: 参数错误

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryLockStatus\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
```

```

conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{FAction:\"QueryLockStatus\",FTokenID:\"3acef045-d302-4032-b
40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))

```

查询锁状态

```
{  
    //返回的是json字符串  
    return reader.ReadToEnd();  
}
```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'  
data = {  
    'FAction': 'QueryLockStatus',  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',  
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c'  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

**Node.js:**

## 远程控制

### 开锁密码信息

---

### 修改锁静态密码

---

### 查询修改静态密码指令执行状态

---

### 开启/关闭动态密码

---

### 查询开启/关闭动态密码指令执行状态

---

# 开锁密码信息

## 简要描述：

- 获取设备开锁密码信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAssetPassword)
FGUIDs	是	string	锁唯一标识，多个用英文半角逗号隔开
FPageSize	否	Int	每页条数,缺省值为10
FPageIndex	否	Int	当前页码,缺省值为1

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FAssetGUID": "3a70d699-ab9d-4426-b394-3319874a7706",
      "FAssetID": "7200622909",
      "FAssetTypeID": 1701,
      "FAgentGUID": "4b3dd67a-00af-4118-87bf-fa9583a83b25",
      "FDynPsd": 1,
      "FStaticPsd": "123456",
      "FDynamicPsd": "677831",
      "RowNo": 1
    },
    {
      "FAssetGUID": "9c1d9a6d-9daf-4e8c-b409-0127743325ce",
      "FAssetID": "7581213453",
      "FAssetTypeID": 701,
      "FAgentGUID": "4b3dd67a-00af-4118-87bf-fa9583a83b25",
      "FDynPsd": 0,
      "FStaticPsd": "123456",
      "FDynamicPsd": null,
      "RowNo": 2
    }
  ]
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
FAssetGUID	String	设备唯一标识

FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FAgentGUID	String	公司唯一标识
FDynPsd	Int	动态密码开启状态
FStaticPsd	String	静态密码
FDynamicPsd	String	动态密码

## 备注:

- 更多返回错误代码如下:
- 105: 系统异常
- 102: 参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryAssetPassword\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUIDs: \"3A70D699-AB9D-4426-B394-3319874A7706,9C1D9A6D-9DAF-4E8C-B409-0127743325CE\", FPageSize: \"20\", FPageIndex: \"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
```

```

pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{FAction: \"QueryAssetPassword\", FTokenID: \"3acef045-d302-403
2-b40a-d9ee6c1519cd\", FGUIDs: \"3A70D699-AB9D-4426-B394-3319874A7706,9C1D9A6
D-9DAF-4E8C-B409-0127743325CE\", FPageSize: \"20\", FPageIndex: \"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    'FAction': 'QueryAssetPassword',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': '3A70D699-AB9D-4426-B394-3319874A7706,9C1D9A6D-9DAF-4E8C-B40

```

```
9-0127743325CE' ,
  'FPageSize': 20 ,
  'FPageIndex': 1
}
data = parse.urlencode(data).encode('utf-8')
headers = {
  'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
              r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
  'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

#### Node.js:

# 修改锁密码

## 简要描述：

- 修改锁静态密码

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateLockPassword)
FAssetGUID	是	string	设备唯一标识
FNewPassword	是	string	新密码
FOldPassword	是	string	旧密码

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
-----	----	----

## 备注:

- 更多返回错误代码如下:
- 105: 系统异常
- 102: 参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"UpdateLockPassword\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"3A70D699-AB9D-4426-B394-3319874A7706\", FNewPassword: \"123456\", FOldPassword: \"123478\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
```

```

conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction: \"UpdateLockPassword\", FTokenID: \"3acef045-d302-403
2-b40a-d9ee6c1519cd\", FAssetGUID: \"3A70D699-AB9D-4426-B394-3319874A7706\", F
NewPassword: \"123456\", FOldPassword: \"123478\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))

```

```
{  
    //返回的是json字符串  
    return reader.ReadToEnd();  
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'  
data = {  
    'FAction': 'UpdateLockPassword' ,  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd' ,  
    'FGUID': '3A70D699-AB9D-4426-B394-3319874A7706' ,  
    'FNewPassword': '123456' ,  
    'FOldPassword': '123478'  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
            r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

### Node.js:

# 查询修改静态密码指令执行状态

## 简要描述：

- 查询修改静态密码指令执行状态

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryUpdateLockPasswordStatus)
FAssetGUID	是	string	设备唯一标识

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "succeed",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 109: 开锁中 (每2s查询1次, 查询10次后还是109则失败)
- 102: 参数错误

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryUpdateLockPasswordStatus\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
```

```

conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryUpdateLockPasswordStatus\", FTokenID: \"3acef0
45-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8
c5f6a3c\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    'FAction': 'QueryUpdateLockPasswordStatus',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                 r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

**Node.js:**

# 开启/关闭动态密码

## 简要描述：

- 开启/关闭动态密码

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (OpenCloseDynamicPassword)
FAssetGUID	是	string	设备唯一标识
FStatus	是	string	1:开启； 0: 关闭

## 返回示例：

**正确时返回:**

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 102: 参数错误
- 108: 指令发送失败
- 123: 设备不支持
- 106: 设备不属于该用户
- 105: 系统异常
- 123: 设备不支持

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"OpenCloseDynamicPassword\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\\",FStatus:\\"1\\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
```

```

URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"OpenCloseDynamicPassword\\",FTokenID:\\"3acef045-d3
02-4032-b40a-d9ee6c1519cd\\",FAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a
3c\\",FStatus:\\"1\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

```

```
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    'FAction': 'OpenCloseDynamicPassword' ,
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd' ,
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c' ,
    'FStatus':1
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

### Node.js:

# 查询开启/关闭动态密码指令执行状态

## 简要描述：

- 查询开启/关闭动态密码指令执行状态

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryOpenCloseDynamicPasswordStatus)
FAssetGUID	是	string	设备唯一标识
FStatus	是	Int	1: 开启, 0: 关闭

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "succeed",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

### 返回参数说明:

参数名	类型	说明
-----	----	----

### 备注:

- 更多返回错误代码如下:
- 109: 开锁中 (每2s查询1次, 查询10次后还是109则失败)
- 105: 系统异常
- 102: 参数错误

### 请求示例:

#### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryOpenCloseDynamicPasswordStatus\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\", FStatus: \"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
```

```

conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryOpenCloseDynamicPasswordStatus\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\", FStatus: \"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{

```

```
//返回的是json字符串  
return reader.ReadToEnd();  
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'  
data = {  
    'FAction': 'QueryOpenCloseDynamicPasswordStatus' ,  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd' ,  
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c' ,  
    'FStatus': 1  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

### Node.js:

# 查询设备授权卡

## 简要描述：

- 查询设备授权卡信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2022-05-09	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAssetCard)
FAssetGUIDs	是	string	设备唯一标识,多个以英文逗号隔开

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FCardNo": "0015291757",
      "FSyncTime": "2023-03-10T02:05:55.427",
      "FAuthorizedPersion": null,
      "FAuthorizedTime": "2023-03-10T02:05:55.427",
      "FAssetID": "79090400001",
      "FAssetTypeID": 1709,
      "FAssetGUID": "c7e11199-8edd-4378-994f-8625ab34cd14",
      "FAssetTypeName": "JT709A"
    }
  ]
}
```

#### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

#### 返回参数说明:

参数名	类型	说明
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FCardNo	String	卡号
FAssetTypeName	String	设备类型名称
FAuthorizedTime	DateTime	授权时间(UTC时间)
FSyncTime	DateTime	同步时间(UTC时间)
FAuthorizedPersion	String	授权人
FAssetGUID	String	设备唯一标识

## 备注:

- 更多返回错误代码如下:
- 105: 系统异常
- 102: 参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryAssetCard\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUIDs:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryAssetCard\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUIDs:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
Encoding encoding = Encoding.UTF8;
HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpResponse response = (HttpResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    'FAction': 'QueryAssetCard' ,
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd' ,
    'FAssetGUIDs': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

**Node.js:**



# 查询车辆拍摄的最新的照片

## 简要描述：

- 查询车辆拍摄的最新的照片

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2024-01-24	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryCameraImgsList)
FVehicleGUID	是	string	车辆唯一标识 如:3C611B54-623E-6FA8-3924-791C9CC94323
FStartTime	是	string	开始时间

FEndTime	是	String	结束时间
FVersion	是	String	版本号

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FVehicleGUID": "a4b40424-cfc4-4b78-84ed-6fbb0b7f28d2",
      "FVehicleName": "876208000137",
      "FAssetGuid": "a64b2e58-9305-4a01-9c51-092205123945",
      "FAssetID": "876208000137",
      "FAssetTypeID": 3705,
      "FCameraID": 1,
      "FType": 0,
      "FDateTime": "2024/1/24 6:17:21",
      "FLongitude": "0",
      "FLatitude": "0",
      "FReasonKey": 0,
      "FUrl": "http://jointech-705c-bucket.obs.ap-southeast-1.myhuaweicloud.com/OBS_LockFiles/876208000137_1706105841/876208000137_1706105841_08_00.jpeg"
    }
  ]
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明

FVehicleGUID	String	车辆ID
FVehicleName	String	车牌号
FAssetID	String	终端ID
FAssetTypeID	String	终端类型
FCameraID	String	摄像头ID
FGpsTime	DateTime	拍照时间
FLongitude	Double	拍照经度
FLatitude	Double	拍照纬度
FShortName	String	公司简称
FReasonKey	String	拍照原因，1表示定时拍照，2表示触发拍照，3表示即时拍照
FUrl	String	图片地址
FCustomAssetTypeName	String	自定义设备类型名称

## 备注:

- 更多返回错误代码如下:
- 105: 系统异常
- 102: 参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"QueryCameraImgsList\",\"FVehicleGUID\":\"a4b40424-cfc4-4b78-84ed-6fbb0b7f28d2\",\"FStartTime\":\"2024-01-23 16:00:00\", \"FEndTime\":\"2024-01-24 15:59:59\", \"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\", \"FTimeDifferent\":28800, \"FLanguage\":\"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
```

```

conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"QueryCameraImgsList\", \"FVehicleGUID\":\"a4b4
0424-cfc4-4b78-84ed-6fbb0b7f28d2\", \"FStartTime\":\"2024-01-23 16:00:00\", \"
FEndTime\":\"2024-01-24 15:59:59\", \"FTokenID\":\"3acef045-d302-4032-b40a-
d9ee6c1519cd\", \"FTimeDifferent\":28800, \"FLanguage\":\"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

```

```
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction"
payload = {
    "FAction": "QueryCameraImgsList",
    "FVehicleGUID": "a4b40424-cfc4-4b78-84ed-6fbb0b7f28d2",
    "FStartTime": "2024-01-23 16:00:00",
    "FEndTime": "2024-01-24 15:59:59",
    "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd",
    "FTimeDifferent": 28800,
    "FLanguage": "1"
}
headers = {
    "Authorization": "eyJhbGciOiJIUzUxMiJ9.eyJsb2dpbiI6Im91d2V5X2tleSI6IjZjNTg4Yjk2LTkyNTQtNDcyMC1iYzMwLWQ0ZTI0Mzc4YzMyMCJ9.oTjYmJFpmPDSqFX8oHkzdBMYhidGWxDaeygdKY3SphFrqx3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkvaNMQxMg",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0",
    "Accept": "application/json, text/plain, */*",
    "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate",
    "Content-Type": "application/json",
    "Origin": "http://120.25.245.20:8081",
    "Connection": "keep-alive",
    "Referer": "http://120.25.245.20:8081/",
    "content-type": "application/json"
}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)
```

### Node.js:

```
`
```

```
` **
```

查询车辆拍摄的最新的照片

# 主动拍照

## 简要描述：

- 主动拍照

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2024-01-24	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (ActivePhotograph)
FAssetGUID	是	string	终端唯一标识
FT_GpActivePhotograph	是	Object	

### FT\_GpActivePhotograph

参数名	是否必须	类型	说明
FCameraID	是	String	摄像头ID，针对705设备，此字段不用传
FInterval	是	int	图像回传间隔 以秒为单位,最大值为3600秒.若设置此值大于0，则表示录像，JT705范围为1-60s.
FCount	是	int	图像回传张数.若设置此值为 0 则表示不再上传图片数据,JT705最大值为10

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

### 备注:

- 更多返回错误代码如下:
- 105: 系统异常
- 108: 发送失败
- 102: 参数错误

## 请求示例:

### Java:

```
String result = "";
```

```

//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"ActivePhotograph\",\"FAssetGUID\":\"a64b2e58-9305-4a01-9c51-092205123945\",\"FT_GpActivePhotograph\":{\"FAssetID\":\"876208000137\",\"FCount\":1,\"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\",\"FTimeDifferent\":28800,\"FLanguage\":\"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"ActivePhotograph\",\"FAssetGUID\":\"a64b2e58-9305-4a01-9c51-092205123945\",\"FT_GpActivePhotograph\":{\"FAssetID\":\"876208000137\",\"FCount\":1,\"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\",\"FTimeDifferent\":28800,\"FLanguage\":\"1\"}";
Encoding encoding = Encoding.UTF8;

```

```

HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction"

payload = {
    "FAction": "ActivePhotograph",
    "FAssetGUID": "a64b2e58-9305-4a01-9c51-092205123945",
    "FT_GpActivePhotograph": {
        "FAssetID": "876208000137",
        "FCount": 1
    },
    "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd",
    "FTimeDifferent": 28800,
    "FLanguage": "1"
}

headers = {
    "Authorization": "eyJhbGciOiJIUzUxMiJ9.eyJsb2dpbl91c2VyX2tleSI6IjZjNTg4Yjk2LTkyNTQtNDcyMC1iYzMwLWQ0ZTI0Mzc0YzMyMCJ9.otjYmJFpmPDSqFX8oHkzdBMYhidGwxDaeygdKY3SphFrqx3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkVAnMQxMg",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0",
    "Accept": "application/json, text/plain, */*",
    "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate",
    "Content-Type": "application/json",
    "Origin": "http://120.25.245.20:8081",
    "Connection": "keep-alive",

```

```

    "Referer": "http://120.25.245.20:8081/",
    "content-type": "application/json"
  }

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

```

### Node.js:

```

const request = require('request');

const options = {
  method: 'POST',
  url: 'https://cloud.assetscontrols.com:4443/OpenApi/Instruction',
  headers: {
    Authorization: 'eyJhbGciOiJIUzUxMiJ9.eyJsb2dpbl91c2VyX2tleSI6IjZjNTg4Yjk2LTkyNTQtNDcyMC1iYzMwLWQ0ZTI0Mzc4YzMyMCJ9.otjYmJFpmPDSqFX8oHkzdBMYhidGwxDayegdKY3SphFrqx3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkVxANMQxMg',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0',
    Accept: 'application/json, text/plain, */*',
    'Accept-Language': 'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2',
    'Accept-Encoding': 'gzip, deflate',
    'Content-Type': 'application/json',
    Origin: 'http://120.25.245.20:8081',
    Connection: 'keep-alive',
    Referer: 'http://120.25.245.20:8081/',
    'content-type': 'application/json'
  },
  body: {
    FAction: 'ActivePhotograph',
    FAssetGUID: 'a64b2e58-9305-4a01-9c51-092205123945',
    FT_GpActivePhotograph: {FAssetID: '876208000137', FCount: 1},
    FTokenID: '3acef045-d302-4032-b40a-d9ee6c1519cd',
    FTimeDifferent: 28800,
    FLanguage: '1'
  },
  json: true
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);

```

主动拍照

```
console.log(body);  
});
```

# 查询设备的联网状态

## 简要描述：

- 查询设备联网状态

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2024-01-24	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Monitor>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Monitor>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAssetOnlineStatus)
FAssetGUID	是	string	设备唯一标识

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "FAssetID": "876208000034",
    "FStatus": 3,
    "FGPSTime": "2023-10-07T07:40:26Z",
    "FRecvTime": "2023-10-07T07:40:31.631Z",
    "FLongitude": 0,
    "FLatitude": 0
  }
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

### 返回参数说明:

参数名	类型	说明
FAssetID	String	设备号
FStatus	String	状态（0：未激活 1：联网中 2：休眠中 3：离线）
FGPSTime	String	GPS时间
FRecvTime	String	接收时间
FLongitude	Double	经度
FLatitude	Double	纬度

### 备注:

- 更多返回错误代码如下：
- 105：系统异常
- 102：参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\r\n    \"FAction\": \"QueryAssetOnlineStatus\", \r\n    \"FAssetGUID\": \"76e63f83-b869-4431-af27-a5dbecb5a49c\", \r\n    \"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\", \r\n    \"FTimeDifferent\": 28800, \r\n    \"FLanguage\": \"1\"\r\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```
//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数,建议用对象传入
```

```

string body = "{\r\n    \"FAction\": \"QueryAssetOnlineStatus\", \r\n    \"FAssetGUID\": \"76e63f83-b869-4431-af27-a5dbecb5a49c\", \r\n    \"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\", \r\n    \"FTimeDifferent\": 28800, \r\n    \"FLanguage\": \"1\"\r\n}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/>";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

## Python :

```

url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor"

payload = {
    "FAction": "QueryAssetOnlineStatus",
    "FAssetGUID": "76e63f83-b869-4431-af27-a5dbecb5a49c",
    "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd",
    "FTimeDifferent": 28800,
    "FLanguage": "1"
}

headers = {
    "Authorization": "eyJhbGciOiJIUzUxMiJ9.eyJsb2dpbl91c2VyX2tleSI6IjZjNTg4Yjk2LTkyNTQtNDcyMCIiYzpwLWQ0ZTI0Mzc4YzMyMCI9Lm9tYmJFpmPDSqFX8oHkzdBMhidGwxDaeygdKY3SphFrXq3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkvanMQxMg",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0",
    "Accept": "application/json, text/plain, */>",
    "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate",
    "Content-Type": "application/json",
}

```

```

    "Origin": "http://120.25.245.20:8081",
    "Connection": "keep-alive",
    "Referer": "http://120.25.245.20:8081/",
    "content-type": "application/json"
  }

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

```

## Node.js:

```

const request = require('request');

const options = {
  method: 'POST',
  url: 'https://cloud.assetscontrols.com:3443/OpenApi/Monitor',
  headers: {
    Authorization: 'eyJhbGciOiJIUzUxMiJ9.eyJsb2dpbl91c2VyX2tleSI6IjZjNTg4Yj
k2LTkyNTQtNDcyMC1iYzMwLWQ0ZTI0Mzc4YzMyMCJ9.otjYmJFpmPDSqFX8oHkzdBMYhidGwxDa
eygdKY3SphFrXq3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkVAnMQxMg',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Geck
o/20100101 Firefox/121.0',
    Accept: 'application/json, text/plain, */*',
    'Accept-Language': 'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,
en;q=0.2',
    'Accept-Encoding': 'gzip, deflate',
    'Content-Type': 'application/json',
    Origin: 'http://120.25.245.20:8081',
    Connection: 'keep-alive',
    Referer: 'http://120.25.245.20:8081/',
    'content-type': 'application/json'
  },
  body: {
    FAction: 'QueryAssetOnlineStatus',
    FAssetGUID: '76e63f83-b869-4431-af27-a5dbecb5a49c',
    FTokenID: '3acef045-d302-4032-b40a-d9ee6c1519cd',
    FTimeDifferent: 28800,
    FLanguage: '1'
  },
  json: true
};

request(options, function (error, response, body) {

```

```
if (error) throw new Error(error);  
  
console.log(body);  
});
```

# 新主动拍照(只支持705C的CAMERA9\_4)

## 简要描述：

- 新的主动拍照,只支持CAMERA9\_4指令,其它暂时用不上

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2026-02-05	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (ActiveNewPhotograph)
FAssetGUID	是	string	终端唯一标识
FT_GpActiveNewPhotograph	是	Object	

### FT\_GpActiveNewPhotograph

参数名	是否必须	类型	说明
FCameraID	是	String	摄像头类型,0:主摄,1顶摄,目前只支持一个摄像头,默认0
FType	是	int	操作类型,1:抓拍,2:录像,3:抓拍+录像
FCount	是	int	抓拍张数,仅在操作类型1和3生效,其余填0即可
FDuration	是	int	录像时长, 单位: 秒; 仅在操作类型2和3生效,其余填0即可
FResolution	是	int	分辨率,0:360p,1:480P,2:720P,3:1080P
FInterval	是	int	抓拍间隔,单位秒.

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": {
    "terminalStrId": "c5b53001-8755-4269-a9c8-b71ca04c6a0d",
    "terminalNum": "804250900200",
    "dateTime": "2025-12-20T06:58:41.04655Z",
    "dateTimestamp": 1766213921046,
    "response": "(804250900200,1,001,CAMERA,9,4,0,700160818000_1769373855)"
  }
}
```

指令应答说明:

(804250900200,1,001,CAMERA,9,4,0,700160818000\_1769373855)

0:执行成功, 1:摄像头处于忙

700160818000\_1769373855:当前关联的文件名称序号

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

}

## 返回参数说明:

参数名	类型	说明
terminalStrId	String	设备唯一标识号
terminalNum	String	设备号
dateTime	String	指令时间
dateTimestamp	String	指令时间戳
response	String	指令回复内容

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误
- 106: 无此设备
- 111: 设备不在线
- 131: 未集成此指令
- 108: 指令发送失败
- 138: 指令执行超时(指令发送成功,等待15秒,设备也未有结果返回)

## 请求示例:

Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"ActiveNewPhotograph\",\"FAssetGUID\":\"a64b2e58-9305-4a01-9c51-092205123945\",\"FT_GpActiveNewPhotograph\":{\"FCameraID\":0,\"FType\":1,\"FCount\":3,\"FDuration\":0,\"FResolution\":2,\"FInterval\":2},\"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\",\"FTimeDifferent\":28800,\"FLanguage\":\"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
```

```

conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"ActiveNewPhotograph\",\"FAssetGUID\":\"a64b2e
58-9305-4a01-9c51-092205123945\",\"FT_GpActiveNewPhotograph\":{\"FCameraID\
\":0,\"FType\":1,\"FCount\":3,\"FDuration\":0,\"FResolution\":2,\"FInterval\
\":2},\"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\",\"FTimeDifferen
t\":28800,\"FLanguage\":\"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);

```

```
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

## Python :

```
url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction"

payload = {
    "FAction": "ActiveNewPhotograph",
    "FAssetGUID": "a64b2e58-9305-4a01-9c51-092205123945",
    "FT_GpActiveNewPhotograph": {
        "FCameraID": 0,
        "FType": 1,
        "FCount": 1,
        "FDuration": 0,
        "FResolution": 2,
        "FInterval": 2,
    },
    "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd",
    "FTimeDifferent": 28800,
    "FLanguage": "1"
}

headers = {
    "Authorization": "eyJhbGciOiJIUzUxMiJ9.eyJsb2dpb191c2VyX2tleSI6IjZjNTg4Yjk2LTkyNTQtNDcyMCIiYzMyMwLWQ0ZTI0Mzc4YzMyMjY5LmVjYmJmPmPDSqFX8oHkzdBMYhidGwxDaeygdKY3SphFrqx3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkvaNMQxMg",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0",
    "Accept": "application/json, text/plain, */*",
    "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate",
    "Content-Type": "application/json",
    "Origin": "http://120.25.245.20:8081",
    "Connection": "keep-alive",
    "Referer": "http://120.25.245.20:8081/",
    "content-type": "application/json"
}

response = requests.request("POST", url, json=payload, headers=headers)
```

```
print(response.text)
```

### Node.js:

```
const request = require('request');

const options = {
  method: 'POST',
  url: 'https://cloud.assetscontrols.com:4443/OpenApi/Instruction',
  headers: {
    Authorization: 'eyJhbGciOiJIUzUxMiJ9.eyJsb2dpbl91c2VyX2tleSI6IjZjNTg4Yjk2LTkyNTQtNDcyMC1iYzMwLWQ0ZTI0Mzc4YzMyMCJ9.otjYmJFpmPDSqFX8oHkzdBMYhidGwxDaygdKY3SphFrXq3yRzyPOq80kFUW1CH-gVW8M6tRlzTrPkvANMQxMg',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0',
    Accept: 'application/json, text/plain, */*',
    'Accept-Language': 'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2',
    'Accept-Encoding': 'gzip, deflate',
    'Content-Type': 'application/json',
    Origin: 'http://120.25.245.20:8081',
    Connection: 'keep-alive',
    Referer: 'http://120.25.245.20:8081/',
    'content-type': 'application/json'
  },
  body: {
    FAction: 'ActiveNewPhotograph',
    FAssetGUID: 'a64b2e58-9305-4a01-9c51-092205123945',
    FT_GpActiveNewPhotograph: {FAssetID: '876208000137', FCount: 1},
    FTokenID: '3acef045-d302-4032-b40a-d9ee6c1519cd',
    FTimeDifferent: 28800,
    FLanguage: '1'
  },
  json: true
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);

  console.log(body);
});
```



## 报表查询

[获取开关锁报表](#)

---

[获取从机轨迹报表](#)

---

[获取温湿度曲线报表](#)

---

[获取报警消息报表](#)

---

# 状态曲线

## 简要描述：

- 状态曲线

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-08-07	2023-08-07

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportStatusTrack )
FGUID	是	string	车辆/设备唯一标识
FStartTime	是	string	开始时间(UTC时间) 例("2023-07-08 16:00:00")
FEndTime	是	string	结束时间(UTC时间) 例( "2023-08-07 15:59:59")
FSelectType	是	int	查询类型(0:车辆 1:设备)

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FVehicleName": "837303000449",
      "FAssetID": "837303000449",
      "FGPSTime": "2023-07-10T09:40:32",
      "Bat": 76,
      "TH": "25.1°C(60%)",
      "Lx": 0,
      "Vib": 4519,
      "Press": 0,
      "Acce": "x:-2880;y:1376;z:-3200",
      "Pos": "x:-2880;y:1376;z:-3200",
      "FD": -1,
      "FLongitude": 113.918228424976,
      "FLatitude": 22.6726779622259,
      "FDir": 0,
      "FTemp1": 25.1000003814697,
      "FTemp2": -1000,
      "FHum1": 60,
      "FHum2": 0
    }
  ]
}
```

### 错误时返回:

```
//表示传入的车辆或设备为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FGUID is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

}

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 返回参数说明:

参数名	类型	说明
FVehicleName	String	车辆ID
FAssetID	Int	设备ID
FGPSTime	String	GPS时间
Bat	Int	电量
TH	String	温湿值
Lx	Int	光照度
Vib	Int	震幅
Press	Int	气压
Acce	String	加速度
Pos	String	姿态
FD	Int	车门状态
FLongitude	Double	当前经度
FLatitude	Double	当前纬度
FDir	Int	描述
FTemp1	Double	温度1
FTemp2	Double	温度2
FHum1	Double	湿度1

FHum2	Double	湿度2
-------	--------	-----

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\r\n  \"FAction\": \"QueryReportStatusTrack\",\r\n  \"FGUID\": \"f6538348-bccf-4da2-86f0-6659cfcaa33d\",\r\n  \"FStartTime\": \"2023-07-08 16:00:00\",\r\n  \"FEndTime\": \"2023-08-07 15:59:59\",\r\n  \"FSelect Type\": 0,\r\n  \"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\",\r\n  \"FTimeDifferent\": 28800,\r\n  \"FLanguage\": \"1\"\r\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"QueryReportStatusTrack\",
    \"FGUID\": \"f6538348-bccf-4da2-86f0-6659cfcaa33d\",
    \"FStartTime\": \"2023-07-08 16:00:00\",
    \"FEndTime\": \"2023-08-07 15:59:59\",
    \"FSelectType\": 0,
    \"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'

data = '{ "FAction": "QueryReportStatusTrack", "FGUID": "f6538348-bccf-4da2-86f0-6659cfcaa33d", "FStartTime": "2023-07-08 16:00:00", "FEndTime": "2023-08-07 15:59:59", "FSelectType": 0, "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd", "FTimeDifferent": 28800, "FLanguage": "1" }'

data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',

```

```
'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 开关锁报表

## 简要描述：

- 获取开关锁报表

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-08-20	2020-08-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportOpenLockList)
FGUIDs	是	string	车辆/设备唯一标识（多个以英文逗号隔开）
FSelectType	是	int	1：按车辆查；0：按终端查
FStartTime	是	datetime	开始时间 (UTC时间)

FEndTime	是	datetime	结束时间 (UTC时间)
FDateType	是	int	1: 按接收时间查; 0: 按GPS时间查
#### 返回示例:			

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "be2a1e27-b36e-4301-b2c8-465a62ad2153",
      "FVehicleGUID": "be2a1e27-b36e-4301-b2c8-465a62ad215f",
      "FVehicleName": "粤BXX3453",
      "FAssetGUID": "9c1d9a6d-9daf-4e8c-b409-0127743325ce",
      "FAssetID": "7581213453",
      "FAssetTypeID": 701,
      "FDateTime": "2020-08-20T01:55:35",
      "FLongitude": 113.917915,
      "FLatitude": 22.580486,
      "FMileage": 1,
      "FType": 1,
      "FOpenType": 2,
      "FUserName": ""
    },
    {
      "FGUID": "be2a1e27-b36e-4301-b2c8-465a62ad2252",
      "FVehicleGUID": "be2a1e27-b36e-4301-b2c8-465a62ad215f",
      "FVehicleName": "粤BXX3453",
      "FAssetGUID": "9c1d9a6d-9daf-4e8c-b409-0127743325ce",
      "FAssetID": "7581213453",
      "FAssetTypeID": 701,
      "FDateTime": "2020-08-20T02:38:03",
      "FLongitude": 113.917475,
      "FLatitude": 22.580903,
      "FMileage": 1,
      "FType": 1,
      "FOpenType": 2,
      "FUserName": ""
    }
  ]
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
FGUID	String	开关锁唯一标识
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车牌号
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FDateTime	DateTime	开关锁GPS时间
FLongitude	Decimal	经度
FLatitude	Decimal	纬度
FMileage	Int	里程
FType	Int	锁状态（1：开；0：关）
FOpenType	Int	开锁类型 (-3:应急仓关 -2拍照上锁 -1:关锁 1: 刷卡； 2: 远程； 3: 蓝牙 4:剪绳开锁 5:短信开锁 6:按键开锁 7:NFC开锁 8:进区域触发开锁 9:应急仓开)
FCreateTime	DateTime	创建时间
FUserName	String	开锁人

**备注:**

- 更多返回错误代码如下：

- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryReportOpenLockList\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUIDs: \"BE2A1E27-B36E-4301-B2C8-465A62AD215F, FE0BDE75-4A30-4B9C-AD2E-66D8A4F44F5E\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:00:00\", FEndTime: \"2020-08-25 15:59:59\", FDateType: \"0\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction: \"QueryReportOpenLockList\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUIDs: \"BE2A1E27-B36E-4301-B2C8-465A62AD215F, FE0BDE75-4A30-4B9C-AD2E-66D8A4F44F5E\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:00:00\", FEndTime: \"2020-08-25 15:59:59\", FDateType: \"0\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Report'
data = {
    'FAction': 'QueryReportOpenLockList',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': 'BE2A1E27-B36E-4301-B2C8-465A62AD215F, FE0BDE75-4A30-4B9C-AD2E-66D8A4F44F5E',
    'FSelectType': 1,
    'FStartTime': '2020-08-05 16:00:00',
    'FEndTime': '2020-08-12 15:59:59',
    'FDateType': 0
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}

```

```
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

### Node.js:

# 从机记录

## 简要描述：

- 获取从机记录

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-08-20	2020-08-20
2.0.0	lenny		2021-11-15

## 请求URL:

- 国际客户：<https://icloud.jointcontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportSubAssetTrack)
FGUID	是	string	车辆/设备唯一标识
FSelectType	是	int	1: 按车辆查; 0: 按终端查

FStartTime	是	datetime	开始时间 (UTC时间)
FEndTime	是	datetime	结束时间 (UTC时间)
#### 返回示例:			

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",
      "FAssetID": "7500313618",
      "FGPSTime": "2020-08-24T05:11:27",
      "FRecvTime": "2020-08-24T05:12:36",
      "FLongitude": 113.916800,
      "FLatitude": 22.580777,
      "FSpeed": 0,
      "FSubInfo": "{\"E0171A0107\":{\"FAssetID\":\"E0171A0107\",\"FDateTimestamp\":\"2020-08-24T04:01:01Z\",\"FDateTimestamp\":1598241661000,\"FEvent\":6,\"FFlowId\":0,\"FHumidity\":0,\"FLockRope\":0,\"FLockStatus\":0,\"FLockTimes\":1038,\"FPower\":68,\"FRSSI\":117,\"FSensorType\":4,\"FTemperature\":-1000.0,\"FVoltage\":\"3.93\"}}}"
    },
    {
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",
      "FAssetID": "7500313618",
      "FGPSTime": "2020-08-24T05:11:27",
      "FRecvTime": "2020-08-24T05:14:39",
      "FLongitude": 113.916800,
      "FLatitude": 22.580777,
      "FSpeed": 0,
      "FSubInfo": "{\"E0171A0107\":{\"FAssetID\":\"E0171A0107\",\"FDateTimestamp\":\"2020-08-24T04:01:01Z\",\"FDateTimestamp\":1598241661000,\"FEvent\":6,\"FFlowId\":0,\"FHumidity\":0,\"FLockRope\":0,\"FLockStatus\":0,\"FLockTimes\":1038,\"FPower\":68,\"FRSSI\":117,\"FSensorType\":4,\"FTemperature\":-1000.0,\"FVoltage\":\"3.93\"}}}"
    },
    {
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",
      "FAssetID": "7500313618",
      "FGPSTime": "2020-08-24T05:11:27",
      "FRecvTime": "2020-08-24T05:17:43",

```

```
"FLongitude": 113.916800,  
"FLatitude": 22.580777,  
"FSpeed": 0,  
"FSubInfo": "{\"E0171A0107\":{\"FAssetID\":\"E0171A0107\",\"FDate  
teTime\":\"2020-08-24T04:01:01Z\",\"FDateTimestamp\":1598241661000,\"FEvent  
\":6,\"FFlowId\":0,\"FHumidity\":0,\"FLockRope\":0,\"FLockStatus\":0,\"FLoc  
kTimes\":1038,\"FPower\":68,\"FRSSI\":117,\"FSensorType\":4,\"FTemperature\  
\":-1000.0,\"FVoltage\":\"3.93\"}}"  
},  
{  
  "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
  "FAssetID": "7500313618",  
  "FGPSTime": "2020-08-24T05:11:27",  
  "FRecvTime": "2020-08-24T05:19:45",  
  "FLongitude": 113.916800,  
  "FLatitude": 22.580777,  
  "FSpeed": 0,  
  "FSubInfo": "{\"E0171A0107\":{\"FAssetID\":\"E0171A0107\",\"FDate  
teTime\":\"2020-08-24T04:01:01Z\",\"FDateTimestamp\":1598241661000,\"FEvent  
\":6,\"FFlowId\":0,\"FHumidity\":0,\"FLockRope\":0,\"FLockStatus\":0,\"FLoc  
kTimes\":1038,\"FPower\":68,\"FRSSI\":117,\"FSensorType\":4,\"FTemperature\  
\":-1000.0,\"FVoltage\":\"3.93\"}}"  
},  
{  
  "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
  "FAssetID": "7500313618",  
  "FGPSTime": "2020-08-24T05:41:37",  
  "FRecvTime": "2020-08-24T05:41:58",  
  "FLongitude": 113.917333,  
  "FLatitude": 22.581298,  
  "FSpeed": 0,  
  "FSubInfo": "{\"E0171A0107\":{\"FAssetID\":\"E0171A0107\",\"FDate  
teTime\":\"2020-08-24T04:31:21Z\",\"FDateTimestamp\":1598243481000,\"FEvent  
\":6,\"FFlowId\":1,\"FHumidity\":0,\"FLockRope\":0,\"FLockStatus\":0,\"FLoc  
kTimes\":1038,\"FPower\":68,\"FRSSI\":118,\"FSensorType\":4,\"FTemperature\  
\":-1000.0,\"FVoltage\":\"3.93\"}}"  
},  
{  
  "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
  "FAssetID": "7500313618",  
  "FGPSTime": "2020-08-24T05:41:37",  
  "FRecvTime": "2020-08-24T05:46:04",  
  "FLongitude": 113.917333,  
  "FLatitude": 22.581298,  
  "FSpeed": 0,  
  "FSubInfo": "{\"E0171A0107\":{\"FAssetID\":\"E0171A0107\",\"FDate
```

```
teTime\":"2020-08-24T04:31:21Z",\FDateTimestamp\:1598243481000,\FEvent
\:6,\FFlowId\:1,\FHumidity\:0,\FLockRope\:0,\FLockStatus\:0,\FLoc
kTimes\:1038,\FPower\:68,\FRSSI\:118,\FSensorType\:4,\FTemperature
\":-1000.0,\FVoltage\":"3.93\}}}"
},
{
  "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",
  "FAssetID": "7500313618",
  "FGPSTime": "2020-08-24T05:41:37",
  "FRecvTime": "2020-08-24T05:49:08",
  "FLongitude": 113.917333,
  "FLatitude": 22.581298,
  "FSpeed": 0,
  "FSubInfo": "{\E0171A0107\":{\FAssetID\":"E0171A0107",\FDa
teTime\":"2020-08-24T04:31:21Z",\FDateTimestamp\:1598243481000,\FEvent
\:6,\FFlowId\:1,\FHumidity\:0,\FLockRope\:0,\FLockStatus\:0,\FLoc
kTimes\:1038,\FPower\:68,\FRSSI\:118,\FSensorType\:4,\FTemperature
\":-1000.0,\FVoltage\":"3.93\}}}"
},
{
  "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",
  "FAssetID": "7500313618",
  "FGPSTime": "2020-08-24T05:11:27",
  "FRecvTime": "2020-08-24T05:11:35",
  "FLongitude": 113.916800,
  "FLatitude": 22.580777,
  "FSpeed": 0,
  "FSubInfo": "{\E0171A0107\":{\FAssetID\":"E0171A0107",\FDa
teTime\":"2020-08-24T04:01:01Z",\FDateTimestamp\:1598241661000,\FEvent
\:6,\FFlowId\:0,\FHumidity\:0,\FLockRope\:0,\FLockStatus\:0,\FLoc
kTimes\:1038,\FPower\:68,\FRSSI\:117,\FSensorType\:4,\FTemperature
\":-1000.0,\FVoltage\":"3.93\}}}"
},
{
  "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",
  "FAssetID": "7500313618",
  "FGPSTime": "2020-08-24T06:25:12",
  "FRecvTime": "2020-08-24T06:25:40",
  "FLongitude": 113.917333,
  "FLatitude": 22.580717,
  "FSpeed": 0,
  "FSubInfo": "{\1018330094\":{\FAssetID\":"1018330094",\FDa
teTime\":"2020-08-23T16:22:29Z",\FDateTimestamp\:1598199749000,\FEvent
\":-1,\FFlowId\:236,\FHumidity\:75,\FLockRope\:1,\FLockStatus\:1,\F
LockTimes\":-1,\FPower\:96,\FRSSI\:97,\FSensorType\:1,\FTemperature
\":28.3,\FVoltage\":"3.00\}}}"
```

```
    },  
    {  
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
      "FAssetID": "7500313618",  
      "FGPSTime": "2020-08-24T06:25:12",  
      "FRecvTime": "2020-08-24T06:26:40",  
      "FLongitude": 113.917333,  
      "FLatitude": 22.580717,  
      "FSpeed": 0,  
      "FSubInfo": "{\"1018330094\":{\"FAssetID\":\"1018330094\",\"FDate  
teTime\":\"2020-08-23T16:22:29Z\",\"FDateTimestamp\":1598199749000,\"FEvent  
\":-1,\"FFlowId\":236,\"FHumidity\":75,\"FLockRope\":1,\"FLockStatus\":1,\"  
FLockTimes\":-1,\"FPower\":96,\"FRSSI\":97,\"FSensorType\":1,\"FTemperature  
\":28.3,\"FVoltage\":\"3.00\"}}"  
    },  
    {  
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
      "FAssetID": "7500313618",  
      "FGPSTime": "2020-08-24T06:25:12",  
      "FRecvTime": "2020-08-24T06:27:42",  
      "FLongitude": 113.917333,  
      "FLatitude": 22.580717,  
      "FSpeed": 0,  
      "FSubInfo": "{\"1018330094\":{\"FAssetID\":\"1018330094\",\"FDate  
teTime\":\"2020-08-23T16:22:29Z\",\"FDateTimestamp\":1598199749000,\"FEvent  
\":-1,\"FFlowId\":236,\"FHumidity\":75,\"FLockRope\":1,\"FLockStatus\":1,\"  
FLockTimes\":-1,\"FPower\":96,\"FRSSI\":97,\"FSensorType\":1,\"FTemperature  
\":28.3,\"FVoltage\":\"3.00\"}}"  
    },  
    {  
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
      "FAssetID": "7500313618",  
      "FGPSTime": "2020-08-24T06:25:12",  
      "FRecvTime": "2020-08-24T06:29:46",  
      "FLongitude": 113.917333,  
      "FLatitude": 22.580717,  
      "FSpeed": 0,  
      "FSubInfo": "{\"1018330094\":{\"FAssetID\":\"1018330094\",\"FDate  
teTime\":\"2020-08-23T16:22:29Z\",\"FDateTimestamp\":1598199749000,\"FEvent  
\":-1,\"FFlowId\":236,\"FHumidity\":75,\"FLockRope\":1,\"FLockStatus\":1,\"  
FLockTimes\":-1,\"FPower\":96,\"FRSSI\":97,\"FSensorType\":1,\"FTemperature  
\":28.3,\"FVoltage\":\"3.00\"}}"  
    },  
    {  
      "FAssetGUID": "8e762610-ed26-4974-ac04-87710890e8ab",  
      "FAssetID": "7500313618",
```

```

"FGPSTime": "2020-08-24T06:25:12",
"FRcvTime": "2020-08-24T06:30:47",
"FLongitude": 113.917333,
"FLatitude": 22.580717,
"FSpeed": 0,
"FSubInfo": "{\"1018330094\":{\"FAssetID\":\"1018330094\",\"FDate
teTime\":\"2020-08-23T16:22:29Z\",\"FDateTimestamp\":1598199749000,\"FEvent
\":-1,\"FFlowId\":236,\"FHumidity\":75,\"FLockRope\":1,\"FLockStatus\":1,\"
FLockTimes\":-1,\"FPower\":96,\"FRSSI\":97,\"FSensorType\":1,\"FTemperature
\":28.3,\"FVoltage\":\"3.00\"}}}"
    }
  ]
}

```

### 错误时返回:

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

### 返回参数说明:

参数名	类型	说明
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID
FGPSTime	DateTime	GPS时间(UTC时间)
FRcvTime	DateTime	接收时间(UTC时间)
FLongitude	Decimal	经度
FLatitude	Decimal	纬度
FSpeed	Int	速度
FSubInfo	String	从机信息
FSubAssetID	String	从机ID
FSensorType	Int	从机类型 (1: JT126; 4: JT709;5:JT801;6:JT802)

FDateTime	DateTime	从机GPS时间(UTC时间)
FEvent	Int	事件类型
FTemperature	String	温度
FHumidity	String	湿度
FLockRope	Int	锁绳状态
FLockStatus	Int	锁状态
FLockTimes	Int	开锁次数
FPower	Int	电量值
FVoltage	float	电压值
FRSSI	Int	信号强度

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryReportSubAssetTrack\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUID: \"EABD84FC-0228-4836-A0A2-3404EC074A13\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:00:00\", FEndTime: \"2020-08-25 15:59:59\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
```

```

indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction:\"QueryReportSubAssetTrack\",FTokenID:\"3acef045-d3
02-4032-b40a-d9ee6c1519cd\",FGUID:\"EABD84FC-0228-4836-A0A2-3404EC074A13\",
FSelectType:\"1\",FStartTime:\"2020-08-18 16:00:00\",FEndTime:\"2020-08-25
15:59:59\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串

```

```
return reader.ReadToEnd();  
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Report'  
data = {  
    'FAction': 'QueryReportSubAssetTrack',  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',  
    'FGUID': 'EABD84FC-0228-4836-A0A2-3404EC074A13',  
    'FSelectType': 1,  
    'FStartTime': '2020-08-18 16:00:00',  
    'FEndTime': '2020-08-24 15:59:59',  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
            r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

### Node.js:

# 报警消息报表

## 简要描述：

- 获取报警消息报表

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-08-20	2020-08-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名(QueryMessageAlarmList)
FGUIDs	是	string	车辆/设备唯一标识；多个用英文半角逗号隔开
FSelectType	是	int	1：按车辆查；0：按终端查
FStartTime	是	datetime	开始时间 (UTC时间)

FEndTime	是	datetime	结束时间 (UTC时间)
FDataTypes	是	string	报警类型集合；多个用英文半角逗号隔开 (传空表示获取所有类型的报警)
FDateType	是	int	1: 按接收时间查；0: 按GPS时间查

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "0d6a1334-8eee-4b9b-822e-0042f4cedfd6",
      "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
      "FVehicleName": "44444",
      "FAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
      "FAssetID": "7560717001",
      "FAssetTypeID": 701,
      "FSubAssetID": null,
      "FDataType": 42,
      "FCurrentStatus": 3,
      "FStartTime": "2020-06-06T09:23:45",
      "FStartLongitude": 113.856122,
      "FStartLatitude": 22.585537,
      "FEndLongitude": 113.856122,
      "FEndLatitude": 22.585537,
      "FEndTime": "2020-06-06T09:23:45",
      "FStartSpeed": 0,
      "FEndSpeed": 0,
      "FKeepSeconds": 0,
      "FStartMileage": 38,
      "FEndMileage": 38,
      "FKeepDistance": 0,
      "FDescribeJSON": "",
      "FRelateGUID": "258dbdb5-1c01-495c-98f5-6165e9fa8b4c",
      "FWaybillInfo": "{\"FGUID\": \"258dbdb5-1c01-495c-98f5-6165e9fa8b4c\", \"FWaybill\": \"\"}"
    },
    {
      "FGUID": "cd2ec92b-6046-47c8-a9f4-00ebf6bde5a8",
      "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",

```

```
"FVehicleName": "44444",
"FAAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
"FAAssetID": "7560717001",
"FAAssetTypeID": 701,
"FSubAssetID": null,
"FDDataType": 42,
"FCurrentStatus": 3,
"FStartTime": "2020-06-06T05:21:42",
"FStartLongitude": 113.856103,
"FStartLatitude": 22.585535,
"FEndLongitude": 113.856103,
"FEndLatitude": 22.585535,
"FEndTime": "2020-06-06T05:21:42",
"FStartSpeed": 0,
"FEndSpeed": 0,
"FKeepSeconds": 0,
"FStartMileage": 23,
"FEndMileage": 23,
"FKeepDistance": 0,
"FDescribeJSON": "",
"FRelateGUID": "258dbdb5-1c01-495c-98f5-6165e9fa8b4c",
"FWaybillInfo": "{\"FGUID\":\"258dbdb5-1c01-495c-98f5-6165e9fa8
b4c\", \"FWaybill\":\"\"}"
  },
  {
    "FGUID": "1de8e107-41d8-45a7-b67e-00fd7ac6b6d7",
    "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
    "FVehicleName": "44444",
    "FAAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
    "FAAssetID": "7560717001",
    "FAAssetTypeID": 701,
    "FSubAssetID": null,
    "FDDataType": 42,
    "FCurrentStatus": 3,
    "FStartTime": "2020-06-05T17:20:24",
    "FStartLongitude": 113.856107,
    "FStartLatitude": 22.585558,
    "FEndLongitude": 113.856107,
    "FEndLatitude": 22.585558,
    "FEndTime": "2020-06-05T17:20:24",
    "FStartSpeed": 0,
    "FEndSpeed": 0,
    "FKeepSeconds": 0,
    "FStartMileage": 22,
    "FEndMileage": 22,
    "FKeepDistance": 0,
```

```

    "FDescribeJSON": "",
    "FRelateGUID": "258dbdb5-1c01-495c-98f5-6165e9fa8b4c",
    "FWaybillInfo": "{ \"FGUID\": \"258dbdb5-1c01-495c-98f5-6165e9fa8b4c\", \"FWaybill\": \"\" }"
  },
  {
    "FGUID": "52824179-d573-430c-b87f-0114cc62c032",
    "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
    "FVehicleName": "44444",
    "FAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
    "FAssetID": "7560717001",
    "FAssetTypeID": 701,
    "FSubAssetID": null,
    "FDataType": 44,
    "FCurrentStatus": 3,
    "FStartTime": "2020-06-05T21:20:48",
    "FStartLongitude": 113.856132,
    "FStartLatitude": 22.585435,
    "FEndLongitude": 113.856132,
    "FEndLatitude": 22.585435,
    "FEndTime": "2020-06-05T21:20:48",
    "FStartSpeed": 0,
    "FEndSpeed": 0,
    "FKeepSeconds": 0,
    "FStartMileage": 22,
    "FEndMileage": 22,
    "FKeepDistance": 0,
    "FDescribeJSON": "{ \"cardno\": \"0003587086\" }",
    "FRelateGUID": "258dbdb5-1c01-495c-98f5-6165e9fa8b4c",
    "FWaybillInfo": "{ \"FGUID\": \"258dbdb5-1c01-495c-98f5-6165e9fa8b4c\", \"FWaybill\": \"\" }"
  }
]
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FGUID	String	报警数据唯一标识
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车牌号
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FSubAssetID	String	从机ID
FDataType	Int	报警类型
FCurrentStatus	Int	当前状态（1：执行中；2：已完成；3：单点报警（瞬间的动作））
FStartTime	DateTime	开始GPS时间
FStartLongitude	Decimal	开始经度
FStartLatitude	Decimal	开始纬度
FEndLongitude	Decimal	结束经度
FEndLatitude	Decimal	结束纬度
FEndTime	DateTime	结束GPS时间
FStartSpeed	Int	开始速度（不显示）
FEndSpeed	Int	结束速度（不显示）
FKeepSeconds	Int	持续时长（单位s）
FStartMileage	Int	开始里程
FEndMileage	Int	结束里程
FKeepDistance	Int	持续距离
FCreateTime	DateTime	数据存储时间
FStartCell	String	开始小区码

FDescribeJSON	String	报警描述 (Json字符串)
FRelateGUID	String	报警关联的运单唯一标识
FWaybillInfo	String	事件绑定的运单信息 ,Json字符串

## FWaybillInfo 参数说明:

参数名	类型	说明
FGUID	String	运单数据唯一标识
FWaybillNo	String	运单号

## FDescribeJSON 参数说明

1.姿态报警 (FDataTypeoe: 197) :

例: {"type":1}

参数名	类型	说明
type	Int	姿态报警类型: 1:前倾 2:后倾 3:左倾 4:右倾

2.温湿度报警 (FDataTypeoe: 63)

例: {"F":"65.0","ID":"867730057935318","N":"27.600000381469727","T":4}

参数名	类型	说明
T	int	报警类型: 1:高温报警 2:低温报警 3:高湿报警 4:低湿报警 5:高光照 6:低光照 7:高气压 8:低气压
F	Double	门限值
N	Double	实际值
ID	String	传感器

3.碰撞预警 (FDataTypeoe: 29)

例: {"fAcceleration":{"x:65488;y:160;z:864"},"fVibrate":254}

参数名	类型	说明
-----	----	----

fAcceleration	String	三轴值
fVibrate	Int	震动值

## 4.电压异常(FDataTypoe: 341)

例:{"fCurrentValue":"23.5"}

参数名	类型	说明
fCurrentValue	String	电压值

## 5.电流异常(FDataTypoe: 342)

例:{"fCurrentValue":"23.5"}

参数名	类型	说明
fCurrentValue	String	电流值

## 6.门锁异常开启(FDataTypoe: 330)

例:{"fDoorExceptionNum":"1"}

参数名	类型	说明
fDoorExceptionNum	String	门锁1 或 门锁2

## 7.刷非法卡(FDataTypoe: 44)

例:{"cardno":"0003587086"}

参数名	类型	说明
cardno	String	卡号

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

**请求示例:**

**Java:**

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryMessageAlarmList\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FGUIDS:\"D128C716-9D21-4FD2-AC40-EF18A0724AF3,D118407E-6AB2-4B59-AA89-3334CA1C34C4\",FSelectType:\"1\",FStartTime:\"2020-06-05 16:00:00\",FEndTime:\"2020-06-11 15:59:59\",FDataTypes:\"42,46\",FDateType:\"0\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C#:**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryMessageAlarmList\",FTokenID:\"3acef045-d302-

```

```

4032-b40a-d9ee6c1519cd\",FGUIDs:\\"D128C716-9D21-4FD2-AC40-EF18A0724AF3,D118
407E-6AB2-4B59-AA89-3334CA1C34C4\",FSelectType:\\"1\",FStartTime:\\"2020-06-0
5 16:00:00\",FEndTime:\\"2020-06-11 15:59:59\",FDataTypes:\\"42,46\",FDateTyp
e:\\"0\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
//返回的是json字符串
return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Report'
data = {
'FAction': 'QueryMessageAlarmList',
'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
'FGUIDs': 'D128C716-9D21-4FD2-AC40-EF18A0724AF3,D118407E-6AB2-4B59-AA8
9-3334CA1C34C4',
'FSelectType':1,
'FStartTime':'2020-08-05 16:00:00',
'FEndTime':'2020-08-11 15:59:59',
'FDataTypes':'42,46',
'FDateType':0
}
data = parse.urlencode(data).encode('utf-8')
headers = {
'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()

```

```
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

### Node.js:

# 事件消息报表

## 简要描述：

- 获取事件消息报表

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2022-02-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名(QueryMessageEventList)
FGUIDs	是	string	车辆/设备唯一标识；多个用英文半角逗号隔开
FSelectType	是	int	1：按车辆查；0：按终端查
FStartTime	是	datetime	开始时间 (UTC时间)

FEndTime	是	datetime	结束时间 (UTC时间)
FDataTypes	是	string	事件类型集合；多个用英文半角逗号隔开 (传空表示获取所有类型的事件)
FDateType	是	int	1: 按接收时间查；0: 按GPS时间查
#### 返回示例:			

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "0d6a1334-8eee-4b9b-822e-0042f4cedfd6",
      "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
      "FVehicleName": "44444",
      "FAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
      "FAssetID": "7560717001",
      "FAssetTypeID": 701,
      "FSubAssetID": null,
      "FDataType": 42,
      "FCurrentStatus": 3,
      "FStartTime": "2020-06-06T09:23:45",
      "FStartLongitude": 113.856122,
      "FStartLatitude": 22.585537,
      "FEndLongitude": 113.856122,
      "FEndLatitude": 22.585537,
      "FEndTime": "2020-06-06T09:23:45",
      "FStartSpeed": 0,
      "FEndSpeed": 0,
      "FKeepSeconds": 0,
      "FStartMileage": 38,
      "FEndMileage": 38,
      "FKeepDistance": 0,
      "FFenceName": "",
      "FRelateGUID": null,
      "FWaybillInfo": "{\"FGUID\": \"258dbdb5-1c01-495c-98f5-6165e9fa8b4c\", \"FWaybill\": \"\", \"FFenceName\": \"\"}"
    },
    {
      "FGUID": "cd2ec92b-6046-47c8-a9f4-00ebf6bde5a8",
```

```
"FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
"FVehicleName": "44444",
"FAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
"FAssetID": "7560717001",
"FAssetTypeID": 701,
"FSubAssetID": null,
"FDataType": 42,
"FCurrentStatus": 3,
"FStartTime": "2020-06-06T05:21:42",
"FStartLongitude": 113.856103,
"FStartLatitude": 22.585535,
"FEndLongitude": 113.856103,
"FEndLatitude": 22.585535,
"FEndTime": "2020-06-06T05:21:42",
"FStartSpeed": 0,
"FEndSpeed": 0,
"FKeepSeconds": 0,
"FStartMileage": 23,
"FEndMileage": 23,
"FKeepDistance": 0,
"FFenceName": "",
"FRelateGUID": null,
"FWaybillInfo": "{\"FGUID\":\"258dbdb5-1c01-495c-98f5-6165e9fa8
b4c\", \"FWaybill\":\"\", \"FFenceName\":\"\"}"}
},
{
  "FGUID": "1de8e107-41d8-45a7-b67e-00fd7ac6b6d7",
  "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
  "FVehicleName": "44444",
  "FAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
  "FAssetID": "7560717001",
  "FAssetTypeID": 701,
  "FSubAssetID": null,
  "FDataType": 42,
  "FCurrentStatus": 3,
  "FStartTime": "2020-06-05T17:20:24",
  "FStartLongitude": 113.856107,
  "FStartLatitude": 22.585558,
  "FEndLongitude": 113.856107,
  "FEndLatitude": 22.585558,
  "FEndTime": "2020-06-05T17:20:24",
  "FStartSpeed": 0,
  "FEndSpeed": 0,
  "FKeepSeconds": 0,
  "FStartMileage": 22,
  "FEndMileage": 22,
```

```

        "FKeepDistance": 0,
        "FFenceName": "",
        "FRelateGUID": null,
        "FWaybillInfo": "{\"FGUID\":\"258dbdb5-1c01-495c-98f5-6165e9fa8
b4c\", \"FWaybill\": \"\", \"FFenceName\": \"\"}"
    },
    {
        "FGUID": "52824179-d573-430c-b87f-0114cc62c032",
        "FVehicleGUID": "d128c716-9d21-4fd2-ac40-ef18a0724af3",
        "FVehicleName": "44444",
        "FAssetGUID": "488c91fb-9a41-4bd8-a433-b93ad58aae04",
        "FAssetID": "7560717001",
        "FAssetTypeID": 701,
        "FSubAssetID": null,
        "FDataType": 42,
        "FCurrentStatus": 3,
        "FStartTime": "2020-06-05T21:20:48",
        "FStartLongitude": 113.856132,
        "FStartLatitude": 22.585435,
        "FEndLongitude": 113.856132,
        "FEndLatitude": 22.585435,
        "FEndTime": "2020-06-05T21:20:48",
        "FStartSpeed": 0,
        "FEndSpeed": 0,
        "FKeepSeconds": 0,
        "FStartMileage": 22,
        "FEndMileage": 22,
        "FKeepDistance": 0,
        "FFenceName": "海谷",
        "FRelateGUID": null,
        "FWaybillInfo": "{\"FGUID\":\"258dbdb5-1c01-495c-98f5-6165e9fa8
b4c\", \"FWaybill\": \"\", \"FFenceName\": \"海谷\"}"
    }
]
}

```

**错误时返回:**

```

{
    "Result": 102,
    "Message": "Action is error",
    "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FGUID	String	事件数据唯一标识
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车牌号
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FSubAssetID	String	从机ID
FDataType	Int	事件类型
FCnName	String	事件类型中文名称
FEnName	String	事件类型英文名称
FCurrentStatus	Int	当前状态（1：执行中；2：已完成；3：单点事件（瞬间的动作））
FStartTime	DateTime	开始GPS时间
FStartLongitude	Decimal	开始经度
FStartLatitude	Decimal	开始纬度
FEndLongitude	Decimal	结束经度
FEndLatitude	Decimal	结束纬度
FEndTime	DateTime	结束GPS时间
FStartSpeed	Int	开始速度（不显示）
FEndSpeed	Int	结束速度（不显示）
FKeepSeconds	Int	持续时长（单位s）
FStartMileage	Int	开始里程
FEndMileage	Int	结束里程
FKeepDistance	Int	持续距离

FCreateTime	DateTime	接收时间
FStartCell	String	开始小区码
FFenceName	String	事件绑定的围栏名称
FRelateGUID	String	事件关联的运单唯一标识
FWaybillInfo	String	事件绑定的运单信息 ,Json字符串

## FWaybillInfo 参数说明:

参数名	类型	说明
FGUID	String	运单数据唯一标识
FWaybillNo	String	运单号
FFenceName	String	运单事件绑定的围栏名称

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryMessageEventList\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUIDs: \"D128C716-9D21-4FD2-AC40-EF18A0724AF3,D118407E-6AB2-4B59-AA89-3334CA1C34C4\", FSelectType: \"1\", FStartTime: \"2020-06-05 16:00:00\", FEndTime: \"2020-06-11 15:59:59\", FDataTypes: \"42,46\", FDateType: \"0\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
```

```

conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction: \"QueryMessageEventList\", FTokenID: \"3acef045-d302-
4032-b40a-d9ee6c1519cd\", FGUIDs: \"D128C716-9D21-4FD2-AC40-EF18A0724AF3,D118
407E-6AB2-4B59-AA89-3334CA1C34C4\", FSelectType: \"1\", FStartTime: \"2020-06-0
5 16:00:00\", FEndTime: \"2020-06-11 15:59:59\", FDataTypes: \"42,46\", FDateTyp
e: \"0\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

```

```

using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Report'
data = {
    'FAction': 'QueryMessageEventList',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': 'D128C716-9D21-4FD2-AC40-EF18A0724AF3,D118407E-6AB2-4B59-AA8
9-3334CA1C34C4',
    'FSelectType':1,
    'FStartTime':'2020-08-05 16:00:00',
    'FEndTime':'2020-08-11 15:59:59',
    'FDataTypes':'42,46',
    'FDateType':0
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

**Node.js:**

# 温湿度曲线报表（新）

## 简要描述：

- 获取温湿度曲线报表（新）

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2022-02-18	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportTemperatureTrackNew)
FGUID	是	string	车辆/设备唯一标识
FSelectType	是	int	1: 按车辆查; 0: 按终端查
FStartTime	是	datetime	开始时间 (UTC时间)

FEndTime	是	datetime	结束时间 (UTC时间)
#### 返回示例:			

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T08:48:40",
      "FDateTime": "2022-02-17T08:48:40",
      "FTemperature": "31",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T08:51:56",
      "FDateTime": "2022-02-17T08:51:56",
      "FTemperature": "32",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T09:21:58",
      "FDateTime": "2022-02-17T09:21:58",
      "FTemperature": "31",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T09:52:00",
      "FDateTime": "2022-02-17T09:52:00",
      "FTemperature": "31",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T10:21:46",
```

```

    "FDateTime": "2022-02-17T10:21:46",
    "FTemperature": "29",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T10:49:46",
    "FDateTime": "2022-02-17T10:49:46",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T11:20:05",
    "FDateTime": "2022-02-17T11:20:05",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T11:50:10",
    "FDateTime": "2022-02-17T11:50:10",
    "FTemperature": "24",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T12:20:26",
    "FDateTime": "2022-02-17T12:20:26",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T12:50:12",
    "FDateTime": "2022-02-17T12:50:12",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {

```

```

    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T13:20:07",
    "FDateTime": "2022-02-17T13:20:07",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T13:50:09",
    "FDateTime": "2022-02-17T13:50:09",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T14:20:07",
    "FDateTime": "2022-02-17T14:20:07",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T14:50:11",
    "FDateTime": "2022-02-17T14:50:11",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T15:20:01",
    "FDateTime": "2022-02-17T15:20:01",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T15:50:10",
    "FDateTime": "2022-02-17T15:50:10",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  }

```

```

    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T16:20:22",
      "FDateTime": "2022-02-17T16:20:22",
      "FTemperature": "22",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T16:50:08",
      "FDateTime": "2022-02-17T16:50:08",
      "FTemperature": "22",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T17:20:11",
      "FDateTime": "2022-02-17T17:20:11",
      "FTemperature": "22",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T17:50:37",
      "FDateTime": "2022-02-17T17:50:37",
      "FTemperature": "23",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T18:20:40",
      "FDateTime": "2022-02-17T18:20:40",
      "FTemperature": "23",
      "FHumidity": 0,
      "FIndex": 1
    },
    {
      "FSubAssetID": "Temperature",
      "FGPSTime": "2022-02-17T18:50:33",
      "FDateTime": "2022-02-17T18:50:33",
      "FTemperature": "23",
  
```

```

    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T19:20:33",
    "FDateTime": "2022-02-17T19:20:33",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T19:50:36",
    "FDateTime": "2022-02-17T19:50:36",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T20:20:36",
    "FDateTime": "2022-02-17T20:20:36",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T20:50:44",
    "FDateTime": "2022-02-17T20:50:44",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T21:20:38",
    "FDateTime": "2022-02-17T21:20:38",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T21:50:30",

```

```

    "FDateTime": "2022-02-17T21:50:30",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T22:20:40",
    "FDateTime": "2022-02-17T22:20:40",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T22:50:32",
    "FDateTime": "2022-02-17T22:50:32",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T23:20:35",
    "FDateTime": "2022-02-17T23:20:35",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-17T23:50:38",
    "FDateTime": "2022-02-17T23:50:38",
    "FTemperature": "22",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T00:20:43",
    "FDateTime": "2022-02-18T00:20:43",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {

```

```

    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T00:50:58",
    "FDateTime": "2022-02-18T00:50:58",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T01:20:48",
    "FDateTime": "2022-02-18T01:20:48",
    "FTemperature": "23",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T01:50:53",
    "FDateTime": "2022-02-18T01:50:53",
    "FTemperature": "26",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T02:21:05",
    "FDateTime": "2022-02-18T02:21:05",
    "FTemperature": "30",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T02:51:01",
    "FDateTime": "2022-02-18T02:51:01",
    "FTemperature": "36",
    "FHumidity": 0,
    "FIndex": 1
  },
  {
    "FSubAssetID": "Temperature",
    "FGPSTime": "2022-02-18T03:21:38",
    "FDateTime": "2022-02-18T03:21:38",
    "FTemperature": "29",
    "FHumidity": 0,
    "FIndex": 1
  }

```

```
    },  
    {  
      "FSubAssetID": "Temperature",  
      "FGPSTime": "2022-02-18T03:51:47",  
      "FDateTime": "2022-02-18T03:51:47",  
      "FTemperature": "27",  
      "FHumidity": 0,  
      "FIndex": 1  
    },  
    {  
      "FSubAssetID": "Temperature",  
      "FGPSTime": "2022-02-18T04:21:07",  
      "FDateTime": "2022-02-18T04:21:07",  
      "FTemperature": "26",  
      "FHumidity": 0,  
      "FIndex": 1  
    },  
    {  
      "FSubAssetID": "Temperature",  
      "FGPSTime": "2022-02-18T04:51:15",  
      "FDateTime": "2022-02-18T04:51:15",  
      "FTemperature": "26",  
      "FHumidity": 0,  
      "FIndex": 1  
    },  
    {  
      "FSubAssetID": "Temperature",  
      "FGPSTime": "2022-02-18T05:21:08",  
      "FDateTime": "2022-02-18T05:21:08",  
      "FTemperature": "25",  
      "FHumidity": 0,  
      "FIndex": 1  
    },  
    {  
      "FSubAssetID": "Temperature",  
      "FGPSTime": "2022-02-18T05:51:20",  
      "FDateTime": "2022-02-18T05:51:20",  
      "FTemperature": "25",  
      "FHumidity": 0,  
      "FIndex": 1  
    }  
  ]  
}
```

#### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
FSubAssetID	String	设备号（作为外设子锁时显示设备号，作为一体机，则显示OutSide、InSide、Temperature）
FGPSTime	DateTime	GPS时间
FDateTime	DateTime	接收数据时间（作为外设子锁时显示接收数据时间，作为一体机，则为FGPSTime）
FTemperature	string	温度值
FHumidity	Int	湿度值
FIndex	Int	传感器序号

## 备注:

- 更多返回错误代码如下：
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"QueryReportTemperatureTrackNew\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FGUID: \"EABD84FC-0228-4836-A0A2-3404EC074A13\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:00:00\", FEndTime: \"2020-08-25 15:59:59\"}";
URL realUrl = new URL(url);
```

```

// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"QueryReportTemperatureTrackNew\", FTokenID: \"3acef
045-d302-4032-b40a-d9ee6c1519cd\", FGUID: \"EABD84FC-0228-4836-A0A2-3404EC074
A13\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:00:00\", FEndTime: \"2020-
08-25 15:59:59\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;

```

```

request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Report'
data = {
    'FAction': 'QueryReportTemperatureTrackNew',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUID': 'EABD84FC-0228-4836-A0A2-3404EC074A13',
    'FSelectType': 1,
    'FStartTime': '2020-08-18 16:00:00',
    'FEndTime': '2020-08-24 15:59:59',
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

**Node.js:**

# 油耗报表

# 加油报告

## 简要描述：

- 加油报告

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-09-07	2023-09-07

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportFuelRefuel )
FVehicleGUIDs	是	string	车辆/设备唯一标识列表 (例: "a524d10f-c543-485c-ad92-381eeb739aaf,a524d10f-c543-485c-ad92-381eeb739aaf")

FStartTime	是	string	开始时间 例("2023-07-08 16:00:00")
FEndTime	是	string	结束时间 例("2023-08-07 15:59:59")

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FTotalCount": 4
      }
    ],
    "Table1": [
      {
        "FVehicleGUID": "a524d10f-c543-485c-ad92-381eeb739aaf",
        "FVehicleName": "14161647151",
        "FAgentName": "久通中试部",
        "FStartTime": "2022-04-27T09:01:03",
        "FEndTime": "2022-04-27T09:06:05",
        "FStartLiter": 70,
        "FEndLiter": 448,
        "FLatitude": 22.58,
        "FLongitude": 113.9174,
        "FRefuel": 378,
        "RowNo": 1
      }
    ]
  }
}
```

### 错误时返回:

```
//表示传入的车辆或设备为空, 应该是数据格式不对
{
  "Result": 102,
  "Message": "FGUID is null",
  "FObject": []
}
//表示有异常报错
```

```
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 返回参数说明:

参数名	类型	说明
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车辆ID
FAgentName	String	公司名称
FStartTime	String	开始时间
FEndTime	String	结束时间
FStartLiter	Double	开始升数
FEndLiter	Double	结束升数
FLongitude	Double	当前经度
FLatitude	Double	当前纬度
FRefuel	Double	加油升数

## 请求示例:

Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
```

```

//请求参数 ,json格式参数,建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportFuelRefuel\", \n\t\"FVehicleGU\n\tIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \n\t\"FStartTime\": \"2022-\n\t04-05 16:00:00\", \n\t\"FEndTime\": \"2022-05-06 15:59:59\", \n\t\"FPageSize\"\n\t\": 50, \n\t\"FPageIndex\": 1, \n\t\"FTokenID\": \"3acef045-d302-4032-b40a-d9e\n\te6c1519cd\", \n\t\"FTimeDifferent\": 28800, \n\t\"FLanguage\": \"1\"\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W\n\tindows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get\n\tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportFuelRefuel\", \n\t\"FVehicleGU\n\tIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \n\t\"FStartTime\": \"2022-\n\t04-05 16:00:00\", \n\t\"FEndTime\": \"2022-05-06 15:59:59\", \n\t\"FPageSize\"\n\t\": 50, \n\t\"FPageIndex\": 1, \n\t\"FTokenID\": \"3acef045-d302-4032-b40a-d9e\n\te6c1519cd\", \n\t\"FTimeDifferent\": 28800, \n\t\"FLanguage\": \"1\"\n}";
Encoding encoding = Encoding.UTF8;

```

```

HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'

data = '{ "FAction": "QueryReportFuelRefuel", "FVehicleGUIDs": "a524d10f-c543-485c-ad92-381eeb739aaf", "FStartTime": "2022-04-05 16:00:00", "FEndTime": "2022-05-06 15:59:59", "FPageSize": 50, "FPageIndex": 1, "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd", "FTimeDifferent": 28800, "FLanguage": "1" }'

data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

# 异常油耗

## 简要描述：

- 异常油耗

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-09-07	2023-09-07

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportFuelImproper )
FVehicleGUIDs	是	string	车辆/设备唯一标识列表 (例: "a524d10f-c543-485c-ad92-381eeb739aaf,a524d10f-c543-485c-ad92-381eeb739aaf")

FStartTime	是	string	开始时间(UTC时间) 例("2023-07-08 16:00:00")
FEndTime	是	string	结束时间(UTC时间) 例("2023-08-07 15:59:59")

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FTotalCount": 5
      }
    ],
    "Table1": [
      {
        "FVehicleGUID": "a524d10f-c543-485c-ad92-381eeb739aaf",
        "FVehicleName": "14161647151",
        "FAgentName": "久通中试部",
        "FStartTime": "2022-04-27T09:16:05",
        "FEndTime": "2022-04-27T09:20:13",
        "FStartLiter": 448,
        "FEndLiter": 70,
        "FLatitude": 22.58,
        "FLongitude": 113.9174,
        "FImproper": 378,
        "RowNo": 1
      }
    ]
  }
}
```

### 错误时返回:

```
//表示传入的车辆或设备为空, 应该是数据格式不对
{
  "Result": 102,
  "Message": "FGUID is null",
  "FObject": []
}
//表示有异常报错
```

```
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 返回参数说明:

参数名	类型	说明
FVehicleGUID	String	车辆唯一标识
FVehicleName	String	车辆ID
FAgentName	String	公司名称
FStartTime	String	开始时间
FEndTime	String	结束时间
FStartLiter	Double	开始升数
FEndLiter	Double	结束升数
FLongitude	Double	当前经度
FLatitude	Double	当前纬度
FImproper	Double	异常油耗

## 请求示例:

Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
```

```

//请求参数 ,json格式参数 ,建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportFuelImproper\", \n\t\"FVehicle
GUIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \n\t\"FStartTime\": \"202
2-04-05 16:00:00\", \n\t\"FEndTime\": \"2022-05-06 15:59:59\", \n\t\"FPageSiz
e\": 50, \n\t\"FPageIndex\": 1, \n\t\"FTokenID\": \"3acef045-d302-4032-b40a-d
9ee6c1519cd\", \n\t\"FTimeDifferent\": 28800, \n\t\"FLanguage\": \"1\"\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportFuelImproper\", \n\t\"FVehicle
GUIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \n\t\"FStartTime\": \"202
2-04-05 16:00:00\", \n\t\"FEndTime\": \"2022-05-06 15:59:59\", \n\t\"FPageSiz
e\": 50, \n\t\"FPageIndex\": 1, \n\t\"FTokenID\": \"3acef045-d302-4032-b40a-d
9ee6c1519cd\", \n\t\"FTimeDifferent\": 28800, \n\t\"FLanguage\": \"1\"\n}";
Encoding encoding = Encoding.UTF8;

```

```

HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'

data = '{ "FAction": "QueryReportFuelImproper", "FVehicleGUIDs": "a524d10f-c543-485c-ad92-381eeb739aaf", "FStartTime": "2022-04-05 16:00:00", "FEndTime": "2022-05-06 15:59:59", "FPageSize": 50, "FPageIndex": 1, "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd", "FTimeDifferent": 28800, "FLanguage": "1" }'

data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

# 油耗分析

## 简要描述：

- 油耗分析

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-09-18	2023-09-18

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportFuelTrack )
FVehicleGUID	是	string	车辆唯一标识
FStartTime	是	string	开始时间(UTC时间) 例("2023-07-08 16:00:00")
FEndTime	是	string	结束时间(UTC时间) 例( "2023-08-07 15:59:59")

## 返回示例:

### 正确时返回:

```
{
  "Result":200,
  "Message":"check token success",
  "FObject":{
    "Table":[
      {
        "Speed":66,
        "Mil":4390,
        "GT":"2023-09-17T16:02:03",
        "FV1":455,
        "FV2":457,
        "FV3":0,
        "FC":912
      }
    ],
    "Table1":[
      {
        "FSensorIndex":1,
        "FFuelSize":473,
        "FSensorName":""
      }
    ]
  }
}
```

### 错误时返回:

```
//表示传入的车辆为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FVehicleGUID is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 返回参数说明:

Table

参数名	类型	说明
Speed	int	速度
Mil	int	里程
GT	String	GPS时间
FV1	int	油位1
FV2	int	油位2
FV3	int	油位3
FC	int	总油位

Table1

参数名	类型	说明
FSensorIndex	int	传感器索引
FFuelSize	int	油箱容积
FSensorName	string	传感器接入的端口名称

## 请求示例:

Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
```

```

//请求参数 ,json格式参数, 建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportFuelTrack\", \n\t\"FVehicleGUID\": \"a4b098c1-66e4-416d-8cf4-2d1cf88b65ab\", \n\t\"FStartTime\": \"2023-09-17 16:00:00\", \n\t\"FEndTime\": \"2023-09-18 15:59:59\", \n\t\"FTokenID\": \"bbd9ab6d-55df-4aa8-96d5-07595519e76d\"\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportFuelTrack\", \n\t\"FVehicleGUID\": \"a4b098c1-66e4-416d-8cf4-2d1cf88b65ab\", \n\t\"FStartTime\": \"2023-09-17 16:00:00\", \n\t\"FEndTime\": \"2023-09-18 15:59:59\", \n\t\"FTokenID\": \"bbd9ab6d-55df-4aa8-96d5-07595519e76d\"\n}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get

```

```

request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'

data = '{ "FAction": "QueryReportFuelTrack", "FVehicleGUID": "a4b098c1-66e4-416d-8cf4-2d1cf88b65ab", "FStartTime": "2023-09-17 16:00:00", "FEndTime": "2023-09-18 15:59:59", "FTokenID": "bbd9ab6d-55df-4aa8-96d5-07595519e76d" }'

data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

# 查看报警/事件/开关锁报表的附件

## 简要描述：

- 获取报警/事件/开关锁的附件

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-08-20	2020-08-20

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名(QueryMessageFiles)
FGUID	是	string	报警/事件/开关锁报表的唯一标识

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FFileName": "80025080021_1762438053_01.mp4",
      "FFileType": 2,
      "FReqURL": "https://hcloud.assetscontrols.com:8068//LockFiles//80025080021_1762438053/80025080021_1762438053_11.mp4",
      "FCreateTime": null,
      "FAssetTypeID": 800
    }
  ]
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
FFileName	String	文件名
FFileType	Int	文件类型 1: 图片 2: 音频 3: 视频 -1: 无效
FReqURL	String	文件url链接地址
FCreateTime	String	创建时间,可忽略
FAssetTypeID	Int	设备类型,可忽略

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryMessageFiles\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FGUID:\"D128C716-9D21-4FD2-AC40-EF18A0724AF3\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```
//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Report";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryMessageFiles\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FGUID:\"D128C716-9D21-4FD2-AC40-EF18A0724AF3\"}";
```

```
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

#### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Report'
data = {
    'FAction': 'QueryMessageFiles',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUID': 'D128C716-9D21-4FD2-AC40-EF18A0724AF3',
    'FDateType': 0
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

#### Node.js:

查看报警/事件/开关锁报表的附件

# 日里程报告

## 简要描述：

- 日里程报告

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2025-02-25	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Report>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Report>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportDayMileageList )
FVehicleGUIDs	是	string	车辆唯一标识列表 (例: "a524d10f-c543-485c-ad92-381eeb739aaf,a524d10f-c543-485c-ad92-381eeb739aaf")

FStartTime	是	string	开始时间,UTC时间 例("2023-07-08 16:00:00")
FEndTime	是	string	结束时间,UTC时间 例("2023-08-07 15:59:59")
FPageSize	是	int	每页数量,默认20
FPageIndex	是	int	页数

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FTotalCount": 2
      }
    ],
    "Table1": [
      {
        "FGUID": "ce716b7c-c778-4a48-b4f8-008479b79a94",
        "FVehicleName": "14141414",
        "FDate": "2026-02-25",
        "FTotalMileage": 0,
        "FStartMileage": 0,
        "FEndMileage": 0,
        "FAssetID": "2525252",
        "RowNo": 1
      },
      {
        "FGUID": "f767264b-6178-482c-b566-1250da6fe6b2",
        "FVehicleName": "854565265246",
        "FDate": "2026-02-25",
        "FTotalMileage": 0,
        "FStartMileage": 0,
        "FEndMileage": 0,
        "FAssetID": "15228655",
        "RowNo": 2
      }
    ]
  }
}
```

## 错误时返回:

```
//表示传入的车辆或设备为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FVehicleGUIDs is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 返回参数说明:

参数名	类型	说明
FGUID	String	数据唯一标识
FVehicleName	String	车辆ID
FDate	String	日期
FTotalMileage	String	总里程,单位km
FStartMileage	int	开始里程,单位km
FEndMileage	int	结束里程,单位km
FAssetID	String	设备号

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportDayMileageList\", \n\t\"FVehicleGUIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \n\t\"FStartTime\": \"2022-04-05 16:00:00\", \n\t\"FEndTime\": \"2022-05-06 15:59:59\", \n\t\"FPageSize\": 50, \n\t\"FPageIndex\": 1, \n\t\"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\" }";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
String body = "{\n\t\"FAction\": \"QueryReportDayMileageList\", \n\t\"FVehicleGUIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \n\t\"FStartTime\": \"2022-04-05 16:00:00\", \n\t\"FEndTime\": \"2022-05-06 15:59:59\", \n\t\"FPageSize\"

```

```

ize\": 50,\\n\\t\\\"FPageIndex\\\": 1,\\n\\t\\\"FTokenID\\\": \\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\\" }\";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = \"post\";
request.Accept = \"*/\";
request.ContentType = \"application/json\";
request.UserAgent = \"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)\";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
//返回的是json字符串
return reader.ReadToEnd();
}

```

#### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'

data = '{ \"FAction\": \"QueryReportDayMileageList\", \"FVehicleGUIDs\": \"a524d10f-c543-485c-ad92-381eeb739aaf\", \"FStartTime\": \"2022-04-05 16:00:00\", \"FEndTime\": \"2022-05-06 15:59:59\", \"FPageSize\": 50, \"FPageIndex\": 1, \"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\" }'

data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```



# 远程指令集

# 子锁开锁

## 简要描述：

- 子锁开锁

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-09-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (SubAssetOpenLock)
FAssetGUID	是	string	设备唯一标识
FSubAssetGUID	是	string	子锁唯一标识
FLanguage	是	Int	0：英文； 1：中文

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

{
  "Result": 108,
  "Message": "ins send error",
  "FObject": []
}

{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误
- 108: 开锁失败
- 113: 未在开锁区域, 禁止开锁
- 115: 未定位, 无法进行开锁
- 110: 没有开锁权限
- 108: 开锁指令发送异常
- 106: 设备不属于该用户
- 105: 系统异常
- 123: 设备不支持
- 111: 设备离线或休眠

- 138:指令执行超时,设备已休眠或者离线

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\n\t\"FAction\": \"SubAssetOpenLock\",\n\t\"FAssetGUID\": \"2d3d4041-dc5f-4a50-8c6a-8aac15835bf0\",\n\t\"FSubAssetGUID\": \"8846d981-a61f-455d-94fc-33c64416ff7a\",\n\t\"FTokenID\": \"3acef045-d302-4032-b40a-d9ee6c1519cd\",\n\t\"FTimeDifferent\": 28800,\n\t\"FLanguage\": \"1\"\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```
//请求路径
```

```

string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\n\t\"FAction\": \"SubAssetOpenLock\", \n\t\"FAssetGUID\": \"\n\t\"2d3d4041-dc5f-4a50-8c6a-8aac15835bf0\", \n\t\"FSubAssetGUID\": \"\n\t\"8846d981-a61f-455d-94fc-33c64416ff7a\", \n\t\"FTokenID\": \"\n\t\"3acef045-d302-4032-b40a-d9ee6c1519cd\", \n\t\"FTimeDifferent\": 28800, \n\t\"FLanguage\": \"\n\t\"1\"\n}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = '{ "FAction": "SubAssetOpenLock", "FAssetGUID": "2d3d4041-dc5f-4a50-8c6a-8aac15835bf0", "FSubAssetGUID": "8846d981-a61f-455d-94fc-33c64416ff7a", "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd", "FTimeDifferent": 28800, "FLanguage": "1" }'`2023-09-07 16:02:52 星期四`
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```



# 子锁列表

## 简要描述：

- 子锁列表

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-09-07	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QuerySubAssetInfoByFAssetGUID)
FAssetGUID	是	string	设备唯一标识
FLanguage	是	Int	0：英文； 1：中文

## 返回示例：

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "8846d981-a61f-455d-94fc-33c64416ff7a",
      "FAssetGUID": "2d3d4041-dc5f-4a50-8c6a-8aac15835bf0",
      "FAssetID": "8052400008",
      "FAssetTypeID": 3701,
      "FSubAssetName": "JT709",
      "FSubAssetID": "E0171E0366",
      "FIndex": 9,
      "FDateTime": "2023-09-07T07:03:05Z",
      "FLockStatus": 0,
      "FLockRope": 0,
      "FOpenIns": 0,
      "FPower": 93,
      "FBle": 1,
      "FSensorType": 4,
      "FStatusJson": "{\"lockRope\":0,\"gateway\":0}",
      "FEvent": 6
    }
  ]
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
FGUID	string	子锁唯一 标识
FAssetGUID	String	主机唯一标识
FAssetID	String	主机ID

FAssetTypeID	Int	主机类型标识
FSubAssetName	String	子锁名称
FSubAssetID	String	子锁设备ID
FIndex	Int	配置的从机序号
FDateTime	String	最后更新时间
FLockStatus	Int	开锁状态(0:关 1:开)
FLockRope	Int	锁绳(0:关 1:开)
FOpenIns	Int	开锁指令执行状态(0:关 1:开)
FPower	Int	电量
FBle	Int	是否支持蓝牙 (1: 支持 0: 不支持)
FSensorType	Int	从机类型 (1: JT126; 4: JT709;5:JT801;6:JT802)
FStatusJson	String	拓展信息
FEvent	Int	事件类型

### FStatusJson

参数名	类型	说明
lockMotor	Int	电机状态(0: 电机关, 1: 电机开)
lockKnob	Int	旋钮状态(0: 关闭, 1: 打开)
lockValve	Int	阀门状态(0:关闭, 1打开)
structuralDisassembly	Int	结构防拆卸状态(0:无拆卸, 1拆卸)
bottomDisassembly	Int	底板防拆卸状态(0:无拆卸, 1拆卸)
emergencyKey	Int	应急钥匙状态(0: 封存, 1: 启用)

### FEvent

状态数值	描述

0	关锁事件
1	蓝牙开锁事件
2	拆后盖事件
3	远程开锁事件
4	锁剪断报警
5	按键唤醒事件
6	心跳包事件
7	充电唤醒事件
8, 20	拔出锁绳事件
9	RFID开锁事件
10	刷非法卡报警
14	从机信号丢失事件
15	阀门关闭事件
16	阀门打开事件
17	低电量报警
18	防拆卸报警
19	电子仓拆卸事件
21	锁绳插入事件
22	蓝牙连接唤醒事件
23	应急仓开报警
24	应急仓关报警
25	异常阀门打开报警
26	锁销关闭事件
27	锁销开启事件

30	NFC触发
-1	未知事件

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";  
//请求路径  
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";  
//请求参数 ,json格式参数, 建议用对象传入  
String body = "{\n\t\"FAction\": \"QuerySubAssetInfoByFAAssetGUID\",\n\t\"FA  
ssetGUID\": \"2d3d4041-dc5f-4a50-8c6a-8aac15835bf0\",\n\t\"FTokenID\": \"3a  
cef045-d302-4032-b40a-d9ee6c1519cd\",\n\t\"FTimeDifferent\": 28800,\n\t\"FL  
anguage\": \"1\"\n}";  
URL realUrl = new URL(url);  
// 设置通用请求的属性  
URLConnection conn = realUrl.openConnection();  
conn.setRequestProperty("accept", "*/*");  
conn.setRequestProperty("connection", "keep-Alive");  
conn.setRequestProperty("Content-Type", "application/json");  
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W  
indows NT 5.1;SV1)");  
conn.setRequestProperty("method", "post");  
// 发送POST请求必须设置如下两行  
conn.setDoOutput(true);  
conn.setDoInput(true);  
PrintWriter pw = new PrintWriter(conn.getOutputStream());  
// 发送请求参数  
pw.print(body);  
// flush输出流的缓冲  
pw.flush();  
// 定义BufferedReader输入流来读取URL的响应  
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge  
tInputStream()));  
// 定义BufferedReader输入流来读取URL的响应  
String line;
```



```
    }  
    req = request.Request(url, headers=headers, data=data)  
    page = request.urlopen(req).read()  
    page = page.decode('utf-8')  
    # json_array = json.loads(page)  
    return page;
```

# 远程指令集查询

## 简要描述：

- 查询设备对应的远程指令集

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2021-12-31	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QuerySystemRemoteInstructions)
FAssetGUID	是	string	设备唯一标识
FLanguage	是	Int	0: 英文; 1: 中文

## 返回示例:

## 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FType": null,
      "FName": "P01-查询终端版本",
      "FInsType": "P01"
    },
    {
      "FType": null,
      "FName": "P03-低电休眠控制",
      "FInsType": "P03"
    },
    {
      "FType": null,
      "FName": "P04-设置/查询数据上传间隔和休眠自动唤醒间隔",
      "FInsType": "P04"
    },
    {
      "FType": null,
      "FName": "P06-设置/查询监控中心IP、端口、APN",
      "FInsType": "P06"
    },
    {
      "FType": null,
      "FName": "P10-设置/查询时差",
      "FInsType": "P10"
    },
    {
      "FType": null,
      "FName": "P11-设置/查询VIP手机号码",
      "FInsType": "P11"
    },
    {
      "FType": null,
      "FName": "P12-设置/查询VIP号码是否允许报警",
      "FInsType": "P12"
    },
    {
      "FType": null,
      "FName": "P13-恢复出厂设置",
      "FInsType": "P13"
    }
  ]
}
```

```
{
  "FType": null,
  "FName": "P14-读取终端的IMEI号",
  "FInsType": "P14"
},
{
  "FType": null,
  "FName": "P15-重启设备",
  "FInsType": "P15"
},
{
  "FType": null,
  "FName": "P22-授时(同步GMT时间)",
  "FInsType": "P22"
},
{
  "FType": null,
  "FName": "P23-设置/取消短信、电话可唤醒工作模式",
  "FInsType": "P23"
},
{
  "FType": null,
  "FName": "P32-主动进入休眠指令",
  "FInsType": "P32"
},
{
  "FType": null,
  "FName": "P37-设置G-sensor参数",
  "FInsType": "P37"
},
{
  "FType": null,
  "FName": "P38-查询/设置开锁报警时间间隔参数",
  "FInsType": "P38"
},
{
  "FType": null,
  "FName": "P40-查询/设置GPRS通道和短消息通道的报警开关",
  "FInsType": "P40"
},
{
  "FType": null,
  "FName": "P54-查询/设置设备追踪模式(设备持续工作)",
  "FInsType": "P54"
},
{
```

```
    "FType": null,
    "FName": "P55-查询/设置剩余电量百分比报警",
    "FInsType": "P55"
  },
  {
    "FType": null,
    "FName": "P56-超速报警速度阈值及时间阈值设置指令",
    "FInsType": "P56"
  },
  {
    "FType": null,
    "FName": "P57-GSM低信号报警阈值设置指令",
    "FInsType": "P57"
  },
  {
    "FType": null,
    "FName": "P61-电量低报警提示阈值设置",
    "FInsType": "P61"
  },
  {
    "FType": null,
    "FName": "P62-查询/设置里程统计相关参数",
    "FInsType": "P62"
  },
  {
    "FType": null,
    "FName": "P63-静态飘移处理功能设置",
    "FInsType": "P63"
  },
  {
    "FType": null,
    "FName": "P64-远程关机",
    "FInsType": "P64"
  },
  {
    "FType": null,
    "FName": "ZZZZ-自定义指令",
    "FInsType": "ZZZZ"
  }
]
}
```

#### 错误时返回:

```
{
```

```

"Result": 102,
"Message": "Action is error",
"FObject": []
}

```

## 返回参数说明:

参数名	类型	说明
FType	String	所属分类（如果为空，则没有分类）
FName	String	指令名称
FInsType	String	指令类型

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QuerySystemRemoteInstructions\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\",FLanguage:\"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);

```

```

conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{FAction:\"QuerySystemRemoteInstructions\",FTokenID:\"3acef0
45-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8
c5f6a3c\",FLanguage:\"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/>";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
  'FAction': 'QuerySystemRemoteInstructions' ,
  'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd' ,
  'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c' ,
  'FLanguage':1
}
data = parse.urlencode(data).encode('utf-8')
headers = {
  'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
               r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
  'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

#### Node.js:

# 下发远程指令

## 简要描述：

- 下发远程指令

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2021-12-31	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (CommandInstruction )
FAssetGUID	是	string	设备唯一标识
FInsType	是	string	指令类型

FOffline	是	string	是否支持离线指令 1: 是 0: 否
P03Ins	是	Object	指令类型对象 (此参数只是举例说明,只有当指令类型是P03时才传,具体指令类型对象,可参考701指令类型对象说明)

[JT701设备类型指令类型对象](#)

[JT707/JT709A/JT705/JT702/JT301设备类型指令类型对象](#)

[GP4000/GP6000/GP6000F设备类型指令类型对象](#)

[JT/T808设备类型指令类型对象](#)

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
无		

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误

- 102: 请求参数错误
- 106: 无此设备
- 131: 未集成此指令
- 108: 指令发送失败

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"CommandInstruction\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\",FInsType:\"P14\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"CommandInstruction\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\",FInsType:\"P14\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    "FAction": "CommandInstruction",
    "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd",
    "FAssetGUID": "8dc24026-649d-4d5b-ae4-f3d24cf56fbc",
    "FOffline": "0",
    "FInsType": "P04",
    "FT_P04Ins": {
        "FOperationType": "1",
        "FUploadInterval": "30",
        "FAwakenInterval": "30",
        "FMotionUploadInterval": "45",
        "FStaticUploadInterval": "60",
        "FJudgingTime": "3"
    }
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.

```

```
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
        'Connection': 'keep-alive'  
    }  
    req = request.Request(url, headers=headers, data=data)  
    page = request.urlopen(req).read()  
    page = page.decode('utf-8')  
    # json_array = json.loads(page)  
    return page;
```

## Node.js:

# 下发自定义指令

## 简要描述：

- 下发自定义远程指令

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2023-03-09	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (CommandDefinedInstruction )
FAssetGUID	是	string	设备唯一标识
FInsContent	是	string	自定义指令内容
FOffline	是	string	是否支持离线指令 1： 是 0： 否

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
无		

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误
- 106: 无此设备
- 131: 未集成此指令
- 108: 指令发送失败

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"CommandDefinedInstruction\",FTokenID:\"3acef045-d
```

```

302-4032-b40a-d9ee6c1519cd\",FAAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6
a3c\",FInsContent:\\"(794203004076,1,001,BASE,1)\\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"CommandDefinedInstruction\",FTokenID:\\"3acef045-d
302-4032-b40a-d9ee6c1519cd\",FAAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6
a3c\",FInsContent:\\"(794203004076,1,001,BASE,1)\\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";

```

```

byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    'FAction': 'CommandDefinedInstruction',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c',
    'FInsContent': '(794203004076,1,001,BASE,1)'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

### Node.js:

# 指令记录查询

## 简要描述：

- 查询该设备的指令记录

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2021-12-31	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryInstructionRecord)
FAssetGUID	是	string	设备唯一标识
FPageSize	否	Int	每页数量(默认20)
FPageIndex	否	Int	页数(默认1)

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FCount": 2
      }
    ],
    "Table1": [
      {
        "FGUID": "27b8c318-af8a-4304-878a-8a5f9c41aad8",
        "FAssetGUID": "9d73e134-84aa-43d1-be9b-77328c5036f6",
        "FAssetID": "231232",
        "FAssetTypeID": 701,
        "FInsType": "P14",
        "FIndex": 0,
        "FInsContent": "(P14)",
        "FBackContent": null,
        "FStatus": 0,
        "FSendTime": "2021-12-31T03:35:06.323",
        "FFinishTime": null,
        "FUserName": "test",
        "RowNo": 1
      },
      {
        "FGUID": "493d64d3-c1aa-4095-936b-6defeab35aad",
        "FAssetGUID": "9d73e134-84aa-43d1-be9b-77328c5036f6",
        "FAssetID": "231232",
        "FAssetTypeID": 701,
        "FInsType": "P14",
        "FIndex": -1,
        "FInsContent": "(P14)",
        "FBackContent": null,
        "FStatus": 0,
        "FSendTime": "2021-12-31T03:04:58.21",
        "FFinishTime": null,
        "FUserName": "test",
        "RowNo": 2
      }
    ]
  }
}
```

}

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
FGUID	String	记录唯一标识
FAssetGUID	String	设备唯一标识
FAssetID	String	设备ID
FAssetTypeID	Int	设备类型
FInsType	String	指令类型
FInsContent	String	指令内容
FBackContent	String	返回内容
FStatus	Int	指令执行状态; 1: 成功; 0: 未成功
FUserName	String	操作人
FSendTime	DateTime	指令发送时间
FFinishTime	DateTime	指令完成时间

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误

**请求示例:**

**Java:**

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryInstructionRecord\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryInstructionRecord\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\"}";

```

```

Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

#### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    'FAction': 'QueryInstructionRecord',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.Loads(page)
return page;

```

#### Node.js:



# 下发在途监管指令(BASE66)

## 简要描述：

- 下发在途监管指令指令

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2021-12-31	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Instruction>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Instruction>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (SendBase66Command )
FAssetGUID	是	string	设备唯一标识
FType	是	string	监管类型 1:启动监管； 2： 结束监管

## 返回示例:

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "terminalStrId": "c5b53001-8755-4269-a9c8-b71ca04c6a0d",
    "terminalNum": "804250900200",
    "dateTime": "2025-12-20T06:58:41.04655Z",
    "dateTimestamp": 1766213921046,
    "response": "(804250900200,6,001,BASE,66,1,0,0)"
  }
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
terminalStrId	String	设备唯一标识号
terminalNum	String	设备号
dateTime	String	指令时间
dateTimestamp	String	指令时间戳
response	String	指令回复内容

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 异常错误
- 102: 请求参数错误
- 106: 无此设备
- 111: 设备不在线

- 131: 未集成此指令
- 108: 指令发送失败
- 138: 指令执行超时(指令发送成功,等待1分钟,设备也未有结果返回)

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction: \"SendBase66Command\", FTokenID: \"3acef045-d302-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\", FT
ype:1}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

### C# :

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Instruction";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"SendBase66Command\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\\",FType:1}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Instruction'
data = {
    "FAction": "CommandInstruction",
    "FTokenID": "3acef045-d302-4032-b40a-d9ee6c1519cd",
    "FAssetGUID": "8dc24026-649d-4d5b-ae4-f3d24cf56fbc",
    "FType": 1,
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```

下发在途监管指令(BASE66)

**Node.js:**

# JT701设备类型指令类型对象

## JT701设备类型指令类型对象:

### FInsType=P01 ( 查询终端版本 )

### FInsType=P03 ( 低电休眠控制 )

#### FT\_P03Ins

```
{
FOperationType: 操作类型 (1: 设置; 0: 查询)
FOpenType: 1是打开此功能, 0关闭此功能
FValue: 设置电量低于多少的时候进入休眠, 默认31%, 可设定范围15%~90%
}
```

### FInsType=P04 ( 设置/查询数据上传间隔和休眠自动唤醒间隔 )

#### FT\_P04Ins

```
{
FOperationType: 操作类型 (1: 设置; 0: 查询)
FUplLoadInterval: 数据上传间隔, 单位秒钟, 默认30秒, 取值范围5-600
FAwakenInterval: 休眠自动唤醒间隔, 单位分钟, 默认30分钟, 取值范围30-1440
FMotionUpLoadInterval: 上锁状态, 运动时, 数据上传间, 单位秒, 默认45秒, 取值范围30-900
FStaticUpLoadInterval: 上锁状态, 静止时, 数据上传间隔, 单位秒, 默认60秒, 取值范围30-3600
FJudgingTime: 运动到静止判断时间默认3分钟, 单位分钟, 取值0~30
}
```

### FInsType=P06 ( 设置/查询监控中心IP、端口、APN )

#### FT\_P06Ins:

```
{
FOperationType: 操作类型 (0: 卡1查询; 1: 卡1设置; 2: 卡2查询; 3: 卡2设置)
FIP: 监控中心的IP地址, 必须为合法IP地址; 为监控中心域名
FPort: 监控中心端口地址, 最大65530
FCMNET: 接入点名称(最长50个字节)
FAPN: APN用户名(最长50个字节)
FPassword: APN密码(最长50个字节)
}
```

### FInsType=P10 ( 设置/查询时差 )

#### FT\_P10Ins:

```
{
FOperationType: 操作类型 (1: 设置; 0: 查询; )
FTimeDifference: 时差值,以分钟为单位.如北京时间与标准时时差为8小时,即为480分钟, 取值范围-1260-1360, 默认0
}
```

**FInsType=P11 ( 设置/查询VIP手机号码 )****FT\_P11Ins:**

{

FOperationType: 操作类型 (1: 设置; 0: 查询) 此处查询时候需要带上参数FIndex, 表示查询第一个VIP号码

FIndex: VIP手机号码索引,取值为1-5,允许有五组VIP手机号码

FPhone: 手机号码,不能超过15位数字, 前面需加国际区号, 中国为86或者+86

}

**FInsType=P12 ( 设置/查询VIP号码是否允许报警 )****FT\_P12Ins:**

{

FOperationType: 操作类型 (1: 设置; 0: 查询)

FVIP1: 设定的值为1表示允许对应的VIP号码报警, 0表示不允许此VIP号码报警

FVIP2: 设定的值为1表示允许对应的VIP号码报警, 0表示不允许此VIP号码报警

FVIP3: 设定的值为1表示允许对应的VIP号码报警, 0表示不允许此VIP号码报警

FVIP4: 设定的值为1表示允许对应的VIP号码报警, 0表示不允许此VIP号码报警

FVIP5: 设定的值为1表示允许对应的VIP号码报警, 0表示不允许此VIP号码报警

}

**FInsType=P13 ( 恢复出厂设置 )****FInsType=P14 ( 读取终端的IMEI号 )****FInsType=P15 ( 重启设备 )****FInsType=P22 ( 授时 )****FT\_P22Ins:**

{

FTime: 日月年时分秒, 该时间为格林威治时间 (如: 180715164328)

}

**FInsType=P23 ( 设置/取消短信、电话可唤醒工作模式 )****FT\_P23Ins:**

{

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAwakenState: 1表示允许短信、电话可唤醒功能, 0表示取消短信、电话可唤醒功能, 默认为开启

}

**FInsType=P32 ( 主动进入休眠指令 )****FInsType=P37 ( 设置G-sensor参数 )****FT\_P37Ins**

{

FOperationType: 操作类型 (1: 设置; 0: 查询)

FValue1: 运动检测加速度的临界点, 范围是63到500,单位是mg; 如果设置为0则为关闭G-sensor相

关全部功能；关闭G-sensor功能后，如需重新开启G-sensor功能只需重新设置有效的G-sensor参数即可.默认值126

FValue2: 运动检测的加速度临界值

}

### **FInsType=P38 ( 查询/设置开锁报警时间间隔参数 )**

#### **FT\_P38Ins**

{

FOperationType: 操作类型（1: 设置；0: 查询）

FValue: 开锁报警时间间隔设置，即从锁电机处于开锁状态时刻算起，如果该状态持续保持超过120分钟，则可触发一个开锁报警并开始下一次报警计时，范围是3到180,单位是分钟；默认为120分钟

}

### **FInsType=P40 ( 查询/设置GPRS通道和短消息通道的报警开关 )**

#### **FT\_P40Ins**

{

FOperationType: 操作类型(1:设置；0: 查询)

FLockRope: 锁挂绳剪断报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FIllegalCard: 刷非法卡报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FLongTimeOpened:开锁状态保持一段时间报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FFivePasswordError:指令开锁密码连续输错5次报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FVibration:震动报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FInGeoFence:进区域报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FOutGeoFence:出区域报警的开关(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FLowPower:低电报警（电量低于满电的30%）(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FUnCoverBack:开后盖报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FLockStuck:卡锁报警(0:GPRS和SMS都关闭，1: 只开启GPRS，2: 只开启SMS，3: GPRS与SMS都开启)

FLowGSM: GSM信号低报警

FOverSpeed: 超速报警

FDynamicToStatic: 动态转静态报警

FStaticToDynamic: 静态转动态报警

}

### **FInsType= P54 ( 查询/设置开关休眠模式指令 )**

**FT\_P54Ins**

```
{
FOperationType: 操作类型（1：设置；0：查询）
FStatus: 0 关闭, 1 开启不休眠
}
```

**FInsType= P61 ( 电量低报警提示阈值设置 )****FT\_P61Ins**

```
{
FOperationType: 操作类型（1：设置；0：查询）
FValue: 电量低报警提示的阈值
}
```

**FInsType= P62 ( 查询/设置里程统计相关参数 )****FT\_P62Ins**

```
{
FOperationType: 操作类型（1：设置；0：查询）
FValue: 速度值/里程值
}
```

**FInsType= P63 ( 静态飘移处理功能设置 )****FT\_P63Ins**

```
{
FOperationType: 操作类型（1：设置；0：查询）
FStatus: 1开启, 0关闭(默认)
}
```

**FInsType= P68 ( 查询SIM卡的IMSI/ICCID号 )****FT\_P68Ins**

```
{
FType: 1: 查询IMSI, 2:查询ICCID
}
```

**FInsType= WLN2 ( 查询/设置从机工作间隔 )****FT\_WLN2Ins**

```
{
FAssetID: 设备ID
FOperationType: 操作类型（1：设置；0：查询）
FValue: 从机工作时间间隔为20分钟，单位分钟，最小1分钟，最大1440一天，默认5分钟
}
```

**FInsType= WLN4 ( 查询主从及版本 )****FInsType= WLN8 ( 从机指令下发 ( JT126 ) )****FT\_WLN8Ins**

```
{
```

FAssetID: 设备ID

FOperationType: 操作类型（1: 设置；0: 查询）

FType: 3: 删除历史数据；4:重置电量

FTime: 生命周期(单位分钟，指令在主机存储的有效时间)

FSubAssetID: 从机ID

}

**FInsType= ZZZZ (自定义指令)**

FInsContent: 指令内容

# JT707/JT709A/JT705/JT702/JT301设备类型指令类型对象

FInsType= BASE1（查询设备版本号）

FInsType= BASE2（终端自动请求授时(同步GMT时间)）

FT\_BASE2Ins

{

FAssetID: 设备ID

FTime: 授时时间UTC时间(20111018123820设置的时间: 年月日时分秒,解析为:2011-10-18 12:38:20)

}

FInsType= BASE3（远程重启终端(硬件支持有效)）

FInsType= BASE4（恢复终端出厂默认值）

FT\_BASE4Ins

{

FAssetID: 设备ID

FOperationType: 1表示全部信息恢复为出厂默认值; 2表示除通信主从IP外, 其它信息全部恢复为出厂默认值; 3表示除通信主从IP、VIP号码外, 其它信息全部恢复为出厂默认值;

}

FInsType= BASE48（设置报警开关）

**-这些字段根据设备协议来传，所有字段非必填**

FT\_BASE48Ins

{

FAssetID: 设备ID

FOperationType: 1: 设置; 0: 查询 ;2 外接模块报警查询 ;3 外接传感器设置设置.

FLockRope: 锁挂绳剪断报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FIllegalCard: 刷非法卡报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLongTimeOpened:开锁状态保持一段时间报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FFivePasswordError:指令开锁密码连续输错5次报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FVibration:震动报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FInGeoFence:进区域报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FOutGeoFence:出区域报警的开关(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLowPower:低电报警（电量低于满电的30%）(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FUnCoverBack:开后盖报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLockStuck:卡锁报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLowGSM: GSM信号低报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（705A产品才有）

FOverSpeed: 超速报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FErrorUnlock: 错误开锁流程(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（705A产品才有）

FTiltAlarm: 倾斜报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（705A产品才有）

FLongTimeStop: 长时间停车(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（JT705A\_RH产品才有）

FInLoadAlarm: 进道路报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（JT705A\_RH产品才有）

FOutLoadAlarm: 出道路报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（JT705A\_RH产品才有）

FOutFenceUnlockAlarm: 出围栏未上锁报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)（JT705A\_RH产品才有）

—以下JT301x设备新增

FHighTempAlarm: 高温报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLowTempAlarm: 低温报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FHighHumidityAlarm: 高湿度报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLowHumidityAlarm: 低湿度报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FHighLightAlarm: 强光度报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLowLightAlarm: 弱光度报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FHighPressureAlarm: 高压报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLowPressureAlarm: 低压报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FOpenDoorAlarm: 开关门报警(0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLedHightVoltage: LED高电压报警 0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLedHighElectric: LED高电流报警0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

FLockException:门锁异常打开报警0:GPRS和SMS都关闭, 1: 只开启GPRS, 2: 只开启SMS, 3: GPRS与SMS都开启)

}

FInsType= BASE6 (查询/设置上传间隔和休眠定时唤醒间隔)

FT\_BASE6Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FUploadInterval: 上传间隔, 以秒为单位, 默认60。最长1小时, 最短5秒

FAwakenInterval: 设备正常工作间隔, 单位分钟, 默认60分钟. 当该值为0时, 启用持续追踪模式, 设备不休眠。最长24小时, 最短5分钟

注: 正常工作间隔的意思是在没有设置闹钟, 也没有被事件触发唤醒

}

FInsType= BASE7 (查询/设置终端休眠模式)

FT\_BASE7Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FType: 0正常休眠模式, 1表示增加短信电话可以唤醒

}

FInsType= BASE8 (设置设备的工作时差, UTC时间转换成本地时间为UTC时间+时差)

FT\_BASE8Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FTimeDifference: 时差(单位分钟; 设备的工作时间以UTC时间为标准, 转换成本地时间为UTC时间+时差)

\*\* JT800和JT501增加摄像头时差配置\*\*

FCameraTimeDiff: 表示摄像头相对于本地时间的时差。出厂默认为: 480; 即8个小时, 可配置参数范围是-720~780

}

FInsType= BASE9 (查询/设置VIP号码)

FT\_BASE9Ins

{

FAssetID: 设备ID

FOperationType: 0表示查询, 后面的参数可以没有; 1表示设置;

FVIP1: 第一个VIP号码 (如果没有设置, 传入0)

FVIP2: 第二个VIP号码 (如果没有设置, 传入0)

FVIP3: 第三个VIP号码 (如果没有设置, 传入0)

FVIP4: 第四个VIP号码 (如果没有设置, 传入0)

FVIP5: 第五个VIP号码 (如果没有设置, 传入0)

### **JT800和JT501增加五个VIP号码**

FVIP6: 第六个VIP号码 (如果没有设置, 传入0)

FVIP7: 第七个VIP号码 (如果没有设置, 传入0)

FVIP8: 第八个VIP号码 (如果没有设置, 传入0)

FVIP9: 第九个VIP号码 (如果没有设置, 传入0)

FVIP10: 第十个VIP号码 (如果没有设置, 传入0)

}

FInsType= BASE10 (查询/设置主从IP地址与端口号,APN及用户名与密码)

FT\_BASE10Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FMainIP: 主IP

FMainPort: 主端口

FSubIP: 从IP

FSubPort: 从端口

FCMNET: APN名称

FAPN: 用户名

FPassword: 密码

}

FInsType= BASE32 (查询/设置设备闹钟)

FT\_BASE32Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FClocks: 闹钟集合, 多个用英文半角逗号隔开, 设备最多支持4个闹钟, 以分钟为单位 (范围为: 0~1439)

}

FInsType= BASE40 (查询/设置仅支持VIP号码)

FT\_BASE40Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FType: 1表示仅支持VIP号码进行设备配置和查询, 0表示非vip号码也可以配置和查询。默认0

}

FInsType= BASE43 (查询/设置里程统计相关参数)

FT\_BASE43Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (0: 查询; 1: 表示设置里程统计的速度限制值操作; 2: 表示设置里

程同步值)

FValue: 如果指令是1,表示里程统计的速度限制值操作(即速度小于改值不统计里程,默认xkm/h,参数0-100);如果指令是2,表示设置里程同步值为x公里

}

FInsType= BASE44 (查询/设置静态漂移处理功能)

FT\_BASE44Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FType: 1打开这个静态漂移处理功能, 0表示关闭静态漂移处理功能

}

FInsType= BASE50 (查询/设置Ble蓝牙工作时间)

FT\_BASE50Ins:

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询) (查询需传入)

FValue: Ble蓝牙工作时间(取值范围1-5,单位分钟,默认3分钟)

}

FInsType= BASE51 (查询/设置休眠唤醒间隔)

FT\_BASE51Ins:

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询) (查询需传入)

FStaticUploadInterval: 静态模式上报间隔, 取值范围10-1440,单位分钟

FDynamicUploadInterval: 运动模式上报间隔,取值范围10-1440,单位分钟, 此项需要设备具备运动检测功能,且打开了此功能,此字段值才会生效.

}

FInsType= BASE52 (查询/设置主机设备名称)

FT\_BASE52Ins:

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FName: 主机设备名称

}

FInsType= BASE58 (查询/设置唤醒模式)

FT\_BASE58Ins:

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FModel: 模式0: 定时模式, 1: 智能模式(自动根据震动切换动静态间隔)

}

FInsType= BASE59（查询/设置环境异常报警参数）

FT\_BASE59Ins:

{

FAssetID: 设备ID

FOperationType: 操作类型（1: 设置; 0: 查询）

FType: 1:温度参数,2:湿度参数,3:光照参数,4:气压参数

FLowThreshold: 低门限值

FHighThreshold: 高门限值

注释:

1:温度参数, 单位°C,范围(-50~125), 值为-50不生效

2:湿度参数, 单位%, 范围0-100%, 值为0不生效

3:光照度参数,单位lx,取值0-65535 ,值为0不生效

4:气压参数,单位mbar,取值0-65535 ,值为0不生效

}

FInsType= BASE61（查询/设置长时间停车参数）

FT\_BASE61Ins

{

FAssetID: 设备ID

FOperationType: 操作类型（1: 设置; 0: 查询）

FValue:停车时间(停车时间, 单位秒, 范围30-86400, 0值则为取消判断)

}

FInsType= BASE78（查询/设置休眠时环境异常检测参数）

FT\_BASE78Ins

{

FAssetID: 设备ID

FOperationType: 操作类型（1: 设置; 0: 查询 2:外接传感器查询 3:外接传感器设置）

FValue: 1:温度参数 2湿度参数 3 光照参数 4气压参数

FTime: 环境监测时间 范围 5s-65535s 0: 关闭环境监测功能

FExceptionTime:再监测环境异常唤醒设备 范围 60,min-1440min

}

FInsType= GSENS1（查询/设置震动检测阈值）

FT\_GSENS1Ins

{

FAssetID: 设备ID

FOperationType: 操作类型（1: 设置; 0: 查询）

FValue: 单位是mg（gsensor芯片63mg为阶梯配置, 实际芯片配置值为500/63取整=7), 最小63, 最大值8000。当震动达到这个值, 会检测到一次震动。如果值等于0, 关闭该功能

}

FInsType= GSENS2（查询/设置震动报警值）

FT\_GSENS2Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FValue: 震动报警值: 有效范围是63到8000,单位是mg, 默认是0关闭震动报警功能。必须比震动唤醒值大

}

FInsType= DEBUG5 (查询/设置短信唤醒)

FT\_DEBUG5Ins

{

FAssetID: 设备ID

FOperationType: 1表示设置, 0表示查询

FType: 0表示关闭短信唤醒, 1表示开启短信唤醒

}

FInsType= DEBUG7 (电池产品入库前进入休眠)

FInsType= DEBUG15 (设置设备进入休眠的时间)

FT\_DEBUG15Ins

{

FAssetID: 设备ID

FTime: 表示设备多少秒钟后进入休眠

}

FInsType= DEBUG27 (查询物联卡信息)

FInsType= DEBUG29 (查询/清除历史(盲区)数据)

FT\_DEBUG29Ins

{

FAssetID: 设备ID

FOperationType: 操作模式,1清除,0查询

}

FInsType= DEBUG30 (查询/设置超速报警速度)

FT\_DEBUG30Ins

{

FAssetID: 设备ID

FOperationType: 操作模式,1清除,0查询

FSpeedThreshold:速度报警阈值

FDuration: 持续时长

}

FInsType= DEBUG31 (查询/设置低电报警数值)

FT\_DEBUG31Ins

{

FAssetID: 设备ID  
 FOperationType: 操作模式,1清除,0查询  
 FAlarmThreshold:低电量报警阈值 (百分比)  
 }

FInsType= DEBUG34 (设置语音播报音量和语言)  
 FT\_DEBUG34Ins  
 {  
 FAssetID: 设备ID  
 FOperationType: 操作模式,1设置,0查询  
 FVolume: 语音播报音量,音量设置范围为1-15,1对应音量最小,默认为9  
 FLanguage: 语言选择 0:中文 1:英文 ,默认中文  
 }

FInsType= ELOCK19 (查询/设置自动上锁时间)  
 FT\_ELOCK19Ins  
 {  
 FAssetID: 设备ID  
 FOperationType: 操作类型 (1: 设置; 0: 查询)  
 FValue: 自动上锁时间, 1表示1分钟后自动上锁,单位分钟,取值1~10  
 }

FInsType= ELOCK22 (查询/设置开锁/关锁上报间隔)  
 FT\_ELOCK22Ins  
 {  
 FAssetID: 设备ID  
 FOperationType: 操作类型 (1: 设置; 0: 查询)  
 FFunctionType: 0关闭该功能, 1: 打开该功能  
 FUnLockUploadInterval: 开锁时上传间隔, 单位(秒)。此值默认为60秒。最小值为5, 最大值为3600  
 FLockUploadInterval: 关锁时上传间隔, 单位(秒)。此值默认为60秒。最小值为5, 最大值为3600  
 }

FInsType= CAMERA1 (查询/设置录像时间)  
 MT\_CAMERA1Ins  
 {  
 FAssetID: 设备ID  
 FOperationType: 操作类型 (1: 设置; 0: 查询)  
 FTime:表示常规录像时间30s,单位秒,取值范围是1-60s (开关锁, 定时事件)  
 }

FInsType= CAMERA2 (查询/设置拍照连拍)  
 MT\_CAMERA2Ins  
 {  
 FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FModel:表示拍照方式1:单次拍照,>1则连续拍照次数,取值范围1-10张

FInterval:连拍的时间间隔,单位mS,取值范围100mS-30000mS

}

FInsType= CAMERA3 (查询/设置视频拍照关联模式)

MT\_CAMERA3Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FModel: 当事件触发时,视频拍照功能关联模式,0:只录像,1:只拍照,2:先拍照后录像,3:同时拍照录像

}

FInsType= CAMERA6 (查询/设置定时拍照录像参数)

MT\_CAMERA6Ins

{

FAssetID: 设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FModel:定时拍照录像功能开启,0功能关闭

FInterval:执行间隔,单位min

}

# GP4000/GP6000/GP6000F设备类型指令类型对象

## “S”系列终端参数设置指令

FInsType= S01（查询当前终端的制造商名称）

FInsType= S02（查询或设置终端当前的主IP地址,端口及APN名称）

```
FT_S02Ins
{
FAssetID:设备ID
FOperationType: 0表示查询,为1表示设置
FIP: IP或者域名
FPort:端口
FAPN: GPRS接入点名称,如中国移动的APN为“CMNET”
FConnectMode: 连接模式, 0: IP, 1: 域名
}
```

FInsType= S03（查询或设置当前的短信息中心号码,即运营商的短信中心号码）

```
FT_S03Ins
{
FAssetID:设备ID
FOperationType: 0表示查询,为1表示设置
FCenterNumber: 短信中心号码,此号码不要添加国际编号
}
```

FInsType= S04（设置或查询当前回传间隔,次数,监控类型,回传类型,打包条数等）

```
FT_S04Ins
{
FAssetID:设备ID
FOperationType: 0表示查询,为1表示设置
FReplyInterval: 为定时回传时,此值表示回传时间间隔,以秒为单位,0表示不回传
FReplyCount: 回传次数,0表示一直回传.从0-65535
FTrackType: 监控类型,是定距监控还是定时监控,0表示定时监控,1表示定距监控
FReturnType: 回传类型,是通过GPRS还是通过SMS回传,0表示通过GPRS回传,1表示通过SMS回传
FPacketCount: 打包条数设置,从[1-15],若为0表示不进行打包,采取默认的一条数据一个定位点的方式.设置为0或1本质上是一样的.都只有一条定位回传信息
}
```

FInsType= S05（查询或设置终端的车主手机号）

```
FT_S05Ins
{
```

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FIndex:序号终端最多支持5路车主手机号,序号从1-5

FTeleNumber: 车主的手机号,不要添加国际编号

}

FInsType= S06 (查询或设置终端的电子里程速比值及里程统计方法)

FT\_S06Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FValue: 速比值,从1-65535,0表示不启用电子里程,采用GPS里程进行里程统计

}

FInsType= S07 (查询或设置终端的备用IP及端口.)

FT\_S07Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FBackupIP: 备用的IP地址,同主IP地址的书写格式一样.

FBackupPort: 备用端口

FOnOrOff: 是否启用备用IP,1为启用,0为不启用

}

FInsType= S09 (查询当前里程值或设置终端当前里程值(里程同步功能))

FT\_S09Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FMileage: 当前要设置的里程起始值,从0-0xFFFFFFFF之间均可.

}

FInsType= S11 (查询或设置当前终端的防爆功能)

FT\_S11Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FOnOrOff: 为0表示不开启,1表示开启.

}

FInsType= S12 (是否允许用短信对终端进行控制)

FT\_S12Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FAllow: 0:不允许任何短信息设置;1:允许任何号码的短信息进行设置. (默认值);2:只允许中心短信息号码进行设置.

}

FInsType= S13 (用来设置接原车油位还是外接油位传感器)

FT\_S13Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FStatus: 为0表示接原车油位, 为1表示接外接油位传感器.

Ftime: 接原车油位时, A C C开多长时间才开始检测油位

}

FInsType= S14 (获取当前软件版本)

FInsType= S15 (设置回传模式)

FT\_S15Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FReturnMode: 回传模式.

模式 说明

0 TCP,终端所有数据均通过TCP连接回传. (默认)

1 UDP,终端与服务器的连接方式为UDP.

2 SMS,终端所有数据均通过短信发送到短信中心.当采用此模式时,终端不会尝试连接GPRS网络.

}

FInsType= S16 (设置或查询用户存储器)

FT\_S16Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FIndex: 索引,从0-2,共三个值,表示三个用户存储器ID.

FValue: 需存储的数据

}

FInsType= S17 (查询本次已经连续驾驶多长时间了)

FT\_S17Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FReturnInterval: 回传间隔,多长时间回传一次当前的运行报告,以分钟为单位.

}

FInsType= S18 (GPS星历数据)

```

FT_S18Ins
{
FAssetID:设备ID
FOperationType: 0表示查询,为1表示设置
FUpInterval: 定时上传星历数据的间隔,以分钟为单位,若为0,则表示不上传星历数据.
}

```

FInsType= S19 (设置或查询当前终端加载的外设)

```

FT_S19Ins
{
FAssetID:设备ID
FOperationType: 0表示查询,为1表示设置
FPortID: 表示端口号,GP6000的端口号从1-3.
FDeviceID:
DviceID 设备 波特率
1 温度采集器 9600bps
2 调度屏 9600bps
3 未定义 -
4 打印机 -
5 扫描枪 9600bps
6 调试手柄 开发中
7 未定义 -
8 摄像头 115200bps
9 RFID 9600bps
10 未定义 -
11 串口数字键盘 9600bps
12 图像采集器 115200bps
13 希腊客户RFID需求 9600
14 数字油位传感器 9600
15 有源RFID 19200
16 胎压外设 9600
17 LEDScreen 9600
18 LEDScreenA 9600
19 混凝土搅拌机 9600
20 JT606(波特率可设) 9600(默认)
21 青青子木摄像头 115200bps
22 计价器 9600
25 双向通话外设 ---
26 保留
27 北方测控(DABECO)的温湿度传感器(只使用温度,波特率默认9600,地址默认1) 9600(默认)
28 GPS数据输出端口 9600设置外设自动设置波特率
29 CANBUS 9600(默认)
30 EPSILON 19200
31 Shafi 定制RFID 19200

```

32 定制RFID 9600(默认)  
 33 RFID刷卡上传报告 9600(默认)  
 34 印度尼西亚调度屏 19200(默认)  
 35 701S无线模块加载 9600默认  
 36 洲斯温湿度传感器 115200  
 37 超声波油位传感器 115200  
 }

FInsType= S20 (设置或查询当前终端外设端口的波特率)

FT\_S20Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FPortID: 示端口号,GP6000的端口号从1-3.

FBaudRate: 某个端口对应的波特率

BaudRate ID 波特率

0 1200bps

1 2400bps

2 4800bps

3 9600bps

4 19200bps

5 28800bps

6 38400bps

7 57600bps

8 115200bps

}

FInsType= S22 (当前查询终端的GSM网络信号值或设置网络信号定时上传间隔)

FT\_S22Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FUpInterval: 定时上传GSM网络信号的时间间隔,以分钟为单位, 为0表示不上传

}

FInsType= S23 (查询当前终端SIM卡的CCID+IMSI号)

FInsType= S24 (设置或查询连接到GPRS网络时要用到的用户名与密码)

FT\_S24Ins

{

FAssetID:设备ID

FOperationType: 0表示查询,为1表示设置

FUserName: 用户名

FPassword: 密码

}

FInsType= S25（查询终端的历史轨迹数据）

FT\_S25Ins

{

FAssetID:设备ID

FStartPos: 起点,表示从当前最后一个点往前移多少个点

FReadCount: 为读取的点的个数, 此值表示从起点值开始读取多少个点

}

FInsType= S27（设置与格林威治标准时的时差）

FT\_S27Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FTimeDifference: 时差,支持正负数,若为正数,则直接写入数字即可,或为负数,则不能取消“-”号.初始化此值为0.以分钟为单位

}

FInsType= S29（设置或查询是否启用日里程功能）

FT\_S29Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FOnOrOff: 是否启用了日里程功能,1表示启用,0表示不启用.初始化为不启用

}

FInsType= S30（设置或查询是否启用了运行段报告）

FT\_S30Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FOnOrOff: 是否启用了运行段报告功能,1表示启用,0表示关闭.默认为关闭

}

FInsType= S31（查询终端的通话记录）

FT\_S31Ins

{

FAssetID:设备ID

FIndex: 读取哪一个通话记录,从1到10,终端最多可以保留10次通话记录

}

FInsType= S32（设置省电模式）

FT\_S32Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FSaveMode: 省电模式,一共有三种省电模式

Mode 说明

0 无省电模式

1 标准省电模式,当ACC关时或者车静止,自动转到所设置的回传间隔.

2 加强型省电模式,当ACC关时,关闭GPS功能,总是以省电模式的回传间隔及次数回传最后一条定位数据.其他车辆状态正常.

3 深度省电模式,当ACC关时,关闭GPS及GSM模块,除非ACC开.

FUpCount: 电模式下的回传次数,为0表示一直回传,正常取值范围为1-65535

FUpInterval: 回传间隔,0表示不回传,正常取值范围为1-65535.以秒为单位

FJudgeMode: 判断模式, ACC或者速度, 0为ACC, 1为GPS速度

}

FInsType= S33 (设置是否启用历史数据实时上传功能)

FT\_S33Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FOnOrOff: 是否启用历史数据实时上传,0表示禁用1表示启用

}

FInsType= S35 (设置历史数据的存储间隔,默认为一分钟)

FT\_S35Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FInterval: 记录间隔,以秒为单位,从0-65535,0表示不进行保存,默认为60秒

}

FInsType= S37 (设置是否启用GPRS/SMS通道互换)

FT\_S37Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FOnOrOff: 是否启用GPRS/SMS通道互换

FStartTime: GPRS网络无效最长时间,也即SMS通道的启用时间,当GPRS网络无信号,且累计时间超过此值时,即启用SMS通道.以秒为单位

FInterval: 当启用SMS通道时,间隔多长时间回传一次当前定位数据.一般而言,出于成本考虑,GPRS通道的回传间隔会比SMS通道的回传间隔要短,如GPRS通道每10秒回传一条,SMS每五分钟回传一条等

}

FInsType= S41 (设置是否启用震动检测)

FT\_S41Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FOnOrOff: 1表示开启此功能,0表示关闭此功能,默认关闭  
}

FInsType= S42 (设置是否启用芯片温度检测上报)

FT\_S42Ins  
{  
FAssetID:设备ID  
FOperationType: 操作类型 (1: 设置; 0: 查询)  
FOnOrOff: 1表示开启此功能,0表示关闭此功能,默认关闭  
FUpInterval:上报时间间隔,以分钟为单位.最大支持1000分钟  
}

FInsType= S43 (设置智能上传时间管理)

FT\_S43Ins  
{  
FAssetID:设备ID  
FOperationType: 操作类型 (1: 设置; 0: 查询)  
FType: 类型,表明当前是设置日期参考,时间参考还是星期参考  
Type值 说明  
"T" 时间参考  
"W" 星期参考  
"D" 日期参考  
FOnOrOff:上报时间间隔,以分钟为单位.最大支持1000分钟  
FPara:  
Type值 参数个数 说明  
"T" 必须为2个 起始时间(如9),结束时间(如17),以小时为单位,即9表示9点.17表示17点.  
"W" 小于等于7个 需要上传的星期按大小依次列出,如1,3,5表示周一,周三,周五上传,其余时间不上传.  
"D" 小于等于31个 需要上传的日期按大小依次列出,如1,2,3,4,5,11,12,13,14,15表示1-5号,11-15号上传,其余时间不上传.  
}

FInsType= S44 (恢复出厂设置,将终端恢复到出厂时的状态)

FInsType= S45 (设置当前车牌号,终端在发送报警短信息时,会将此车牌信息附加报警短信中)

FT\_S45Ins  
{  
FAssetID:设备ID  
FOperationType: 操作类型 (1: 设置; 0: 查询)  
FCarNo: 车牌号,长度不超过15字节,且只能是ASCII字符  
}

FInsType= S46 (设置从连接(UDP)的IP地址及端口号)

FT\_S46Ins  
{  
FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）  
 FUDPIP: UDP连接的IP地址.形如“XXX.XX.X.XX”样式  
 FUDPPort: UDP连接的端口  
 FUDPDomainOrIP: 是以域名还是IP的形式连接指定地址.0表示IP地址,1表示域名  
 }

FInsType= S46（设置从连接(UDP)的IP地址及端口号）  
 FT\_S46Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FUDPIP: UDP连接的IP地址.形如“XXX.XX.X.XX”样式  
 FUDPPort: UDP连接的端口  
 FUDPDomainOrIP: 是以域名还是IP的形式连接指定地址.0表示IP地址,1表示域名  
 }

FInsType= S47（设置终端是否自动接听来电）  
 FT\_S47Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FAutoPick: 是否自动接听来电  
 }

FInsType= S47（设置终端是否自动接听来电）  
 FT\_S47Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FAutoPick: 是否自动接听来电  
 }

FInsType= S49（设置最长通话时间,默认最长为15分钟）  
 FT\_S49Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FValue: 时长,以秒为单位.最大值为900秒  
 }

FInsType= S51（设置终端的网络心跳）  
 FT\_S51Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）

FOnOrOff: 是否开启此功能,1表示开,0表示关,若当前使用UDP方式,则终端自动打开此功能.若当前使用TCP方式,则可选择是否开启,若当前使用SMS方式,则不会开启

FMaxLostTimes: 允许最大心跳丢失次数,心跳丢失是指发出心跳数据,无回复,当无回复累积到一定次数时,会重启当前连接

}

FInsType= S52 (设置与查询是否启用连接维护心跳包, 以及上传时间间隔)

FT\_S52Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FOnOrOff: 是否启用连接维护心跳包(1:是;0:否)

FInterval: 当启用连接维护心跳包时,间隔多长时间,以秒为单位

}

FInsType= S54 (设置是否开启转弯上传数据功能)

FT\_S54Ins

{

FAssetID:设备ID

FTrunUp: 1表示开启此功能,0表示关闭此功能

FTrunValue: 转弯角度门限值,只有转弯角度大于此值时,才会上传当前点数据.默认为10

}

FInsType= S55 (设置与查询重启时间)

FT\_S55Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FResetTime: 重启时间,范围是900到65535,单位是秒

}

FInsType= S56 (设置与查询副卡APN,用户名,密码)

FT\_S56Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAPN: 接入点名称

FUserName: 用户名

FPassword: 密码

}

FInsType= S57 (设置与查询禁用双卡中的某张卡)

FT\_S57Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）  
 FCardName: 卡号,1是主卡,2是副卡,0是不禁用任何卡  
 }

FInsType= S58（设置Gsensor的门限值,当加速度达到此值时认为产生了紧急刹车）  
 FT\_S58Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FValue: 报警门限值,范围(10~75). 设置的值放大了10倍,比如设置75表示的是7.5g,当加速度达到7.5g时产生紧急刹车报警,默认是75  
 }

FInsType= S61（设置发电机运行时间统计值的初值）  
 FT\_S61Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FValue: 发电机运行时间统计值，以分钟为单位，范围0~71582788.  
 }

FInsType= S62（设置GPS里程速比）  
 FT\_S62Ins  
 {  
 FAssetID:设备ID  
 FOperationType: 操作类型（1：设置；0：查询）  
 FSign: 为0表示设置是负数,减少多少公里，为1表示设置为正数，增加多少公里  
 FValue: GPS里程统计速比值,从1~100  
 }

## “C”系列终端远程控制指令

FInsType= C01（要求终端进行断电重启）

FInsType= C02（要求对超速报警喇叭进行测试,一般用于安装过程中测试喇叭是否安装正确）

FInsType= C06（请求终端拨打指令号码,以便进行监听）

FT\_C06Ins  
 {  
 FAssetID:设备ID  
 FPhone: 请求终端拨打指令号码,以便进行监听  
 }

FInsType= C07（进行自定义控制IO口）

FT\_C07Ins

```

{
FAssetID:设备ID
FOperationType: 操作类型（1： 设置； 0： 查询）
FID: 控制IO口的ID号.如下:1表示喇叭.2表示转向灯3表示开门4表示关门
FCtrlStatus: 1 表示有输出, 0 表示取消当前输出或无输出
FKeepTime: 保持时间,当CtrlStatus为 1 时才有用,最多254秒.当该值等于0时就一直控制
}

FInsType= C08（提醒司机有疲劳驾驶行为）
FT_C08Ins
{
FAssetID:设备ID
FOperator: 为1表示，系统收到SOS报警后下发该指令;为2表示，当系统统计每一司机有疲劳驾驶行为
}

FInsType= C09（下发有关RFID卡的状态信息）
FT_C09Ins
{
FAssetID:设备ID
FOperators:
[
{
FOnDutyCardNumber: 上班卡号
FOffDutyCardNumber: 下班卡号
FOnDutyCardStatus: 上班卡状态, 1表示空闲状态，即可以登录，2表示登陆中，3表示休息中，4表示单班超14小时/总班超70小时，5表示单班超13.5小时/总班超69.5小时
FSingleOnDutyTime: 单次上班时间
FTotalOnDutyTime: 总上班时间
}
{
FOnDutyCardNumber: 上班卡号
FOffDutyCardNumber: 下班卡号
FOnDutyCardStatus: 上班卡状态, 1表示空闲状态，即可以登录，2表示登陆中，3表示休息中，4表示单班超14小时/总班超70小时，5表示单班超13.5小时/总班超69.5小时
FSingleOnDutyTime: 单次上班时间
FTotalOnDutyTime: 总上班时间
}
{
FOnDutyCardNumber: 上班卡号
FOffDutyCardNumber: 下班卡号
FOnDutyCardStatus: 上班卡状态, 1表示空闲状态，即可以登录，2表示登陆中，3表示休息中，4表示单班超14小时/总班超70小时，5表示单班超13.5小时/总班超69.5小时
FSingleOnDutyTime: 单次上班时间
FTotalOnDutyTime: 总上班时间
}
}
}

```

```
]
}
```

FInsType= C10（RFID强制登出）

FT\_C10Ins

```
{
```

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FOnDutyCardNumber: 上班卡号

FOffDutyCardNumber: 下班卡号

```
}
```

## “A”系列报警设置/控制指令

FInsType= A01（用于设置终端的报警开关）

FT\_A01Ins

```
{
```

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FAlarmID:

ID号 报警类型

1 进入电子围栏报警

2 出电子围栏报警

3 超时停车报警

4 GPRS模块重启报警(视版本开放)

5 疲劳驾驶报警

6 超速报警

7 SOS报警

8 后备电池报警

9 非法开门报警(视终端版本)

10 非法点火报警(视终端版本)

11 油位报警

12 拖吊报警

13 震动报警

14 热点进出报警

15 区域道路进出报警

16 区域道路超速与超时停车报警

17 区域道路时间相关报警,如在禁止时间内到达某路

18 外接数字油位报警(适用于GP6000)

19 万能端口脉冲计数报警

20 扩展AD一报警(适用于GP6000)

21 扩展AD二报警(适用于GP6000)

22 外设之温度报警

23 安全带报警

- 24 紧急刹车报警
- 25 空挡滑行报警
- 26 胎压报警
- 27 GPRS重连报警
- 28 GPS天线断开报警
- 29 JT606断开报警
- 30 调度屏断开报警
- 31 CANBUS连接断开报警
- 32 CANBUS故障码上传报警
- 33 规定时间段内不允许开车报警

FAlarmSend: 是否发送报警.即当报警产生后,是否将当前警情发送到中心

FAlarmAckTime: 最大确认时间间隔,当在此时间间隔内,没有收到中心的确认,则会再次发送报警信息; 此值最大值为7,以10秒为单位,即最多不能超过70秒就会再次发送报警

FAlarmSMSSend: 是否将报警以短信息形式发送到车主手机号码中

}

FinsType=A02 (初始化报警设置指令)

FInsType= A04 (设置超时停车报警门限值)

FT\_A04Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FStopTime: 停车报警门限值,以分钟为单位,取值范围为1-4320分钟之间.当确认停车之后,再累计停车超过此值,即产生超时停车报警

FStart: 有效时间起点, (024表示从0点到24点)

Fend: 有效时间结束点, (024表示从0点到24点)

}

FInsType= A05 (设置超时停车报警门限值)

FT\_A05Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FSpeedValue: 超速门限值,从0-255,以节为单位,若为0,则表示关闭超速监控功能

FKeepTime: 持续时间,从3-255,以秒为单位,最小值为3秒钟

FGpsValid: 定位与否,若为1,则表示只有定位时(A)才会判断速度是否有效,否则直接认为未超过速度限制,若为0,则表示不管定位与否,均对其速度进行判断

FUpInterval: 上报间隔时间,若遇持续超速驾驶,则设定每隔一段就上报一次当前超速状态.以秒数为单位, 最小值限定为60秒

}

FInsType= A06 (是否开启车内超速报警提示)

FT\_A06Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FBeep: 为0表示不开启车内超速提示,为1表示开启车内超速提示

}

FInsType= A07（疲劳驾驶报警设置）

FT\_A07Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FAlarmValue: 疲劳驾驶的门槛值,以秒为单位,最大支持60000秒.为0表示取消疲劳驾驶报警功能

FReleaseTime: 疲劳驾驶解除时间,当停车超过此时间时,即视为疲劳驾驶报警解除,以秒为单位.此值不能大于AlarmValue,否则会取消疲劳驾驶功能

FUpInterval: 报警上报间隔,以秒为单位.若疲劳驾驶报警未被解除,此时会以此间隔上报当前的报警状态,其中已连续驾驶时间参数会不断变化

}

FInsType= A08（查看疲劳驾驶报警记录）

FT\_A08Ins

{

FAssetID:设备ID

FIndex: 疲劳驾驶索引,最后一次索引值为1,次新为2,最旧的一次疲劳驾驶报警记录为10

}

FInsType= A09（设置主油位报警参数指令）

FT\_A09Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FAlarm: 为0表示不启用油位报警,为1表示启用油位报警.

FValue: 油位报警门限值,最低为5,一般推荐为169

FCheckInterval: 检测间隔,初始化为5分钟,以分钟为单位

FReleaseTime: 油位的报警判断门限时间值,即当一次油位变化的时间超过此值(如30分钟)时,不再判断是否报警.以分钟为单位;若此值为0,表示只要油位变动超过设置值,均报警.初始值为0,最大值为2000分钟

FReferenceVelocity: 油位报警是否参考速度参数,参考速度时(参数为1),不参考速度时(参数为0)

}

FInsType= A10（设置拖吊报警参数指令）

FT\_A10Ins

{

FAssetID:设备ID

FOperationType: 操作类型（1：设置；0：查询）

FAlarm: 是否启用拖吊报警.1表示启用,0表示不启用,初始化为0.

FTime: 保持时间,即达到最小速度的最小保持时间.初始化为0秒  
}

FInsType= A11 (设置安全带报警参数指令)

FT\_A11Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAlarm: 是否启用安全带报警.1表示启用,0表示不启用,初始化为0.

FSpeed: 最小速度门限值,超过此速度即可以判断是否扣好安全带。在此速度、以下,即使未系安全带,不做判断也不会报警

FTime: 保持时间,即达到最小速度的最小保持时间.初始化为0秒

}

FInsType= A12 (设置紧急刹车报警参数指令)

FT\_A12Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAlarm: 是否启用紧急刹车报警.1为启用,0为不启用,初始为0.

FSpeed: 最小速度门限值,超过此速度且检测到有刹车信号,即开始监测是否会产生紧急刹车

FRate: 速度在指定时间内的下降比率,当超过此比率时,即产生紧急刹车报警。为5-99之间的任一值。若不在此值内,则会返回一条ERROR的回复信息

FTime: 时间容限值,只有在此时间内,速度下降比率超过Rate才会报警。以秒为单位。为1-100之间任一值。若超过此值会被限定为100。若为0,且启用此报警,则会被设置为1

}

FInsType= A13 (设置空档滑行报警参数指令)

FT\_A13Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAlarm: 是否启用空档滑行报警.1为启用,0为不启用,初始为0

FSpeed: 最小速度门限值,超过此速度且启用了空档滑行检测,则会对当前是否空档滑行进行检测

FTime: 时间门限值,当空档滑行保持这个时间以上,才会产生空档滑行报警,以秒为单位

}

FInsType= A14 (设置手工确认短信报警功能)

FT\_A14Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FIndex: 设置第几个手工确认报警功能,取值范围1~5,一共有5个.一次只能设置一个

FAlarmID: 报警的ID号

FSwitch: 是否开启手工确认功能, 1表示开启,0表示关闭  
}

FInsType= A15 (设置等待手工确认短信报警的超时时间)

FT\_A15Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FTime: 超时时间,单位分钟,取值范围3~4294967295,出厂默认3分钟

}

FInsType= A16 (设置时间段内不允许开车的开始和结束时间)

FT\_A16Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FTimeStart: 起始时间,包括小时和分钟 (如: 20:00则传入2000)

FTimeEnd: 结束时间,包括小时和分钟 (如: 20:00则传入2000)

}

FInsType= A17 (设置主油位报警参数指令(区分运动与静止))

FT\_A17Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAlarm: 为0表示不启用油位报警,为1表示启用油位报警

FValue: 车辆静止时的油位报警门限值,最低为5,一般推荐为169

FValue1: 车辆运动中的油位报警门限值,最低为5,一般推荐为169

FCheckInterval: 检测间隔,初始化为5分钟,以分钟为单位.

FReleaseTime: 油位的报警判断门限时间值,即当一次油位变化的时间超过此值(如30分钟)时,不再判断是否报警.以分钟为单位.若此值为0,表示只要油位变动超过设置值,均报警.初始值为0,最大值为2000分钟

FReferenceVelocity: 油位报警是否参考速度参数, 参考速度时 (参数为1), 不参考速度时 (参数为0)

}

## “G”系列地物点设置指令

FInsType= G01 (设置电子围栏整体控制指令)

FT\_G01Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAlarm: 为0表示不启用油位报警,为1表示启用油位报警

FValue: 车辆静止时的油位报警门限值,最低为5,一般推荐为169

FValue1: 车辆运动中的油位报警门限值,最低为5,一般推荐为169

FCheckInterval: 检测间隔,初始化为5分钟,以分钟为单位.

FReleaseTime: 油位的报警判断门限时间值,即当一次油位变化的时间超过此值(如30分钟)时,不再判断是否报警.以分钟为单位.若此值为0,表示只要油位变动超过设置值,均报警.初始值为0,最大值为2000分钟

FReferenceVelocity: 油位报警是否参考速度参数,参考速度时(参数为1),不参考速度时(参数为0)

}

FInsType= G02 (设置单个电子围栏指令)

FT\_G02Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FID: 电子围栏号

FModel: 操作模式; 1:设置围栏信息; 2:此围栏号对应的围栏无效(不会判断后面的值)

FAlarmMode: 报警模式;0:无报警; 1: 无论进出电子围栏,均报警; 2: 进电子围栏报警; 3: 出电子围栏报警

FLTLongitude: 左上角经度

FLTLatitude: 左上角纬度

FRBLongitude: 右下角经度

FRBLatitude: 右下角纬度

FName: 电子围栏名称,其长度不能超过15个字节.名称中不能有","号

FStopRecordData: 当车辆进入此围栏,是否停止记录及上传GPS数据.值为1表示停止,若为0表示允许上传及记录GPS数据.默认为0

FOverSpeed: 该围栏的速度门限值,取值0-255,速度单位为节,为0表示不单独设置。所有的围栏速度判断同样适合A05的速度监控。由于电子围栏只有在GPS定位的情况下才能判断,所以电子围栏内的超速检测只支持GPS定位时检测

}

FInsType= G03 (设置热点指令)

FT\_G03Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FPointID: 为热点ID号,最多可以存储64个热点

FLongitude: 热点的经度

FLatitude: 热点的纬度

FName: 热点的名称,长度不能超过15个字节,名称中不能有","号

FStopRecordData: 当车辆进入此热点,是否停止记录及上传GPS数据.值为1表示停止,若为0表示允许上传及记录GPS数据.默认为0

}

FInsType= G04 (设置是否启用区域与道路功能)

FT\_G04Ins

```
{
FAssetID:设备ID
FOperationType: 操作类型（1： 设置； 0： 查询）
FValid: 1表示启用区域及道路功能,0表示不启用区域及道路功能.默认不启用
}
```

FInsType= G05（设置区域道路的有效距离）

```
FT_G05Ins
{
FAssetID:设备ID
FOperationType: 操作类型（1： 设置； 0： 查询）
FDistance: 有效距离,以米为单位,初始化为100米.
}
```

FInsType= G06（设置某条道路或区域是否有效）

```
FT_G06Ins
{
FAssetID:设备ID
FOperationType: 操作类型（1： 设置； 0： 查询）
FAreaRoadID: 所设定的区域道路ID值.从1开始编号,最大值为100.
FValid: 1表示有效,0为无效,其他值未定义
FStopRecordData: 当车辆进入此道路或区域时,是否停止记录及上传GPS数据.值为1表示停止,若为0
表示允许上传及记录GPS数据.默认为0
}
```

FInsType= G07（设置区域及道路的详细节点）

```
FT_G07Ins
{
FAssetID:设备ID
FOperationType: 操作类型（1： 设置； 0： 查询）
FAreaRoadID: 所设定的区域道路ID值.从1开始编号,最大值为100.
FType: 类型,表明当前是区域还是道路.1表示道路,0或其他值表示区域
FPageID: 详细节点的页ID值.每页最多10个点
FPointCount: 当前设置的点的个数
FPoints: 节点信息,节点必须成对出现,即一个经度一个纬度成对出现,一个经度一个纬度构成一个完整的点,节点信息均是以DDDMM.MMMM形式出现,如11323.1234
}
```

FInsType= G08（设置区域道路监控模式及名称）

```
FT_G08Ins
{
FAssetID:设备ID
FOperationType: 操作类型（1： 设置； 0： 查询）
FAreaRoadID: 所设定的区域道路ID值.从1开始编号,最大值为100.
FAlarmType: 可以取四个值,进报警(值为1)/出报警(值为2)/进出均报警(值为3)/无报警(值为0)
```

FUserDef: 用户定义字段,0-255之间的任一值,终端只对此值进行保存或查询,用于对区域道路进行分组管理

FMaxSpeed: 最大速度,在区域道路内,若车速超过此值,则会产生超速报警,无论是否设置了超速报警(A05指令).若为0表示不判断超速

FTimeStart: 时间起点

FTimeEnd: 时间结束点

FTimeAdjustType:时间判断类型,1表示在时间起点前到达此区域道路则报警,2表示在时间结束点之后到达此区域道路则报警,3表示在时间起点前或时间结束点之后到达此区域道路均报警,4则表示在时间起点及时间终点之间到达此区域道路则报警.若TimeStart与TimeEnd均为0则表示不判断时间类型报警.若此值为0则表示不进行时间相关的判断

FMaxStopTime: 最大停车时间,即在此区域道路内停车超过此值,则会产生报警,无论是否设置了A04指令.以秒为单位

FName: 此区域道路的名称,其长度不能超过15字节.且名称中不能有","号

FInsType= G09 (设置热点的半径)

FT\_G09Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FRadius : 半径值,以米为单位

}

FInsType= G10 (设置区域道路的页总数)

FT\_G10Ins

{

FAssetID:设备ID

FOperationType: 操作类型 (1: 设置; 0: 查询)

FAreaRoadID : 所设定的区域道路ID值.从1开始编号,最大值为100

FpageCount: 区域道路的总页数,最多20页

}

## “D”系列数据传输指令

FInsType= D01 (查询当前有多少外设端口及当前的外设类型)

## “E”系列外设控制指令

FInsType= E01 (温度采集器外设指令集)

设置温度数据上传间隔

FT\_E01Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FoperationType: 操作类型 (1: 设置; 0: 查询)  
 FupInterval: 上传间隔, 以秒为单位  
 }

设置温度报警参数

```
FT_E01Ins
{
  FAssetID: 设备ID
  FOpCode: 固定2
  FoperationType: 操作类型 (1: 设置; 0: 查询)
  FIndex: 温度探头索引号.从1开始编号(若为北方测控温湿度传感器则设为1)
  FAlarmType: 报警类型,0表示无报警,1为高于上限报警,2为低于下限报警,3为高于上限报警及低于下限报警
  FReferACC: 是否参考ACC,1表示参考,当ACC开时,才进行温度的判断.若此值设置为0,则无论ACC是否开关,均对温度是否产生报警进行判断
  FTimeStart: 起始时间.如8,表示8点钟为起始时间,即从8点开始才需要判断温度是否满足报警条件
  FTimeEnd: 结束时间.如18,表示超过18点之后就不需要判断温度报警
  FMax: 温度上限值.因为此值为单字节存储,所以其取值范围为[-128,127].且此值必须要大于Min值
  FMin: 温度下限值. 因为此值为单字节存储,所以其取值范围为[-128,127].且此值必须要小于Max值
  FDuration: 持续时间, 温度超过门限值持续多久才报警,默认为10,取值范围,1~65535,单位为秒.
}
```

FInsType= E02 (调度屏外设指令集)

设置监听号码

```
FT_E02Ins
{
  FAssetID: 设备ID
  FOpCode: 固定1
  FoperationType: 操作类型 (1: 设置; 0: 查询)
  FTelOrder: 号码序号
  FTel: 监听号码, 号码长度最长15位, 最短1位
}
```

设置调度屏的来电授权号码,最多设置15个

```
FT_E02Ins
{
  FAssetID: 设备ID
  FOpCode: 固定2
  FoperationType: 操作类型 (1: 设置; 0: 查询)
  FTelCount: 号码个数
  FTel: 监听号码, 号码长度最长15位, 最短1位
}
```

FInsType= E08 (摄像头外设指令集)

设置各个摄像头的图像大小

FT\_E08Ins  
 {  
 FAssetID: 设备ID  
 FOpCode: 固定1  
 FoperationType: 操作类型 (1: 设置; 0: 查询)  
 FCameraID: 摄像头ID号,从 1 -7,最多支持7个摄像头.  
 FJpegSize: 图像大小  
 Size值 大小  
 1 最小  
 3 较小  
 5 中等  
 7 大  
 }

要求摄像头拍照,回传数据

FT\_E08Ins  
 {  
 FAssetID: 设备ID  
 FOpCode: 固定2  
 FInterval: 图像回传间隔.若值为 0 则表示尽最大努力上传图像数据  
 FCount: 图像回传张数.若设置此值为 0 则表示一直上传,无张数限制.若设置一直上传,则此设置会掉电保存..  
 FCameraID: 摄像头ID号,从 1 -7,最多支持7个摄像头  
 }

设置或查询当前接入的摄像头

FT\_E08Ins  
 {  
 FAssetID: 设备ID  
 FOpCode: 固定3  
 FOperationType: 为0表示查询当前已经接入的摄像头详细数目与ID号,为1表示要求终端对接入的摄像头进行巡检,巡检完成后上报巡检结果  
 }

设置摄像头拍摄的触发条件

FT\_E08Ins  
 {  
 FAssetID: 设备ID  
 FOpCode: 固定4  
 FOperationType: 为0表示查询,为1表示设置  
 FOnOrOff: 是否启用,0表示不启用,1表示启用  
 FSignalID: 触发的信号ID.  
 101 表示产生油位报警时,触发拍照.  
 102 表示进入电子围栏触发拍照.  
 103 表示印度尼西亚MDT按键触发拍照.

FSignalMode: 信号形式,有两种,1表示从开到关,2表示从关到开,3表示从开到关及从关到开均触发

FTimeStart: 起始时间.以小时为单位.

FTimeEnd: 结束时间.以小时为单位.

}

重启摄像头

FT\_E08Ins

{

FAssetID: 设备ID

FOpCode: 固定5

}

查询或者调取TF卡时间段图片

FT\_E08Ins

{

FAssetID: 设备ID

FOpCode: 固定9

FOperationType: 设置或调取, 1为调取, 0为查询

FTimeStart: 开始时间, 格式是年月日时分秒分别占两位.

FTimeEnd: 结束时间.

}

FInsType= E09 (摄像头外设指令集)

设置或查询RFID的ID号

FT\_E09Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 设置或调取, 1为调取, 0为查询

FIDIndex: 要操作的ID号索引, 1-5

FIDValue: 10个字节的ID号.

}

设置或查询RFID上下班卡的ID号

FT\_E09Ins

{

FAssetID: 设备ID

FOpCode: 固定2

FOperationType: 设置或调取, 1为调取, 0为查询

FIDIndex: 要操作的ID号索引, 1-5

FIDValue: 10个字节的上班ID号

FIDValue1: 10个字节的下班ID号.

}

FInsType= E11 (密码小键盘外设指令集)

设置或查询小键盘密码

```
FT_E11Ins
{
FAssetID: 设备ID
FOpCode: 固定1
FOperationType: 设置或调取, 1为调取, 0为查询
FPasswordIndex: 要操作的ID号索引, 1-5
FPassword: 为6个数字的密码
}
```

FInsType= E14 (数字油位传感器外设指令集)

查询当前已经接入的油位传感器列表

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定1
FOperationType: 设置或调取, 1为调取, 0为查询
}
```

设置各个油位传感器的报警门限值

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定2
FOperationType: 设置或调取, 1为调取, 0为查询
FCount: 当前设置的油位传感器个数
FParas:
[
{
Index: 表示第一个油位传感器ID号
Value: 表示此油位传感器对应的报警门限值(报警门限值单位为“升”, 比如说50升)
ReleaseTime: 表示此油位传感器对应的报警 时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000, 在该时间门限值内只上传一次报警信息)
},
{
Index: 表示第一个油位传感器ID号
Value: 表示此油位传感器对应的报警门限值(报警门限值单位为“升”, 比如说50升)
ReleaseTime: 表示此油位传感器对应的报警 时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000, 在该时间门限值内只上传一次报警信息)
},
{
Index: 表示第一个油位传感器ID号
Value: 表示此油位传感器对应的报警门限值(报警门限值单位为“升”, 比如说50升)
ReleaseTime: 表示此油位传感器对应的报警 时间判断门限值(以分钟为单位,若为0表示只要油位变
```

动达到设定值即报警,最大值为2000, 在该时间门限值内只上传一次报警信息)

```
}
]
}
```

查询当前已经接入的油位传感器列表

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定3
FOperationType: 设置或调取, 1为调取, 0为查询
FInterval: 油位值上报的时间间隔, 以秒为单位,若为0,则不再上传.最大值为65535
}
```

设置当前油位值为空油箱值

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定4
FIndex: 需要设置0值的数字油位传感器ID号
}
```

设置当前油位值为满油箱值

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定5
FIndex: 需要设置满值的数字油位传感器ID号
}
```

设置当前接入油位传感器的ID号

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定6
FIndex: 设置当前接入油位传感器的ID号
}
```

设置油位传感器的阻尼

```
FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定7
FIndex: 设置当前接入油位传感器的ID号
FDamp: 阻尼系数0到9,0表示5阻尼是5秒,1至9是1至9分钟
}
```

}

设置油位传感器的体积

```

FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定8
FOperationType: 设置或调取, 1为调取, 0为查询
FCount: 油箱个数
FVolumes:
{
ID: 传感器ID
Volume: 邮箱体积
}
}

```

设置油位传感器的体积

```

FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定8
FOperationType: 设置或调取, 1为调取, 0为查询
FCount: 油箱个数
FVolumes:
{
ID: 传感器ID
Volume: 邮箱体积
}
}

```

设置各个油位传感器的报警门限值(区分运动与静止)

```

FT_E14Ins
{
FAssetID: 设备ID
FOpCode: 固定9
FOperationType: 设置或调取, 1为调取, 0为查询
FCount: 油箱个数
FAlarms
[[
Index: 油位传感器ID号
Value: 车辆静止时此油位传感器对应的报警门限值(报警门限值单位为“升”)
MotionValue: 车辆运动时此油位传感器对应的报警门限值
ReleaseTime: 油位传感器对应的报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到
设定值即报警,最大值为2000, 在该时间门限值内只上传一次报警信息)
]},{

```

Index: 油位传感器ID号

Value: 车辆静止时此油位传感器对应的报警门限值(报警门限值单位为“升”)

MotionValue: 车辆运动时此油位传感器对应的报警门限值

ReleaseTime: 油位传感器对应的报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000, 在该时间门限值内只上传一次报警信息)

},

{

Index: 油位传感器ID号

Value: 车辆静止时此油位传感器对应的报警门限值(报警门限值单位为“升”)

MotionValue: 车辆运动时此油位传感器对应的报警门限值

ReleaseTime: 油位传感器对应的报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000, 在该时间门限值内只上传一次报警信息)

}

]

}

FInsType= E15 (有源RFID外设指令集)

设置RFID标签搜集时间

FT\_E15Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 设置或调取, 1为调取, 0为查询

}

请求重新搜集当前所有的RFID标签

FT\_E15Ins

{

FAssetID: 设备ID

FOpCode: 固定2

}

上传最后一次搜集到的RFID标签

FT\_E15Ins

{

FAssetID: 设备ID

FOpCode: 固定3

}

设置RFID标签搜集时间

FT\_E15Ins

{

FAssetID: 设备ID

FOpCode: 固定4

FOperationType: 设置或调取, 1为调取, 0为查询

FInterval: 0表示不自动上传,以分钟为单位.默认为2分钟  
}

FInsType= E16 (胎压外设指令集)

设置胎压外设上传间隔

FT\_E16Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 设置或调取, 1为调取, 0为查询

FInterval: 胎压值上报的时间间隔, 以分钟为单位,若为0,则不再上传.最大值为1000,默认为1分钟

}

FInsType= E19 (混凝土外设指令集)

设置混凝土转鼓外设上传时间间隔

FT\_E19Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 设置或调取, 1为调取, 0为查询

FInterval: 上报的时间间隔, 以秒为单位,若为0,则不再上传.最大值为1000. 默认60秒.

}

置混凝土转鼓转动的停转门限值

FT\_E19Ins

{

FAssetID: 设备ID

FOpCode: 固定2

FOperationType: 设置或调取, 1为调取, 0为查询

FMax: 停转门限值, 以秒为单位,小于5秒都定为5秒,默认60秒,最大值是65535.即超过Max值则表示已经停止.

}

FInsType= E20 (JT606外设指令集)

查询当前已经接入的JT606列表

FT\_E20Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 设置或调取, 1为调取, 0为查询

}

设置JT606的报警门限值

FT\_E20Ins

{

FAssetID: 设备ID

FOpCode: 固定2

FOperationType: 设置或调取, 1为调取, 0为查询

FCount: 本次共设置多少个传感器的报警参数, 最大值为4

FAlarms

{

Index: 传感器索引, 即表示为第几根JT606.最大值为4, 最小值为1

VolumeChange: 变化的体积报警门限值(单位是升)

ReleaseTime: 报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000)

},{

Index: 传感器索引, 即表示为第几根JT606.最大值为4, 最小值为1

VolumeChange: 变化的体积报警门限值(单位是升)

ReleaseTime: 报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000)

},{

Index: 传感器索引, 即表示为第几根JT606.最大值为4, 最小值为1

VolumeChange: 变化的体积报警门限值(单位是升)

ReleaseTime: 报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000)

}]

}

设置当前油位值上报的时间间隔

FT\_E20Ins

{

FAssetID: 设备ID

FOpCode: 固定3

FOperationType: 设置或调取, 1为调取, 0为查询

FInterval: 油位值上报的时间间隔, 以秒为单位,若为0,则不再上传.默认是十秒钟,最大值为65535.

}

设置当前油位值为空油箱值

FT\_E20Ins

{

FAssetID: 设备ID

FOpCode: 固定4

FIndex: 需要设置0值的数字油位传感器ID号

}

设置当前油位值为满值

FT\_E20Ins

{

FAssetID: 设备ID

FOpCode: 固定5

FIndex: 需要设置满值的数字油位传感器安装号  
}

设置或者查询当前接入的JT606ID号

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定6
FOperationType: 1为设置,0为查询
FIndex: 设置当前接入油位传感器的安装号
}
```

设置JT606的阻尼

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定7
FOperationType: 1为设置,0为查询
FIndex: 设置当前接入油位传感器的安装号
FDamp: 阻尼,单位是秒,范围是5到600秒
}
```

设置当前接入油位传感器的时间

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定8
FOperationType: 1为设置,0为查询
FIndex: 设置当前接入油位传感器的安装号
FTime: 时间（如：20110318110145）
}
```

设置JT606的体积

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定12
FOperationType: 1为设置,0为查询
FCount: 油箱个数
FVolumes
{
Index: 邮箱传感器ID
Volume: 邮箱值
}
}
```

设置JT606的体积

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定15
FOperationType: 1为设置,0为查询
FIndex: JT606ID:ID号
FFreq: 脉冲频率, 范围是90到500
FZero: 零值, 0到4095, 并且小于满值
FFull: 满值, 0到4095
}
```

恢复JT606的出厂参数

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定17
FIndex: JT606ID:ID号
}
```

开启或关闭JT606加油报警

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定18
FOperationType: 1为设置,0为查询
FSwitch: 1为开启,0为关闭
}
```

开启或关闭JT606温飘处理功能

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定19
FOperationType: 1为设置,0为查询
FIndex: JT606ID:ID号
FSwitch: 1为开启,0为关闭
}
```

设置JT606的报警门限值(区分运动与静止)

```
FT_E20Ins
{
FAssetID: 设备ID
FOpCode: 固定20
FOperationType: 设置或调取, 1为调取, 0为查询
}
```

FCount: 本次共设置多少个传感器的报警参数, 最大值为4

FAlarms

{

Index: 传感器索引, 即表示为第几根JT606.最大值为4, 最小值为1

VolumeChange: 变化的体积报警门限值(单位是升)

MotionVolumeChange: 车辆运动时变化的体积报警门限值(单位是升)

ReleaseTime: 报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000)

}{

Index: 传感器索引, 即表示为第几根JT606.最大值为4, 最小值为1

VolumeChange: 变化的体积报警门限值(单位是升)

MotionVolumeChange: 车辆运动时变化的体积报警门限值(单位是升)

ReleaseTime: 报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000)

}{

Index: 传感器索引, 即表示为第几根JT606.最大值为4, 最小值为1

VolumeChange: 变化的体积报警门限值(单位是升)

MotionVolumeChange: 车辆运动时变化的体积报警门限值(单位是升)

ReleaseTime: 报警时间判断门限值(以分钟为单位,若为0表示只要油位变动达到设定值即报警,最大值为2000)

}}

}

开启或关闭JT606温飘处理功能

FT\_E20Ins

{

FAssetID: 设备ID

FOpCode: 固定22

FOperationType: 1为设置,0为查询

FIndex: JT606ID:ID号

FRefuelingThreshold: 加油门限值, 1099L

FOilReductionThreshold:减油门限值, 1099L

}

FInsType= E21 (新版摄像头指令集)

设置当前接入的摄像头ID

FT\_E21Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 1为设置,0为查询

FCount: 当前接入的摄像头个数.最大值为6,即表示最多只允许接入6个摄像头

FSetPara: 接入的摄像头的ID号(如2,5,13)

}

设置图像触发拍照模式

FT\_E21Ins

{

FAssetID: 设备ID

FOpCode: 固定3

FOperationType: 1为设置,0为查询

FCameraID: 摄像头ID号.若指定摄像头ID号不存在,则返回一个错误信息.若此值为0,则表示轮流拍照

FSize:

值 分辨率

1 160 \* 128

2 320 \* 240

3 640 \* 480

**FTriggerValue** : 触发拍照类型,若为0,表示不开启触发拍照,其他值定义见下表

ID 名称 描述

1 ACC

2 发动机

3 刹车

6 空调

7 车门

8 高预留

9 电源

10 断油电

11 SOS报警

12 GPS天线开路

13 GPS天线短路

14 超速驾驶报警

15 万能端口1之IO口状态

16 万能端口2之IO口状态

26 油位检测线3

27 油位检测线2, 空调检测线,刹车检测线,脉冲信号线,车门检测线

28 报警按钮

101 油位异常报警

102 进入电子围栏 进入围栏按照设定的Trigger\_Interval时间间隔一直拍照,直到出围栏停止拍照

103 印度尼西亚MDT

**FTriggerCount**: 表示此次触发拍照需要拍摄的张数,若为0,则表示不拍照.最大值为100.

**FTriggerInterval**: 触发拍照的时间间隔,以秒为单位,表示隔多长时间拍一张图片.需要注意的是,这个时间间隔是指从传完上一张图片所有帧之后至拍新的图片之间的等待间隔.最小值为10秒

}

设置图像定时拍照模式

## FT\_E21Ins

{

FAssetID: 设备ID

FOpCode: 固定4

FOperationType: 1为设置,0为查询

FCameraID: 当前接入的摄像头个数.最大值为6,即表示最多只允许接入6个摄像头

FSize:

值 分辨率

1 160 \* 128

2 320 \* 240

3 640 \* 480

FClockUpCount: 需要上传多少张图片

FClockUpInterval: 图片之间的间隔,以秒为单位,最小为10秒.最大可为65534

FCount: 有多少个时刻点,每一个时刻点为数值,表示从0时开始的分钟数,如480,则表示早上的8:00整,495则表示8:15分.最多不能超过30个

FSetPara: 设定的分钟值,如480,720等,表示从0点开始的分钟数.最多不能超过30个

}

## FInsType= E22 (计价器外设指令集)

车辆状态

## FT\_E22Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FStatus: Vacant/Hired: 0(Vacant)或者1(Hired) 车辆状态(空载/重载)

}

## FInsType= E25 (双向通话外设指令集)

设置电话号码

## FT\_E25Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 1为设置,0为查询

FNUM1: 第1个电话号码

FNUM2: 第2个电话号码

}

## FInsType= E29 (CANBUS外设指令集)

设置二进制实时数据的上传

## FT\_E29Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 1为设置,0为查询

FBinfalg: 是否上传二进制数据, 1为上传, 0为取消上传  
}

FInsType= E30 (EPSILON油位传感器外设指令集)

读取EPSILON油位传感器的供电电压

FT\_E30Ins

{

FAssetID: 设备ID

FOpCode: 固定1

FOperationType: 1为设置,0为查询

}

设置EPSILON油位传感器的报警门限值

FT\_E30Ins

{

FAssetID: 设备ID

FOpCode: 固定2

FOperationType: 1为设置,0为查询

FVolumeChange: 变化的体积报警门限值(单位是升),最小为10, 小于10时,设置为10;建议最少为总油的3%

FReleaseTime: 报警时间判断门限值(以分钟为单位,最小10分钟,最大值为60,小于十分钟的设置为10分钟,大于60分钟的设置为60分钟)

}

开启加油/漏油报警

FT\_E30Ins

{

FAssetID: 设备ID

FOpCode: 固定3

FOperationType: 1为设置,0为查询

FOpenAddFuelAlarm: 加油报警开关,1为开启,0为关闭; 默认开启

FOpenLeakFuelAlarm: 漏油报警开关,1为开启,0为关闭;默认开启

}

设置EPSILON油位传感器对应邮箱体积

FT\_E30Ins

{

FAssetID: 设备ID

FOpCode: 固定4

FOperationType: 1为设置,0为查询

FCount: 油箱个数

FVolumes

{

Port: 端口号,需要设置的EPSILON油位传感器所在的端口号

Volume: 油箱体积(最大是65535)

```

}
Port: 端口号,需要设置的EPSILON油位传感器所在的端口号
Volume: 油箱体积(最大是65535)
}
Port: 端口号,需要设置的EPSILON油位传感器所在的端口号
Volume: 油箱体积(最大是65535)
}
}
}

```

设置EPSILON油位传感器的报警门限值(区分运动与静止)

```

FT_E30Ins
{
FAssetID: 设备ID
FOpCode: 固定5
FOperationType: 1为设置,0为查询
FVolumeChange: 变化的体积报警门限值(单位是升),最小为10, 小于10时,设置为10;建议最少为总油的3%
FMotionVolumeChange: 车辆运动时变化的体积报警门限值(单位是升),最小为10, 小于10时,设置为10;建议最少为总油的3%
FReleaseTime: 报警时间判断门限值(以分钟为单位,最小10分钟,最大值为60,小于十分钟的设置为10分钟,大于60分钟的设置为60分钟)
}

```

FInsType= E37 (EPSILON油位传感器外设指令集)

设置油箱高度

```

FT_E37Ins
{
FAssetID: 设备ID
FOpCode: 固定1
FOperationType: 1为设置,0为查询
FHigh: 表示为油箱高度为x毫米
}

```

## “X”系列万能端口设置指令

FInsType= X01 (查询及设置万能端口的使用模式)

```

FT_X01Ins
{
FAssetID: 设备ID
FOperationType: 1为设置,0为查询
FRunMode: 运行模式,共有以下几种模式:
注意: 目前只支持FUEL MODE及AD MODE.
ID 名称 说明
1 IO MODE IO模式,设置此模式之后,默认为输入.

```

2 AD MODE AD模式

3 PLUSE MODE 脉冲模式

}

# JT/T808设备类型指令类型对象

## JT/T808设备类型指令类型对象:

FInsType=Heart（终端心跳间隔）

```
FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FInterval:终端心跳发送间隔,单位为秒(s)
}
```

FInsType=MsgResp（消息应答）

```
FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FTCPResponseTimeout:TCP消息应答超时时间,单位为秒(s)
  FTCPRetransmission:TCP消息重传次数
  FUDPResponseTimeout:UDP消息应答超时时间,单位为秒(s)
  FUDPRetranmission:UDP消息重传次数
  FSMSResponseTimeout:SMS消息应答超时时间,单位为秒(s)
  FSMSRetranmission:SMS消息重传次数
}
```

FInsType=Master（主服务器网络）

```
FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FAPN:服务器APN,无线通信拨号访问点。若网络制式为CDMA,则该处为PPP拨号
  号码
  FUserName:服务器无线通信拨号用户名
  FPassword:服务器无线通信拨号密码
  FIp:服务器地址,IP或域名
  FTcpPort:服务器TCP端口
  FudpPort:服务器UDP端口
}
```

FInsType=Backup（备用服务器网络）

**FT\_JT808Ins**

```
{
  FOperationType:操作类型(1:设置;0:查询)
  FAPN: 服务器 APN, 无线通信拨号访问点。若网络制式为 CDMA, 则该处为 PPP 拨号
  号码
  FUserName: 服务器无线通信拨号用户名
  FPassword: 服务器无线通信拨号密码
  FIp: 服务器地址, IP 或域名
  FTcpPort: 服务器 TCP 端口
  FudpPort: 服务器 UDP 端口
}
```

FInsType= MasterIC (道路运输证IC卡认证主服务器)

**FT\_JT808Ins**

```
{
  FOperationType:操作类型(1:设置;0:查询)
  FIp: 道路运输证 IC 卡认证主服务器 IP 地址或域名
  FTcpPort: 道路运输证 IC 卡认证主服务器 TCP 端口
  FudpPort: 道路运输证 IC 卡认证主服务器 UDP 端口
}
```

FInsType= BackupIC (道路运输证IC卡认证备用服务器)

**FT\_JT808Ins**

```
{
  FOperationType:操作类型(1:设置;0:查询)
  FIp: 道路运输证 IC 卡认证主服务器 IP 地址或域名
  FTcpPort: 道路运输证 IC 卡认证主服务器 TCP 端口
  FudpPort: 道路运输证 IC 卡认证主服务器 UDP 端口
}
```

FInsType= Location (定位信息汇报)

**FT\_JT808Ins**

```
{
  FOperationType:操作类型(1:设置;0:查询)
  FReportDataType: 0: 定时汇报; 1: 定距汇报; 2: 定时和定距汇报
  FReportPlan: 0: 根据 ACC 状态; 1: 根据登录状态和 ACC 状态, 先判断登录状态,
  若登录再根据 ACC 状态

  FNotLoginReportInterval: 驾驶员未登录汇报时间间隔, 单位为秒 (s), >0
  FSleepReportInterval: 休眠时汇报时间间隔, 单位为秒 (s), >0
}
```

```

FAlarmReportInterval：紧急报警时汇报时间间隔，单位为秒（s），>0
FReportInterval：缺省时间汇报间隔，单位为秒（s），>0
根据FReportDataType传入上面或者下面颜色标记的字段
FReportMileage：缺省距离汇报间隔，单位为米(m),>0
FNotLoginReportMileage：驾驶员未登录汇报距离间隔，单位为米（m），>0
FSleepReportMileage：休眠时汇报距离间隔，单位为米（m），>0
FAlarmReportMileage：紧急报警时汇报距离间隔，单位为米（m），>0
FSupplementaryOfAngle：拐点补传角度，< 180
}

```

FInsType= Phone（电话号码设置）

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置；0：查询)
  FMonitorPhoneNumber：监控平台电话号码
  FResetPhoneNumber：复位电话号码，可采用此电话号码拨打终端电话让终端复位
  FRestoreFactoryPhoneNumber：恢复出厂设置电话号码，可采用此电话号码拨打终端电
话让终端恢复出厂设置
  FMonitorSmsPhoneNumber：监控平台 SMS 电话号码
  FSmsAlarmPhoneNumber：接收终端 SMS 文本报警号码
  FListeningStrategy：终端电话接听策略，0：自动接听；1：ACC ON 时自动接听，OF
F 时手动接听
  FMaximumTalkTimeOnce：每次最长通话时间，单位为秒（s），0 为不允许通话，0xFF
FFFFFF 为不限制
  FMaximumTalkTimeMonth：当月最长通话时间，单位为秒（s），0 为不允许通话，0xFF
FFFFFF 为不限制
  FMonitoringPhoneNumber：监听电话号码
  FMonitorVIPPhoneNumber：监管平台特权短信号码
}

```

FInsType= OverSpeed（超速报警参数设置）

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置；0：查询)
  FMaxSpeed：最高速度，单位为公里每小时（km/h）
  FOverSpeedDuration：超速持续时间，单位为秒（s）
  FOverSpeedAlarmDifference：超速报警预警差值，单位为 1/10Km/h
}

```

FInsType= Driving（超时驾驶报警参数设置）

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FFatigueThreshold:连续驾驶时间门限,单位为秒(s)
  FDrivingTimeDay:当天累计驾驶时间门限,单位为秒(s)
  FMinRestTime:最小休息时间,单位为秒(s)
  FFatigueAlarmDifference:疲劳驾驶预警差值,单位为秒(s),>0
}

```

FInsType= Parking (超时停车报警参数设置)

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FMaxParkingTime:最长停车时间,单位为秒(s)
}

```

FInsType= Collision (碰撞报警参数设置)

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FAcceleration:碰撞加速度,单位0.1g,设置范围在:0-79之间,默认为10。(高字节)
  FCollisionTime:碰撞时间,单位4ms;(低字节)
}

```

FInsType= Rollover (侧翻报警参数设置)

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FRolloverAngle:侧翻角度,单位1度,默认为30度
}

```

FInsType= Photo (拍照控制)

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FTimingControl:定时拍照控制
  FDistanceControl:定距拍照控制
  FPictureQuality:图像/视频质量,1-10,1最好图像/视频质量,1-10,1最好图像
}

```

```

/视频质量, 1-10, 1 最好
  FBrightness: 亮度
  FContrastRatio: 对比度
  FSaturation: 饱和度
  FChroma: 色度
}

```

FInsType= Vehicle (车辆信息设置)

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  MileageValue: 车辆里程表读数, 1/10km
  FProvinceId: 车辆所在的省域 ID
  FCityId: 车辆所在的市域 ID
  FLicensePlate: 公安交通管理部门颁发的机动车号牌
  FLicenseColor: 车牌颜色, 按照 JT/T415-2006 的 5.4.12(参见表2)
}

```

FInsType= GNSS (GNSS设置)

```

FT_JT808Ins
{
  FOperationType:操作类型(1:设置;0:查询)
  FLocationType: GNSS 定位模式
  /// bit0, 0:禁用 GPS 定位, 1:启用 GPS 定位;
  /// bit1, 0:禁用北斗定位, 1:启用北斗定位;
  /// bit2, 0:禁用 GLONASS 定位, 1:启用 GLONASS 定位;
  /// bit3, 0:禁用 Galileo 定位, 1:启用 Galileo 定位。
  如果只启动GPS则传入 $2^0=1$ 
  如果只启动北斗则传入 $2^1=2$ 
  如果GPS与北斗都启动则为 $2^0+2^1=3$ 
  依次类推...
  FBaudRate: GNSS 波特率, 定义如下
  /// 0x00: 4800; 0x01: 9600;
  /// 0x02: 19200; 0x03: 38400;
  /// 0x04: 57600; 0x05: 115200。
  FModelDetailLocationInfo: GNSS 模块详细定位数据输出频率, 定义如下:
  /// 0x00: 500ms; 0x01: 1000ms (默认值);
  /// 0x02: 2000ms; 0x03: 3000ms;
  /// 0x04: 4000ms。
  FAcquisitionFrequency: GNSS 模块详细定位数据采集频率, 单位为秒, 默认为 1。
  FUploadType: GNSS 模块详细定位数据上传方式
  /// 0x00, 本地存储, 不上传 (默认值);
  /// 0x01, 按时间间隔上传;

```

```

    /// 0x02, 按距离间隔上传;
    /// 0x0B, 按累计时间上传, 达到传输时间后自动停止上传; (传入数值11)
    /// 0x0C, 按累计距离上传, 达到距离后自动停止上传; (传入数值12)
    /// 0x0D, 按累计条数上传, 达到上传条数后自动停止上传(传入数值13)。
}

```

FInsType= FOTA (无线升级)

```

FT_JT808Ins
{
    FUrl: URL 地址;
    FAPN: 拨号点 名称;
    FUserName: 拨号用户名;
    FPassword: 拨号密码;
    FIp: 地址;
    FTcpPort: TCP 端口;
    FudpPort: UDP 端口;
    FManufacturerID: 制造商 ID;
    FHardwareVersion: 硬件版本;
    FSoftwareVersion: 固件版本;
    FConnectionTimeLimit: 连接到指定服务器时限
}

```

FInsType= Control (控制终端连接指定服务器)

```

FT_JT808Ins
{
    FControlType: 连接控制;
    FAuthCode: 监管平台鉴权码
    FAPN: 拨号点 名称;
    FUserName: 拨号用户名;
    FPassword: 拨号密码;
    FIp: 地址;
    FTcpPort: TCP 端口;
    FudpPort: UDP 端口;
    FConnectionTimeLimit: 连接到指定服务器时限
}

```

FInsType= Shutdown (终端关机)

FInsType= Reset (终端复位)

FInsType= Factory (终端恢复出厂设置)

FInsType= TODC (关闭数据通信)

FInsType= TOAWC（关闭所有无线通信）

FInsType= Attribute（查询终端属性）

FInsType= SGBK（发送GBK指令）

```
FT_JT808Ins
{
  FCommand:指令内容
}
```

FInsType= Abnormal（油位异常报警参数设置）

```
FT_JT808Ins
{
  FFuelMode::模式 00=静止, 01 = 运动
  FFuelLevel:油量为15升(异常油位变化最小设置限值为5L)
  FFuelTimeInterval:时间5分钟（注：最小时间间隔为5分钟，最大值为60分钟，0分钟
表示关闭该功能）
}
```

FInsType= Parameter（查询终端参数）

FInsType= SensorParam（查询油位传感器关键参数）

FInsType= SensorDamp（查询和设置油位传感器阻尼）

```
FT_JT808Ins
{
  FOperationType:操作类型(1:设置; 0: 查询)
  FDamp:阻尼, 单位是秒, 范围是 5 到 600;
}
```

FInsType= Volume（查询/设置油箱体积）

```
FT_JT808Ins
{
  FOperationType:操作类型(1:设置; 0: 查询)
  FVolume:油箱体积, 表示为 400L. 若为 60L, 则为 00060, 最大值为 09999
}
```

FInsType= Version（查询终端版本号）



# 自定义类型说明

## 设备类型：

设备ID	设备名称
100	JTR100
201	JT301A
400	GP4000
500	JT300B
501	JT704-04
502	JT300B-04
503	JT300D-02
504	JT300D-04
551	JT300DB
552	GP2000
553	GP7000
600	GP6000
601	GP6000F
602	JT600C
700	JT700DT
701	JT701
702	JT702
703	JT703
704	JT704
705	JT705

706	JT706
707	JT707A
709	JT709
802	JT802
808	JT/T808
1002	RTCC
1010	ConcoxTrack
1011	ConcoxTrack(MQTT)
1020	Skycniot(GT60)
1021	Skycniot(TX60)
1030	MeiTrack
1040	Queclink
1050	Eelink
1060	EasyTrack
1070	CanTrack
1080	TopTen
1090	KHDTrack
1100	BiidiWatch
1110	SeeWord(S208)
1111	SeeWord(S112U)
1120	AnyTrack
1130	HCIOT
1140	JTS6SA
1150	AOVX-A/G

1151	AOVX-V
1160	Teltonika
1170	JT300T1
1200	MiWiTrack
1301	JT301B
1600	eLock
1701	JT701-TRA
1704	JT704H
1705	JT705A
1707	JT707B
1709	JT709A
1808	CB-2019
2000	UDP
2301	JT301C
2701	JT701D(MQTT)
2705	JT705A_RH
2707	JT707C
2709	JT709B
2808	JT/T 808-2019
3301	JT301D
3701	JT701D
3709	JT709C
4701	Vietnam701
4709	JT709C(DMP)

5701	JT701D_BLE
5709	JT709N
6701	JT701(HWIOT)
6709	JT709C(MQTT)

## 报警类型：

报警类型	报警名称
1	超速报警
2	疲劳驾驶
3	危险预警
4	GNSS模块发生故障
5	GNSS天线未接或被剪断
6	GNSS天线短路
7	终端主电源欠压
8	终端主电源掉电
9	终端LCD或显示器故障
10	TTS模块故障
11	摄像头故障
12	道路运输证IC卡模块故障
13	超速预警
15	断电报警
19	超时停车
23	路线偏离报警
24	车辆VSS故障
26	车辆被盗

27	车辆非法点火
28	车辆非法位移
29	碰撞预警
30	侧翻预警
31	非法开门报警
32	视频信号丢失报警
33	视频信号遮挡报警
34	存储单元故障报警
35	其他视频设备故障报警
36	客车超员报警
37	异常驾驶行为报警
38	特殊报警录像达到存储阈值报警
40	锁绳剪断
41	震动
42	长时间开锁
43	开锁密码连续5次错误
44	刷非法卡
45	低电量
46	开后盖
47	卡锁
48	进区域报警
49	出区域报警
50	启用后备电池
51	SOS

52	拖吊报警
54	油位报警
55	进热点报警
56	出热点报警
57	进道路报警
58	出道路报警
63	温湿度报警
64	前向碰撞报警
65	车道偏离报警
66	车距过近报警
67	行人碰撞报警
68	频繁变道报警
69	道路标识超限报警
70	障碍物报警
71	道路标志识别事件
72	主动抓拍事件
73	实线变道报警
74	车厢过道行人监测报警
92	疲劳驾驶报警
93	接打电话报警
94	抽烟报警
95	分神驾驶报警
96	驾驶员异常报警
97	自动抓拍事件

98	驾驶员变更事件
99	探头遮挡报警
100	超时驾驶报警
101	未系安全带报警
102	红外阻断型墨镜失效报警
103	双脱把报警
104	玩手机报警
110	胎压报警
125	后方接近报警
126	左侧后方接近报警
127	右侧后方接近报警
131	急加速报警
132	急减速报警
133	急转弯报警
134	怠速报警
135	异常熄火报警
136	空挡滑行报警
137	发动机超转报警
141	超速报警
145	超过车辆额定载重报警
146	超过道路承重报警
151	限高报警
160	安全带报警
161	紧急刹车报警

162	空挡滑行
163	GPRS重连
164	调度屏连接
165	调度屏断开
166	CANBUS断开连接报警
167	CANBUS故障码上传报警
168	限制开车报警
180	主机拆卸
190	上盖破坏
191	Gps天线干扰
192	低电量休眠报警
193	锁异常
194	开锁密码错误
195	没有定位不执行开锁
196	围栏外禁止开锁
197	姿态报警
198	从机信号丢失
199	监管状态禁止开锁
200	主电池更换
201	自检异常
202	存储空间不足
203	锁条弯曲
204	锁条锯断
206	锁条丢失

316	光感报警
332	气压报警
400	超时怠速报警
402	围栏超时停留报警
403	出围栏未上锁报警
404	晚发预警
405	晚发报警
406	晚到预警
407	晚到报警
408	非法卸油
409	非法装油
410	LoRa通讯异常
10000	自定义报警

## 事件类型：

事件类型	事件名称
1	ACC开
2	ACC关
3	开锁
4	关锁
5	子锁开锁
6	子锁关锁
12	设备离线
17	充电上报
18	车辆保养

19	按键唤醒
20	拔出锁绳
21	阀门关闭
22	阀门打开
24	钥匙解封
25	钥匙施封
30	NFC触发
31	开箱
32	关箱
33	出区域施封
34	进区域解封
35	按下解锁键
36	仪表解锁
37	仪表锁定
38	启动加料
39	结束加料
40	蓝牙施封
41	蓝牙解封
42	远程施封
43	远程解封
44	抓拍
45	定时拍
46	离开起点
47	到达目的地

48	离开目的地
49	完成运单
50	进入途经点
51	离开途经点
52	进闸
53	出闸
54	合法卸油
55	合法装油
68	进入起点
10000	自定义事件

## 开锁类型：

开锁类型	开锁名称
1	刷卡开锁
2	远程开锁
3	蓝牙开锁
4	剪绳开锁
5	短信开锁
6	按键开锁
7	NFC开锁

## 开关锁状态：

类型	名称
1	开
0	关



# 运单管理

# 启动/结束监管

## 简要描述：

- 监管启动分析：运单开始进行启动分析，监控车辆状态（车开出起点围栏则变为已发车）
- 启动监管：车次派发后，开始启动监管分析，即运单状态直接变为已发车
- 结束监管：结束启动监管分析，即运单状态直接变为已结束

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-10-14	2020-10-14

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (SendCommandToRouteAnalysis )
FAssetID	是	string	设备号
FType	是	int	0: 监管启动分析 1: 启动监管 2: 结束监管

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": null
}
```

### 错误时返回:

```
//表示传入的参数有问题
{
  "Result": 102,
  "Message": "param is error",
  "FObject": null
}
//表示有异常报错,参数条件不同, message也会不同
{
  "Result": 105,
  "Message": "start analysis task fail" / "start analysis fail" / "check fail",
  "FObject": null
}
//表示找不到此设备
{
  "Result": 106,
  "Message": "device does not exist",
  "FObject": null
}
//表示找不到设备相关联的运单
{
  "Result": 121,
  "Message": "no task found",
  "FObject": null
}

//表示已启动分析或者启动监管,无需再次启动
{
  "Result": 135,
  "Message": "task started" ,
  "FObject": null
}
//表示分析中找不到此任务,需要等待分析程序初始化
```

```

{
  "Result": 134,
  "Message": "task is being initialized",
  "FObject": null
}

//表示启动监管前需要启动分析
{
  "Result": 136,
  "Message": "analysis task is not started" ,
  "FObject": null
}

```

## 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误
- 105: 表示有异常报错
- 106: 表示找不到此设备号
- 121: 表示找不到设备相关联的运单
- 134: 表示分析中找不到此任务, 需要等待分析程序初始化
- 135: 表示已启动分析或者启动监管, 无需再次启动
- 136: 表示启动监管前需要启动分析

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
String body = "{FAction:\\"SendCommandToRouteAnalysis\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAssetID:\\"13301781203\\",FType: 0}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");

```

```

// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"SendCommandToRouteAnalysis\", FTokenID: \"3acef045-
d302-4032-b40a-d9ee6c1519cd\", FAssetID: \"13301781203\", FType: 0}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'SendCommandToRouteAnalysis',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAssetID': '13301781203',
    'FType': '0',
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                 r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 车次派发

## 简要描述：

- 车次派发用于车辆派发运输任务

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (AddPublicRouteDeparture)
FName	否	string	公司名称，此字段传空时， 则默认为当前用户所属公司

FListRouteDeparture	是	List<MT_RouteDeparture>	运单内容,可派发
---------------------	---	-------------------------	----------

**MT\_RouteDeparture 对象说明:**

参数名	是否必须	类型	说明
FVehicleName	是	string	车牌号
FAssetID	是	string	设备号
FRouteName	是	string	路线名称
FPlanStartTime	是	DateTime	计划开始时间（UTC时间）
FPlanEndTime	是	DateTime	计划结束时间（UTC时间）
FWaybill	是	string	订单号
FGoods	是	string	货物名称
FCarrier	是	string	承运商
FDriverName	是	string	司机名
FContainerNo	是	string	货箱
FCargoDescribe	是	string	货物描述
FConsignee	是	string	收货人
FFinishType	是	int	完成规则 0: 出目的地完成 1: 到达目的地完成 2: 目的地内开锁完成 5:回到起点完成
FDepartureType	是	int	运单类型 0: 普通单 1: 计划单
FPlannedDuration	是	int	计划单提前时间（分钟）
FIsLbsAnalysis	是	string	是否启用基站定位分析 1: 启用 0: 不启用，默认启用

**返回示例:****正确时返回:**

```

{
  "Result": 200,
  "Message": "success",
  "FObject": [
    {
      "RowNo": 0, //对应传入对象 ( FListRouteDeparture ) 的序号
      "FWaybillGUID": "d0ddcad6-c5d6-4876-8174-f97e3956f72e" //该运单
的唯一标识
    }
  ]
}

```

### 错误时返回:

```

//表示传入的对象为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FListRouteDeparture is null",
  "FObject": null
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": null
}
//表示传入的车牌号或者设备号有空数据
{
  "Result": 106,
  "Message": "FVehicleName or FAssetID has empty data",
  "FObject": null
}

//表示传入数据中的车牌号有重复的
{
  "Result": 107,
  "Message": "FVehicleName has duplicate data:" + 重复的车牌号，多个以逗号
隔开，
  "FObject": null
}
//表示传入数据中的设备号有重复的
{
  "Result": 108,
  "Message": "FVehicleName has duplicate data:" + 重复的设备号，多个以逗号
隔开，

```

```
"FObject": null
}

//表示此公司在当前用户下找不到，建议登录平台查看
{
  "Result": 109,
  "Message": "FAgentName has no exists data:" + 公司名称,
  "FObject": null
}

//表示这些设备号不存在，或者不在该公司下
{
  "Result": 110,
  "Message": "FAssetID has no exists data:" + 不存在的设备号，多个以逗号隔开,
  "FObject": null
}

//表示这些设备号有正在执行中的，建议到平台上结束运单后再派发
{
  "Result": 111,
  "Message": "FAssetID has running data:" + 正在执行任务的的设备号，多个以逗号隔开,
  "FObject": null
}

//表示这些车牌号有正在执行中的，建议到平台上结束运单后再派发
{
  "Result": 112,
  "Message": "FVehicleName has running data:" + 正在执行任务的的车牌号，多个以逗号隔开,
  "FObject": null
}

//表示这些路线名称不存在或者不在该公司下，建议到平台上确认下
{
  "Result": 113,
  "Message": "FRouteName has no exists data:" + 不存在的路线，多个以逗号隔开,
  "FObject": null
}
```

## 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误

- 105: 表示有异常报错
- 106: 表示传入的车牌号或者设备号有空数据
- 107: 表示传入数据中的车牌号有重复的
- 108: 表示传入数据中的设备号有重复的
- 109: 表示此公司在当前用户下找不到, 建议登录平台查看
- 110: 表示这些设备号不存在, 或者不在该公司下
- 111: 表示这些设备号有正在执行中的, 建议到平台上结束运单后再派发
- 112: 表示这些车牌号有正在执行中的, 建议到平台上结束运单后再派发
- 113: 表示这些路线名称不存在或者不在该公司下, 建议到平台上确认下

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"AddPublicRouteDeparture\\",FName:\\"S\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FListRouteDeparture": "[{\\"FVehicleName\\":\\"13301781203\\",\\"FWaybill\\":\\"6757787\\",\\"FGoods\\":\\"223434\\",\\"FAssetID\\":\\"13301781203\\",\\"FRouteGUID\\":\\"42bfae96-cbed-4d2e-8e9f-c1cd4ad0d5df\\",\\"FRouteName\\":\\"43534\\",\\"FCarrier\\":\\"656546\\",\\"FDriverGUID\\":\\"\\",\\"FDriverName\\":\\"435435\\",\\"FDriverPhone\\":\\"6546\\",\\"FPlanStartTime\\":\\"2022-09-22 16:00:00\\",\\"FPlanEndTime\\":\\"2022-09-22 16:00:00\\",\\"FAgentGUID\\":\\"d6de6745-9b7b-42cd-b624-b4659f5f5ce5\\",\\"FRouteCode\\":\\"20220622103851\\",\\"FVehicleTypeID\\":\\"\\",\\"FVehicleImgType\\":\\"0\\",\\"FContainerNo\\":\\"6422f\\",\\"FCargoDescribe\\":\\"gfdgd\\",\\"FConsignee\\":\\"retgfd\\",\\"FFinishType\\":1}],{\\"FVehicleName\\":\\"13301781203\\",\\"FWaybill\\":\\"6757787\\",\\"FGoods\\":\\"223434\\",\\"FAssetID\\":\\"13301781203\\",\\"FRouteGUID\\":\\"42bfae96-cbed-4d2e-8e9f-c1cd4ad0d5df\\",\\"FRouteName\\":\\"43534\\",\\"FCarrier\\":\\"656546\\",\\"FDriverGUID\\":\\"\\",\\"FDriverName\\":\\"435435\\",\\"FDriverPhone\\":\\"6546\\",\\"FPlanStartTime\\":\\"2022-09-22 16:00:00\\",\\"FPlanEndTime\\":\\"2022-09-22 16:00:00\\",\\"FAgentGUID\\":\\"d6de6745-9b7b-42cd-b624-b4659f5f5ce5\\",\\"FRouteCode\\":\\"20220622103851\\",\\"FVehicleTypeID\\":\\"\\",\\"FVehicleImgType\\":\\"0\\",\\"FContainerNo\\":\\"6422f\\",\\"FCargoDescribe\\":\\"gfdgd\\",\\"FConsignee\\":\\"retgfd\\",\\"FFinishType\\":1}]}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
```

```

conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\\"AddPublicRouteDeparture\\",FAction:\\"AddPublicRout
eDeparture\\",FName:\\"S\\",FListRouteDeparture":["{\\"FVehicleName\\":\\"133017
81203\\",\\"FWaybill\\":\\"6757787\\",\\"FGoods\\":\\"223434\\",\\"FAssetID\\":\\"13301
781203\\",\\"FRouteGUID\\":\\"42bfae96-cbed-4d2e-8e9f-c1cd4ad0d5df\\",\\"FRouteNa
me\\":\\"43534\\",\\"FCarrier\\":\\"656546\\",\\"FDriverGUID\\":\\"\\",\\"FDriverName\\
:\\435435\\",\\"FDriverPhone\\":\\"6546\\",\\"FPlanStartTime\\":\\"2022-09-22 16:00
:00\\",\\"FPlanEndTime\\":\\"2022-09-22 16:00:00\\",\\"FAgentGUID\\":\\"d6de6745-9b
7b-42cd-b624-b4659f5f5ce5\\",\\"FRouteCode\\":\\"20220622103851\\",\\"FVehicleTyp
eID\\":\\"\\",\\"FVehicleImgType\\":\\"0\\",\\"FContainerNo\\":\\"6422f\\",\\"FCargoDes
cribe\\":\\"gfdgd\\",\\"FConsignee\\":\\"retgfd\\",\\"FFinishType\\":1},{\\"FVehicleN
ame\\":\\"13301781203\\",\\"FWaybill\\":\\"6757787\\",\\"FGoods\\":\\"223434\\",\\"FAss
etID\\":\\"13301781203\\",\\"FRouteGUID\\":\\"42bfae96-cbed-4d2e-8e9f-c1cd4ad0d5d
f\\",\\"FRouteName\\":\\"43534\\",\\"FCarrier\\":\\"656546\\",\\"FDriverGUID\\":\\"\\",\
"FDriverName\\":\\"435435\\",\\"FDriverPhone\\":\\"6546\\",\\"FPlanStartTime\\":\\"20
22-09-22 16:00:00\\",\\"FPlanEndTime\\":\\"2022-09-22 16:00:00\\",\\"FAgentGUID\\
:\\d6de6745-9b7b-42cd-b624-b4659f5f5ce5\\",\\"FRouteCode\\":\\"20220622103851\\
\\",\\"FVehicleTypeID\\":\\"\\",\\"FVehicleImgType\\":\\"0\\",\\"FContainerNo\\":\\"6422f
\\",\\"FCargoDescribe\\":\\"gfdgd\\",\\"FConsignee\\":\\"retgfd\\",\\"FFinishType\\":1
}]}";
Encoding encoding = Encoding.UTF8;

```

```

HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'AddPublicRouteDeparture',
    'FName': 'S',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FListRouteDeparture': '["FVehicleName": "13301781203", "FWaybill": "6757787", "FGoods": "223434", "FAssetID": "13301781203", "FRouteGUID": "42bfae96-cbed-4d2e-8e9f-c1cd4ad0d5df", "FRouteName": "43534", "FCarrier": "656546", "FDriverGUID": "", "FDriverName": "435435", "FDriverPhone": "6546", "FPlanStartTime": "2022-09-22 16:00:00", "FPlanEndTime": "2022-09-22 16:00:00", "FAgentGUID": "d6de6745-9b7b-42cd-b624-b4659f5f5ce5", "FRouteCode": "20220622103851", "FVehicleTypeID": "", "FVehicleImgType": "0", "FContainerNo": "6422f", "FCargoDescribe": "gfdgd", "FConsignee": "retgfd", "FFinishType": 1], {"FVehicleName": "13301781203", "FWaybill": "6757787", "FGoods": "223434", "FAssetID": "13301781203", "FRouteGUID": "42bfae96-cbed-4d2e-8e9f-c1cd4ad0d5df", "FRouteName": "43534", "FCarrier": "656546", "FDriverGUID": "", "FDriverName": "435435", "FDriverPhone": "6546", "FPlanStartTime": "2022-09-22 16:00:00", "FPlanEndTime": "2022-09-22 16:00:00", "FAgentGUID": "d6de6745-9b7b-42cd-b624-b4659f5f5ce5", "FRouteCode": "20220622103851", "FVehicleTypeID": "", "FVehicleImgType": "0", "FContainerNo": "6422f", "FCargoDescribe": "gfdgd", "FConsignee": "retgfd", "FFinishType": 1}]'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.

```

```
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
        'Connection': 'keep-alive'  
    }  
    req = request.Request(url, headers=headers, data=data)  
    page = request.urlopen(req).read()  
    page = page.decode('utf-8')  
    # json_array = json.loads(page)  
    return page;
```

# 查询公司车次列表 ( 未完成车次 )

## 简要描述：

- 查询公司车次列表(未完成车次)

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryDepartureListByFAgentGUID )
FAgentGUID	否	string	公司唯一标识
FStatus	是	int	运单状态(0:未发车;1:已发出;2:已到达 3:计划单)
FPageSize	否	int	分页查询 每页记录数
FPageIndex	否	int	分页查询 当前页数

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FTotalCount": 2,
        "FPlanCount": 0,
        "FWaitingCount": 2,
        "FRunningCount": 0,
        "FFinishCount": 0
      }
    ],
    "Table1": [
      {
        "FGUID": "58a297aa-0703-41f4-95c2-d41440d4631d",
        "FVehicleGUID": "c6d3ca9d-ec2f-4ce9-8e54-9c3b1ade194b",
        "FVehicleName": "791011001167",
        "FWaybill": "20230316-001",
        "FGoods": "日用品",
        "FAssetGUID": "c4b7730e-f74b-4dee-833b-0128bf5dd7f4",
        "FAssetID": "791011001167",
        "FRouteGUID": "c5ed320f-5eb8-4e89-91ab-f480060bd427",
        "FCarrier": "久通",
        "FDriverGUID": "00000000-0000-0000-0000-000000000000",
        "FDriverName": "xx",
        "FDriverPhone": "885252522",
        "FPlanStartTime": "2023-03-16T23:00:00",
        "FActualStartTime": null,
        "FPlanEndTime": "2023-03-17T00:00:00",
        "FActualEndTime": null,
        "FContainerNo": "20230316-001",
        "FConsignee": "霍万庆",
        "FVehicleTypeID": 1,
        "FCargoDescribe": "",
        "FVehicleImgType": 0,
        "FFinishType": 1,
        "FDepartureType": 1,
        "FPlannedDuration": 30,
        "FLocationStatus": 1,
      }
    ]
  }
}
```

```

"RowNo": 2,
"FRouteName": "JT-TEST运单02",
"FRouteCode": "20230314-02",
"FAssetTypeID": 1709
},
{
"FGUID": "aaf42ad9-5b26-4a79-ac1b-be9f1d53423d",
"FVehicleGUID": "b0d32ae5-77bf-4abe-abc-b-1b94c9f249c7",
"FVehicleName": "20221103017",
"FWaybill": "",
"FGoods": "",
"FAssetGUID": "c4a8a015-c645-437b-ad83-56ba65a9a449",
"FAssetID": "20221103017",
"FRouteGUID": "2619f283-89b9-488b-9c81-83c9ac56a4dd",
"FCarrier": "",
"FDriverGUID": "00000000-0000-0000-0000-000000000000",
"FDriverName": "",
"FDriverPhone": "",
"FPlanStartTime": "2023-04-11T16:00:00",
"FActualStartTime": null,
"FPlanEndTime": "2023-04-18T16:00:00",
"FActualEndTime": null,
"FContainerNo": "",
"FConsignee": "",
"FVehicleTypeID": 0,
"FCargoDescribe": "",
"FVehicleImgType": 0,
"FFinishType": 0,
"FDepartureType": 0,
"FPlannedDuration": 30,
"FLocationStatus": 1,
"RowNo": 1,
"FRouteName": "JT-TEST1",
"FRouteCode": "20230223",
"FAssetTypeID": 709
}
]
}
}

```

### 错误时返回:

```

//表示传入的公司的唯一标识为空，应该是数据格式不对
{
  "Result": 102,

```

```

    "Message": "FAgentGUID is null",
    "FObject": []
}
//表示token过期异常报错
{
    "Result": 104,
    "Message": "token error or timeout",
    "FObject": []
}
//表示有异常报错
{
    "Result": 105,
    "Message": "fail",
    "FObject": null
}

```

## 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误
- 104: token过期异常报错
- 105: 表示有异常报错

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\": \"QueryDepartureListByFAgentGUID\", \"FPageSize\": 50, \"FPageIndex\": 0, \"FStatus\": 0, \"FAgentGUID\": \"661E7209-C77E-4A54-AFB8-6D956F962BDF\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\", \"FTimeDifferent\": 28800, \"FLanguage\": \"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "/*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");

```

```

conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\": \"QueryDepartureListByFAgentGUID\", \"FPageSiz
e\": 50, \"FPageIndex\": 0, \"FStatus\": 0, \"FAgentGUID\": \"661E7209-C77E
-4A54-AFB8-6D956F962BDF\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2a
e741\", \"FTimeDifferent\": 28800, \"FLanguage\": \"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

```
}
```

## Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryDepartureListByFAgentGUID',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae741',
    'FAgentGUID': '661E7209-C77E-4A54-AFB8-6D956F962BDF',
    'FStatus': 0,
    'FPageSize': 50,
    'FPageIndex': 0,
    'FTimeDifferent': 28800,
    'FLanguage': 1
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                 r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 查询运单报告 ( 所有 )

## 简要描述：

- 按设备或车辆查询车次报告

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryReportRouteDepartureListByFGUIDs )
FGUIDs	是	string	车辆/设备唯一标识 (多个以逗号隔开)
FSelectType	是	int	1: 按车辆查; 0: 按终端查

FStatus	是	int	运单状态(-2:查询全部; -1: 作废; 0:查询未完成; 1: 查询已完成 2:已超期(实际到达时间大于计划完成时间表示已超期) 3: 计划单)
FStartTime	是	datetime	开始时间 (UTC时间)
FEndTime	是	datetime	结束时间 (UTC时间)
FTimeDifferent	是	int	时差 (秒)
FKey	否	string	按运单号模糊查询
FPageSize	否	int	分页查询 每页记录数
FPageIndex	否	int	分页查询 当前页数

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FTotalCount": 2
      }
    ],
    "Table1": [
      {
        "FGUID": "fa7cc011-21d5-4c78-bca1-b02871122314",
        "FVehicleGUID": "d7ca53d2-eb4c-4ab9-ba1a-9477becd4563",
        "FVehicleName": "KAA111A",
        "FWaybill": "",
        "FGoods": "",
        "FAssetGUID": "ac305cdc-24d0-4b2b-9581-69bf53fc1235",
        "FAssetID": "3454354353",
        "FRouteGUID": "7ed92857-8770-4351-9904-0d66d0933452",
        "FCarrier": "",
        "FDriverGUID": "00000000-0000-0000-0000-000000000000",
        "FDriverName": "",
        "FDriverPhone": "",
        "FPlanStartTime": "2024-12-03T06:02:00",
        "FActualStartTime": "2024-12-03T06:01:13.937",
      }
    ]
  }
}
```

```

"FPlanEndTime": "2024-12-03T21:00:00",
"FActualEndTime": null,
"FStatus": 4,
"FFinishTime": "2024-12-03T06:02:59.903",
"FAgentGUID": "0f0f2e76-4733-4388-9242-9ed7db004562",
"FCreateTime": "2024-12-03T06:00:54.07",
"FContainerNo": "",
"FCargoDescribe": "",
"FConsignee": "",
"FFinishType": 2,
"FDepartureType": 0,
"FPlannedDuration": 30,
"RowNo": 1,
"FRouteName": "a-c",
"FRouteCode": "20241017094717",
"FAgentName": "KPM Tech Solution",
"TransportationTime": "451",
"TransportationMileage": "0",
"ViewDetails": ""

```

```

    }

```

```

  ]

```

```

}

```

```

}

```

### 错误时返回:

//表示传入的公司的唯一标识为空，应该是数据格式不对

```

{
  "Result": 102,
  "Message": "FGUIDs is null",
  "FObject": []
}

```

//表示token过期异常报错

```

{
  "Result": 104,
  "Message": "token error or timeout",
  "FObject": []
}

```

//表示有异常报错

```

{
  "Result": 105,
  "Message": "fail",
  "FObject": null
}

```

**返回Table参数说明:**

参数名	类型	说明
FTotalCount	int	分页总数量

**返回Table1参数说明:**

参数名	类型	说明
FGUID	string	运单唯一标识
FVehicleGUID	String	车辆唯一标识
FVehicleName	string	车牌号
FAssetGUID	string	设备唯一标识
FAssetID	string	设备号
FRouteGUID	string	路线唯一标识
FRouteName	string	路线名称
FRouteCode	string	路线编码
FAgentGUID	string	公司唯一标识
FAgentName	string	公司名称
FPlanStartTime	DateTime	计划开始时间 (UTC时间)
FPlanEndTime	DateTime	计划结束时间 (UTC时间)
FActualStartTime	DateTime	实际发车时间 (UTC时间)
FActualEndTime	DateTime	实际到达时间 (UTC时间)
FFinishTime	DateTime	完成时间 (UTC时间)
FCreateTime	DateTime	创建时间 (UTC时间)
FWaybill	string	订单号
FGoods	string	货物名称

FCarrier	string	承运商
FDriverName	string	司机名
FDriverPhone	string	司机电话
FContainerNo	string	货箱
FCargoDescribe	string	货物描述
FConsignee	string	收货人
FFinishType	int	完成规则 0: 出目的地完成 1: 到达目的地完成 2: 目的地内开锁完成 5:回到起点完成
FDepartureType	int	运单类型 0: 普通单 1: 计划单
FPlannedDuration	int	计划单提前时间 (分钟)
FStatus	int	运单状态(-1: 作废; 0:未完成; 1:自动完成 2:已超期(实际到达时间大于计划完成时间表示已超期) 3: 计划单) 4: 手动完成
TransportationTime	int	耗时 (秒)
TransportationMileage	int	里程值
ViewDetails	string	运单详情链接

## 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误
- 104: token过期异常报错
- 105: 表示有异常报错

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
```

```

String body = "{\"FAction\": \"QueryReportRouteDepartureListByFGUIDs\", \"F
PageSize\": 50, \"FPageIndex\": 1, \"FStatus\": 0, \"FGUIDs\": \"661E7209-C
77E-4A54-AFB8-6D956F962BDF\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:0
0:00\", FEndTime: \"2020-08-25 15:59:59\", \"FTokenID\": \"ec51d691-d9a7-4f74
-95a8-1856be2ae741\", \"FTimeDifferent\": 28800, \"FLanguage\": \"1\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{\"FAction\": \"QueryReportRouteDepartureListByFGUIDs\", \"F
PageSize\": 50, \"FPageIndex\": 1, \"FStatus\": 0, \"FGUIDs\": \"661E7209-C
77E-4A54-AFB8-6D956F962BDF\", FSelectType: \"1\", FStartTime: \"2020-08-18 16:0
0:00\", FEndTime: \"2020-08-25 15:59:59\", \"FTokenID\": \"ec51d691-d9a7-4f74
-95a8-1856be2ae741\", \"FTimeDifferent\": 28800, \"FLanguage\": \"1\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

```

```

//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryReportRouteDepartureListByFGUIDs',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae741',
    'FGUIDs': 'BE2A1E27-B36E-4301-B2C8-465A62AD215F,FE0BDE75-4A30-4B9C-A
D2E-66D8A4F44F5E',
    'FSelectType':1,
    'FStartTime':'2020-08-05 16:00:00',
    'FEndTime':'2020-08-12 15:59:59',
    'FStatus':0,
    'FPageSize':50,
    'FPageIndex':1,
    'FTimeDifferent':28800,
    'FLanguage':1
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```



# 删除运单

## 简要描述：

- 删除车次

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (DeleteRouteDepartureByFGUIDs )
FGUIDs	是	string	车次唯一标识列表,多个唯一标识之间用逗号分隔 (例:"aaf42ad9-5b26-4a79-ac1b-be9f1d53423d,aaf42ad9-5b26-4a79-ac1b-be9f1d53423d)

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "succeed",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的车次唯一标识列表为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FGUIDs is null",
  "FObject": []
}
//表示token 过期
{
  "Result": 104,
  "Message": "token error or timeout",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": null
}
```

## 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
    \"FAction\": \"DeleteRouteDepartureByFGUIDs\",
    \"FGUIDs\": \"aaf42ad9-5b26-4a79-ac1b-be9f1d53423d\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{

```

```
\ "FAction\": \ "DeleteRouteDepartureByFGUIDs\ ",
\ "FGUIDs\": \ "aaf42ad9-5b26-4a79-ac1b-be9f1d53423d\ ",
\ "FTokenID\": \ "ec51d691-d9a7-4f74-95a8-1856be2ae745\ ",
\ "FTimeDifferent\": 28800,
\ "FLanguage\": \ "1\ "
}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
//返回的是json字符串
return reader.ReadToEnd();
}
```

# 修改运单信息

## 简要描述：

- 修改车次信息

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateRouteDepartureByFGUID)
FT_RouteDeparture	是	FT_RouteDeparture	运单内容

## FT\_RouteDeparture 对象说明：

参数名	是否必须	类型	说明
FGUID	是	string	运单唯一标识
FVehicleName	是	string	车牌号
FAssetGUID	是	string	设备唯一标识
FAssetID	是	string	设备号
FRouteGUID	是	string	路线唯一标识
FAgentGUID	是	string	公司唯一标识
FPlanStartTime	是	DateTime	计划开始时间（UTC时间）
FPlanEndTime	是	DateTime	计划结束时间（UTC时间）
FRouteName	是	string	路线名称
FWaybill	是	string	订单号
FGoods	是	string	货物名称
FCarrier	是	string	承运商
FDriverName	是	string	司机名
FContainerNo	是	string	货箱
FCargoDescribe	是	string	货物描述
FWaybill	是	string	订单号
FConsignee	是	string	收货人
FFinishType	是	int	完成规则 0: 出目的地完成 1: 到达目的地完成 2: 目的地内开锁完成 5:回到起点完成
FDepartureType	是	int	运单类型 0: 普通单 1: 计划单
FPlannedDuration	是	int	计划单提前时间（分钟）
FIsLbsAnalysis	是	string	是否启用基站定位分析 1: 启用 0: 不启用，默

FlsLbsAnalysis	是	string	认启用
----------------	---	--------	-----

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "Succeed",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的对象为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "Parameters are error",
  "FObject": null
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": null
}
//表示运单开始时间异常报错
{
  "Result": 126,
  "Message": "plan start time error",
  "FObject": null
}
```

## 备注:

- 更多返回错误代码如下:
- 106: 账户过期
- 103: 用户名或者密码错误
- 102: 请求参数错误
- 105: 表示有异常报错
- 126: 表示传入运单开始时间报错

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
    \"FAction\": \"UpdateRouteDepartureByFGUID\",
    \"FT_RouteDeparture\": {
        \"FGUID\": \"58a297aa-0703-41f4-95c2-d41440d4631d\",
        \"FVehicleName\": \"791011001167\",
        \"FAssetGUID\": \"c4b7730e-f74b-4dee-833b-0128bf5dd7f4\",
        \"FAssetID\": \"791011001167\",
        \"FRouteGUID\": \"c5ed320f-5eb8-4e89-91ab-f480060bd427\",
        \"FAgentGUID\": \"661E7209-C77E-4A54-AFB8-6D956F962BDF\",
        \"FPlanStartTime\": \"2023-03-16T23:00:00\",
        \"FPlanEndTime\": \"2023-03-17T00:00:00\"
    },
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
```

```

while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"UpdateRouteDepartureByFGUID\",
    \"FT_RouteDeparture\": {
        \"FGUID\": \"58a297aa-0703-41f4-95c2-d41440d4631d\",
        \"FVehicleName\": \"791011001167\",
        \"FAssetGUID\": \"c4b7730e-f74b-4dee-833b-0128bf5dd7f4\",
        \"FAssetID\": \"791011001167\",
        \"FRouteGUID\": \"c5ed320f-5eb8-4e89-91ab-f480060bd427\",
        \"FAgentGUID\": \"661E7209-C77E-4A54-AFB8-6D956F962BDF\",
        \"FPlanStartTime\": \"2023-03-16T23:00:00\",
        \"FPlanEndTime\": \"2023-03-17T00:00:00\"
    },
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'UpdateRouteDepartureByFGUID',
    'FT_RouteDeparture': {
        'FGUID': '58a297aa-0703-41f4-95c2-d41440d4631d',
        'FVehicleName': '791011001167',
        'FAssetGUID': 'c4b7730e-f74b-4dee-833b-0128bf5dd7f4',
        'FAssetID': '791011001167',
        'FRouteGUID': 'c5ed320f-5eb8-4e89-91ab-f480060bd427',
        'FAgentGUID': '661E7209-C77E-4A54-AFB8-6D956F962BDF',
        'FPlanStartTime': '2023-03-16T23:00:00',
        'FPlanEndTime': '2023-03-17T00:00:00'
    },
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
    'FTimeDifferent': 28800,
    'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                 r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 查询单个运单详情

## 简要描述：

- 按运单唯一标识查询单个运单详情

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryDepartureInfoByFGUID )
FGUID	是	string	运单唯一标识（多个以逗号隔开）

## 返回示例：

### 正确时返回：

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": [
    {
      "FGUID": "258dbdb5-1c01-495c-98f5-6165e9fa8b4c",
      "FVehicleGUID": "9e527040-b78c-4d88-8740-a2a90690a701",
      "FVehicleName": "796506000001",
      "FWaybill": "",
      "FGoods": "",
      "FAssetGUID": "cd72f9fe-1508-41b5-9457-81568890d447",
      "FAssetID": "796506000001",
      "FRouteGUID": "04d48d4b-dc20-4592-bf9d-ebd0271020f5",
      "FCarrier": "",
      "FDriverName": "",
      "FDriverPhone": "",
      "FPlanStartTime": "2025-07-31T06:36:08",
      "FActualStartTime": "2025-07-31T07:01:38.1",
      "FPlanEndTime": "2025-07-31T07:36:08",
      "FActualEndTime": null,
      "FContainerNo": "",
      "FConsignee": "",
      "FAgentGUID": "661e7209-c77e-4a54-afb8-6d956f962bdf",
      "FCargoDescribe": "",
      "FStatus": 4,
      "FCreateTime": "2025-07-31T06:43:01.583",
      "FFinishTime": "2025-07-31T08:43:24.773",
      "FFinishType": 1,
      "FRouteName": "海谷-天格01",
      "FRouteCode": "001",
      "FStartFence": "5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d",
      "FEndFence": "1b1fe658-01b9-4c9d-81fb-c879974173ff",
      "FAssetTypeID": 1709,
      "FStartFenceName": "海谷",
      "FEndFenceName": "天格",
      "FAssetTypeName": "JT709A",
      "FAgentName": "软件测试部"
    }
  ]
}

```

**错误时返回:**

```

//表示传入的公司的唯一标识为空，应该是数据格式不对
{

```

```

    "Result": 102,
    "Message": "FGUIDs is null",
    "FObject": []
  }
  //表示token过期异常报错
  {
    "Result": 104,
    "Message": "token error or timeout",
    "FObject": []
  }
  //表示有异常报错
  {
    "Result": 105,
    "Message": "fail",
    "FObject": null
  }
}

```

## 返回参数说明:

参数名	类型	说明
FGUID	string	运单唯一标识
FVehicleGUID	String	车辆唯一标识
FVehicleName	string	车牌号
FAssetGUID	string	设备唯一标识
FAssetID	string	设备号
FAssetTypeID	string	设备类型ID
FAssetTypeName	string	设备类型名称
FRouteGUID	string	路线唯一标识
FRouteName	string	路线名称
FRouteCode	string	路线编码
FAgentGUID	string	公司唯一标识
FAgentName	string	公司名称
FPlanStartTime	DateTime	计划开始时间（UTC时间）

FPlanEndTime	DateTime	计划结束时间（UTC时间）
FActualStartTime	DateTime	实际发车时间（UTC时间）
FActualEndTime	DateTime	实际到达时间（UTC时间）
FFinishTime	DateTime	完成时间（UTC时间）
FCreateTime	DateTime	创建时间（UTC时间）
FWaybill	string	订单号
FGoods	string	货物名称
FCarrier	string	承运商
FDriverName	string	司机名
FDriverPhone	string	司机电话
FContainerNo	string	货箱
FCargoDescribe	string	货物描述
FConsignee	string	收货人
FFinishType	int	完成规则 0：出目的地完成 1：到达目的地完成 2：目的地内开锁完成 5:回到起点完成
FDepartureType	int	运单类型 0：普通单 1：计划单
FPlannedDuration	int	计划单提前时间（分钟）
FStatus	int	运单状态(-1：作废； 0:未完成； 1:自动完成 2:已超期(实际到达时间大于计划完成时间表示已超期) 3：计划单) 4：手动完成
FStartFence	string	开始围栏唯一标识
FEndFence	string	结束围栏唯一标识
FStartFenceName	string	开始围栏名称
FEndFenceName	string	结束围栏名称

**备注:**

- 更多返回错误代码如下：
- 106：账户过期
- 103：用户名或者密码错误
- 102：请求参数错误
- 104：token过期异常报错
- 105：表示有异常报错

## 请求示例：

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\": \"QueryDepartureInfoByFGUID\", \"FGUID\": \"661E7209-C77E-4A54-AFB8-6D956F962BDF\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\": \"QueryDepartureInfoByFGUID\", \"FGUID\": \"661E7209-C77E-4A54-AFB8-6D956F962BDF\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryDepartureInfoByFGUID',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae741',
    'FGUID': 'BE2A1E27-B36E-4301-B2C8-465A62AD215F'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)

```

return page;

# 查询所有路线信息

## 简要描述：

- 查询路线列表

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryRoutePlanList)

## 返回示例：

### 正确时返回：

```
{
```

```

"Result": 200,
"Message": "check token success",
"FObject": [
  {
    "FRouteGUID": "794270c9-c74c-40c6-8e3f-00754ff5a28f",
    "FRouteName": "海谷-美国专线",
    "FRouteCode": "20231023163308",
    "FAgentGUID": "661e7209-c77e-4a54-afb8-6d956f962bdf",
    "FAgentName": "软件测试部"
  },
  {
    "FRouteGUID": "8bcf571b-ea0a-4cc7-a674-0326d9b6887a",
    "FRouteName": "充电柜测试",
    "FRouteCode": "22222",
    "FAgentGUID": "661e7209-c77e-4a54-afb8-6d956f962bdf",
    "FAgentName": "软件测试部"
  }
]
}

```

#### 错误时返回:

```

//表示传入的公司的唯一标识为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FParam is null",
  "FObject": []
}
//表示token过期异常报错
{
  "Result": 104,
  "Message": "token error or timeout",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": null
}

```

#### 返回参数说明:

--	--	--

参数名	类型	说明
FAgentGUID	String	公司唯一标识
FName	String	公司名
FRouteGUID	String	路线唯一标识
FRouteName	String	路线名称
FRouteCode	String	路线编码

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期异常报错
- 105: 表示有异常报错

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\": \"QueryRoutePlanList\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
```

```

pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
string body = "{\"FAction\": \"QueryRoutePlanList\", \"FTokenID\": \"ec51d
691-d9a7-4f74-95a8-1856be2ae741\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryDepartureListByFAgentGUID',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae741'
}
data = parse.urlencode(data).encode('utf-8')

```

```
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 查询单条路线具体信息

## 简要描述：

- 查询单条路线具体信息

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryRoutePlanByFGUID )
FGUID	是	string	路线唯一标识

## 返回示例：

### 正确时返回：

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Result": 200,
    "Message": "check token success",
    "FObject": {
      "Table": [
        {
          "FGUID": "e286fe13-2034-45eb-9da2-d929e8a2558a",
          "FName": "8789746",
          "FCode": "20200830145345",
          "FStartFence": "5b81e7f0-1e4c-4e4e-a4ba-32a7138caf03",
          "FStartFenceName": "海谷出围栏测试",
          "FStartFormType": 1,
          "FStartTypeName": "测试",
          "FStartPath": "[{"lat":22.67071837531768,"lng":113.9109
9242249336},{"lat":22.66875631707885,"lng":113.90927486128945},{"lat":
:22.66580899152102,"lng":113.91246440413853},{"lat":22.666602014375645,
"lng":113.91647473726114},{"lat":22.669011541033097,"lng":113.9171537
2235394},{"lat":22.671964991193125,"lng":113.91618420544815},{"lat":2
2.671898164421865,"lng":113.91389170362235}]",
          "FStartRadius": 0.0,
          "FStartCenterLon": 113.91365407695481,
          "FStartCenterLat": 22.668931600149243,
          "FEndFence": "38978700-bdc9-4857-bb0b-9826b59e83e4",
          "FEndFenceName": "测试围栏2",
          "FEndFormType": 1,
          "FEndTypeName": "测试",
          "FEndPath": "[{"lat":24.70994521713008,"lng":113.575092
78388251},{"lat":24.70994521713008,"lng":113.57657521611749},{"lat":2
4.711750782978523,"lng":113.57657521611749},{"lat":24.711750782978523,\
"lng":113.57509278388251}]",
          "FEndRadius": 0.0,
          "FEndCenterLon": 113.575834,
          "FEndCenterLat": 24.710848,
          "FRoadPoints": "[{"lat":22.66885928808014,"lng":113.913
63220848557},{"lat":22.668999470705682,"lng":113.91294237794823},{"lat
":22.669019771511334,"lng":113.91277192775806},{"lat":22.6696096144012
78,"lng":113.91290200213997},...,{"lat":24.70851446100423,"lng":113.5
7876801084377}]",
          "FRemark": ""
        }
      ],
      "Table1": [
        {

```



```

    "Message": "token error or timeout",
    "FObject": []
}
//表示有异常报错
{
    "Result": 105,
    "Message": "fail",
    "FObject": null
}

```

## 返回参数Table说明:

参数名	类型	说明
FAgentGUID	String	公司唯一标识
FGUID	String	路线唯一标识
FName	String	路线名称
FCode	String	路线名称
FStartFence	String	开始围栏唯一标识
FStartFenceName	String	开始围栏名称
FStartFormType	int	开始围栏类型 1: 多边形 2: 圆形
FStartTypeName	String	开始围栏类型名称
FStartPath	String	开始围栏点集合, Json字符串
FStartRadius	double	开始围栏半径
FStartCenterLon	String	开始围栏中心点经度
FStartCenterLat	String	开始围栏中心点纬度
FEndFence	String	结束围栏唯一标识
FEndFenceName	String	结束围栏名称
FEndFormType	int	结束围栏类型 1: 多边形 2: 圆形
FEndTypeName	String	结束围栏类型名称
FEndPath	String	结束围栏点集合, Json字符串

FEndRadius	double	结束围栏半径
FEndCenterLon	String	结束围栏中心点经度
FEndCenterLat	String	结束围栏中心点纬度
FRoadPoints	String	路线所有点集合，Json字符串
FRemark	String	路线描述

## 返回参数Table1说明：所有途经围栏，如果为null，表示该路线无途经围栏

参数名	类型	说明
FRouteGUID	String	所属路线唯一标识
FPassFenceGUID	String	途径围栏唯一标识
FName	String	途径围栏名称
FFormType	int	围栏类型 1：多边形 2：圆形
FTypeName	String	围栏类型名称
FPath	String	围栏点集合，Json字符串
FRadius	double	开始围栏半径
FCenterLon	String	开始围栏中心点经度
FCenterLat	String	开始围栏中心点纬度
FIndex	int	途径围栏序号

## 备注：

- 更多返回错误代码如下：
- 102：请求参数错误
- 104：token过期异常报错
- 105：表示有异常报错

## 请求示例：

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"QueryRoutePlanByFGUID\",\"FGUID\":\"e286fe13-2034-45eb-9da2-d929e8a2558a\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"QueryRoutePlanByFGUID\",\"FGUID\":\"e286fe13-2034-45eb-9da2-d929e8a2558a\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

```

```
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryRoutePlanByFGUID',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae741',
    'FGUID': 'e286fe13-2034-45eb-9da2-d929e8a2558a'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 查询单条路线信息,包含分支路线

## 简要描述：

- 查询单条路线具体信息,包含分支路线

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAllRoutePlanByFGUID )
FGUID	是	string	路线唯一标识

## 返回示例:

### 正确时返回:

```

{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FGUID": "1e8ea88a-60f3-42bc-8176-37c676b677d8",
        "FName": "李孝祯测试001",
        "FCode": "001",
        "FStartFence": "dfd0d889-7a01-4cd4-b8ed-88825c5fb72b",
        "FStartFenceName": "海谷科技大厦",
        "FStartFormType": 2,
        "FStartTypeName": "李孝祯测试",
        "FStartPath": "[]",
        "FStartRadius": 500.0,
        "FStartCenterLon": 113.92131186,
        "FStartCenterLat": 22.67073578,
        "FEndFence": "a156ffcf-9285-4756-95f1-0cb830fdae2a",
        "FEndFenceName": "龙海家园",
        "FEndFormType": 1,
        "FEndTypeName": "李孝祯测试",
        "FEndPath": "[{\"lat\":22.54519516,\"lng\":113.89590926},{\"lat\":22.53213027,\"lng\":113.89155722},{\"lat\":22.53244843,\"lng\":113.9106911},{\"lat\":22.54478113,\"lng\":113.90775015}]",
        "FEndRadius": 0.0,
        "FEndCenterLon": 113.90137618,
        "FEndCenterLat": 22.53813873,
        "FRoadPoints": "[{\"lat\":22.67123239,\"lng\":113.92147994},{\"lat\":22.67109185,\"lng\":113.92189991},{\"lat\":22.67097135,\"lng\":113.92223909},{\"lat\":22.67084065,\"lng\":113.92266904},{\"lat\":22.67073004,\"lng\":113.92301921},{\"lat\":22.5381929,\"lng\":113.90218871}]",
        "FRemark": "",
        "FIsMainRoute": 1,
        "FRoutePassInfo": [
          {
            "FRouteGUID": "1e8ea88a-60f3-42bc-8176-37c676b677d8",
            "FPassFenceGUID": "6c19e0c2-b36c-48d3-a1be-cba35f0dcc35",
            "FName": "AEE",
            "FArea": 785000.0,
            "FFormType": 2,
            "FTypeName": "李孝祯测试",
            "FPath": "[]",
            "FRadius": 500.0,
            "FCenterLon": 113.91507376,

```

```

        "FCenterLat": 22.65319852,
        "FIndex": 1
    }
]
},
"Table2": [
    {
        "FGUID": "f22a64b3-35fa-401a-b46d-12dea683326f",
        "FName": "李孝帧测试003",
        "FCode": "20250728003",
        "FStartFence": "dfd0d889-7a01-4cd4-b8ed-88825c5fb72b",
        "FStartFenceName": "海谷科技大厦",
        "FStartFormType": 2,
        "FStartTypeName": "李孝帧测试",
        "FStartPath": "[]",
        "FStartRadius": 500.0,
        "FStartCenterLon": 113.92131186,
        "FStartCenterLat": 22.67073578,
        "FEndFence": "a156ffcf-9285-4756-95f1-0cb830fdae2a",
        "FEndFenceName": "龙海家园",
        "FEndFormType": 1,
        "FEndTypeName": "李孝帧测试",
        "FEndPath": "[{"lat":22.54519516,"lng":113.89590926},{
"lat":22.53213027,"lng":113.89155722},{
"lat":22.53244843,"lng":113.9106911},{
"lat":22.54478113,"lng":113.90775015}]",
        "FEndRadius": 0.0,
        "FEndCenterLon": 113.90137618,
        "FEndCenterLat": 22.53813873,
        "FRoadPoints": "[{"lat":22.67053234,"lng":113.92123008},
{"lat":22.67044206,"lng":113.92149997},{
"lat":22.67027167,"lng":113.92194974},{
"lat":22.67023145,"lng":113.92214964},{
"lat":22.67023145,"lng":113.92214964}]",
        "FRemark": "test",
        "FIsMainRoute": 0,
        "FRoutePassInfo": [
            {
                "FRouteGUID": "f22a64b3-35fa-401a-b46d-12dea683326f",
                "FPassFenceGUID": "cd599f34-a408-4c20-8aab-6a9f246afc6d",
                "FName": "光明城站",
                "FArea": 8998000.0,
                "FFormType": 1,
                "FTypeName": "运输路线",
                "FPath": [{"lat":22.74129463,"lng":113.9512792

```

```

9},{\"lat\":22.7344746,\"lng\":113.94214189},{\"lat\":22.72191399,\"lng\":1
13.93294184},{\"lat\":22.71563635,\"lng\":113.95853057},{\"lat\":22.7350435
,\"lng\":113.98881357}],
      "FRadius": 0.0,
      "FCenterLon": 113.95822878,
      "FCenterLat": 22.7290518,
      "FIndex": 1
    }
  ]
},
{
  "FGUID": "2c84ac97-1568-4384-925b-fa2f54ca3dde",
  "FName": "李孝祯测试004",
  "FCode": "20250729181624",
  "FStartFence": "dfd0d889-7a01-4cd4-b8ed-88825c5fb72b",
  "FStartFenceName": "海谷科技大厦",
  "FStartFormType": 2,
  "FStartTypeName": "李孝祯测试",
  "FStartPath": "[]",
  "FStartRadius": 500.0,
  "FStartCenterLon": 113.92131186,
  "FStartCenterLat": 22.67073578,
  "FEndFence": "a156ffcf-9285-4756-95f1-0cb830fdae2a",
  "FEndFenceName": "龙海家园",
  "FEndFormType": 1,
  "FEndTypeName": "李孝祯测试",
  "FEndPath": "[{\"lat\":22.54519516,\"lng\":113.89590926},{\"
\"lat\":22.53213027,\"lng\":113.89155722},{\"lat\":22.53244843,\"lng\":113.9
106911},{\"lat\":22.54478113,\"lng\":113.90775015}]",
  "FEndRadius": 0.0,
  "FEndCenterLon": 113.90137618,
  "FEndCenterLat": 22.53813873,
  "FRoadPoints": "[{\"lat\":22.67053234,\"lng\":113.92123008}
,{\"lat\":22.67044206,\"lng\":113.92149997},{\"lat\":22.67027167,\"lng\":11
3.92194974},{\"lat\":22.67023145,\"lng\":113.92214964},{\"lat\":22.67023145
,\"lng\":113.92214964}]",
  "FRemark": "",
  "FIsMainRoute": 0,
  "FRoutePassInfo": [
    {
      "FRouteGUID": "2c84ac97-1568-4384-925b-fa2f54ca3dde
",
      "FPassFenceGUID": "f7500fc4-1c24-426f-9459-86d99c69
8ed5",
      "FName": "深港花卉",
      "FArea": 3330000.0,

```

```

        "FFormType": 1,
        "FTypeName": "运输路线",
        "FPath": "[{"lat":22.61231688,"lng":113.9326152},{"lat":22.61206188,"lng":113.92250332},{"lat":22.59246153,"lng":113.92435955},{"lat":22.59231458,"lng":113.94366762}]",
        "FRadius": 0.0,
        "FCenterLon": 113.93113228,
        "FCenterLat": 22.6013057,
        "FIndex": 1
    }
]
}
}
}

```

### 错误时返回:

```

//表示传入的公司的唯一标识为空，应该是数据格式不对
{
    "Result": 102,
    "Message": "FParam is null",
    "FObject": []
}
//表示token过期异常报错
{
    "Result": 104,
    "Message": "token error or timeout",
    "FObject": []
}
//表示有异常报错
{
    "Result": 105,
    "Message": "fail",
    "FObject": null
}

```

### 返回参数Table说明:

参数名	类型	说明
FAgentGUID	String	公司唯一标识
FGUID	String	路线唯一标识

FName	String	路线名称
FCode	String	路线名称
FStartFence	String	开始围栏唯一标识
FStartFenceName	String	开始围栏名称
FStartFormType	int	开始围栏类型 1: 多边形 2: 圆形
FStartTypeName	String	开始围栏类型名称
FStartPath	String	开始围栏点集合, Json字符串
FStartRadius	double	开始围栏半径
FStartCenterLon	String	开始围栏中心点经度
FStartCenterLat	String	开始围栏中心点纬度
FEndFence	String	结束围栏唯一标识
FEndFenceName	String	结束围栏名称
FEndFormType	int	结束围栏类型 1: 多边形 2: 圆形
FEndTypeName	String	结束围栏类型名称
FEndPath	String	结束围栏点集合, Json字符串
FEndRadius	double	结束围栏半径
FEndCenterLon	String	结束围栏中心点经度
FEndCenterLat	String	结束围栏中心点纬度
FRoadPoints	String	路线所有点集合, Json字符串
FRemark	String	路线描述
FIsmainRoute	int	是否主路线 1:是 0:否
FRoutePassInfo	Object	所有途经围栏, 如果为null, 表示该路线无途经围栏

## 返回参数FRoutePassInfo说明

参数名	类型	说明
-----	----	----

FRouteGUID	String	所属路线唯一标识
FPassFenceGUID	String	途径围栏唯一标识
FName	String	途径围栏名称
FFormType	int	围栏类型 1: 多边形 2: 圆形
FTypeName	String	围栏类型名称
FPath	String	围栏点集合, Json字符串
FRadius	double	开始围栏半径
FCenterLon	String	开始围栏中心点经度
FCenterLat	String	开始围栏中心点纬度
FIndex	int	途径围栏序号

## 返回参数Table1为分支路线, 如果为null ,表示无此路线

### 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期异常报错
- 105: 表示有异常报错

### 请求示例:

#### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数, 建议用对象传入
String body = "{\"FAction\":\"QueryAllRoutePlanByFGUID\",\"FGUID\":\"e286fe13-2034-45eb-9da2-d929e8a2558a\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
```

```

conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"QueryAllRoutePlanByFGUID\",\"FGUID\":\"e286fe13-2034-45eb-9da2-d929e8a2558a\", \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae741\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{

```

```
//返回的是json字符串  
return reader.ReadToEnd();  
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'  
data = {  
    'FAction': 'QueryAllRoutePlanByFGUID',  
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae741',  
    'FGUID': 'e286fe13-2034-45eb-9da2-d929e8a2558a'  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
            r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 新增路线

## 简要描述：

- 批量添加路线

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (AddRoutePlans )
FRoutePlanList	是	List< MT_RoutePlan >	路线信息列表

## MT\_RoutePlan 对象说明:

	是否		
--	----	--	--

	必须		
FName	是	string	路线名称
FCode	否	string	路线名称编码
FStartFence	是	string	开始围栏唯一标识
FEndFence	是	string	结束围栏唯一标识
FRemark	否	string	围栏备注
FRouteType	是	int	路线类型 0: 有具体路线 1: 无路线
FRoadPoints	否	string	路线所有点集合, Json字符串 [{"lat":22.67009126,"lng":113.92249915}, {"lat":22.67014155,"lng":113.9224293}, {"lat":22.67027207,"lng":113.92198938}, {"lat":22.67034136,"lng":113.92202024}, {"lat":22.6708117,"lng":113.92216942}]. <b>注意:当 FRouteType=0时, 此字段为必填</b>
FDistance	否	int	运输总里程
FDuration	否	int	运输总时长,单位小时
FPassFenceGUIDs	否	string	途径围栏唯一标识,多个以英文逗号隔开

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的对象为空, 应该是数据格式不对
{
  "Result": 102,
  "Message": "FRoutePlanList is null",
}
```

```

    "Message": "FRoutePlanList is null",
    "FObject": []
  }
  //表示有异常报错
  {
    "Result": 105,
    "Message": "fail",
    "FObject": []
  }
  //102表示传入的参数不符合规则，具体看Message提示
  {
    "Result": 102,
    "Message": {model.FName}:FStartFence or FEndFence is null,
    "FObject": []
  }
}

```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
  \"FAction\": \"AddRoutePlans\",
  \"FRoutePlanList\": [
    {
      \"FName\": \"000154336\",
      \"FCode\": \"000154336\",
      \"FStartFence\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",
      \"FEndFence\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",
      \"FPassFenceGUIDs\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",
      \"FRoadPoints\": \"[{\"lat\":22.67009126,\"lng\":113.92249915},{
        \"lat\":22.67014155,\"lng\":113.9224293},{\"lat\":22.67027207,\"lng\":113.9
        2198938},{\"lat\":22.67034136,\"lng\":113.92202024},{\"lat\":22.6708117,\"l
        ng\":113.92216942},{\"lat\":22.67093116,\"lng\":113.92221918}]\",
      \"FRouteType\":0,
    }
  ]
}";

```

```

    ],
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae742\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"AddRoutePlans\",
    \"FRoutePlanList\": [
        {
            \"FName\": \"000154336\",
            \"FCode\": \"000154336\",
            \"FStartFence\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",

```

```

        \ "FEndFence\" : \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\" ,
        \ "FPassFenceGUIDs\" : \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\" ,
        \ "FRoadPoints\" : \"[{\ \"lat\" : 22.67009126, \ \"lng\" : 113.92249915}, {
\ \"lat\" : 22.67014155, \ \"lng\" : 113.9224293}, {\ \"lat\" : 22.67027207, \ \"lng\" : 113.9
2198938}, {\ \"lat\" : 22.67034136, \ \"lng\" : 113.92202024}, {\ \"lat\" : 22.6708117, \ \"l
ng\" : 113.92216942}, {\ \"lat\" : 22.67093116, \ \"lng\" : 113.92221918}]\" ,
        \ "FRouteType\" : 0,
    }
],
\ "FTokenID\" : \"ec51d691-d9a7-4f74-95a8-1856be2ae742\" ,
\ "FTimeDifferent\" : 28800,
\ "FLanguage\" : \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

## Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'AddRoutePlans',
    'FRoutePlanList': [
        {
            'FName': '000154336',
            'FCode': '000154336',
            'FStartFence': '5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d',
            'FEndFence': '5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d',
            'FPassFenceGUIDs': '5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d',
            'FRoadPoints': '[{\ \"lat\" : 22.67009126, \ \"lng\" : 113.92249915}, {\ \"l
at\" : 22.67014155, \ \"lng\" : 113.9224293}, {\ \"lat\" : 22.67027207, \ \"lng\" : 113.9219

```

```
8938},{\"lat\":22.67034136,\"lng\":113.92202024},{\"lat\":22.6708117,\"lng\"
\":113.92216942},{\"lat\":22.67093116,\"lng\":113.92221918}]',
    'FRouteType':0,
  }
],
'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
'FTimeDifferent': 28800,
'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 修改路线

## 简要描述：

- 修改路线

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateRoutePlan )
FT_RoutePlan	是	object	路线信息

## FT\_RoutePlan 对象说明：

	是否		
--	----	--	--

	必须		
FGUID	是	string	路线唯一标识
FName	是	string	路线名称
FCode	否	string	路线名称编码
FStartFence	是	string	开始围栏唯一标识
FEndFence	是	string	结束围栏唯一标识
FRemark	否	string	围栏备注
FRouteType	是	int	路线类型 0: 有具体路线 1: 无路线
FRoadPoints	否	string	路线所有点集合, Json字符串 "[{"lat":22.67009126,"lng":113.92249915}, {"lat":22.67014155,"lng":113.9224293}, {"lat":22.67027207,"lng":113.92198938}, {"lat":22.67034136,"lng":113.92202024}, {"lat":22.6708117,"lng":113.92216942}]. <b>注意:当 FRouteType=0时, 此字段为必填</b>
FDistance	否	int	运输总里程
FDuration	否	int	运输总时长,单位小时
FPassFenceGUIDs	否	string	途径围栏唯一标识,多个以英文逗号隔开

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的对象为空, 应该是数据格式不对
{
```

```

{
  "Result": 102,
  "Message": "FRoutePlanList is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
//102表示传入的参数不符合规则，具体看Message提示
{
  "Result": 102,
  "Message": {model.FName}:FStartFence or FEndFence is null,
  "FObject": []
}

```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
  \"FAction\": \"UpdateRoutePlan\",
  \"FT_RoutePlan\":
    {
      \"FGUID\": \"09cf2885-b998-4917-bb29-fc9dbbb054c8\",
      \"FName\": \"000154336\",
      \"FCode\": \"000154336\",
      \"FStartFence\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",
      \"FEndFence\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",
      \"FPassFenceGUIDs\": \"5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\",
      \"FRoadPoints\": \"[{\"lat\":22.67009126,\"lng\":113.92249915},{
        \"lat\":22.67014155,\"lng\":113.9224293},{\"lat\":22.67027207,\"lng\":113.9

```

```

ng\":113.92216942},{\"lat\":22.67093116,\"lng\":113.92221918}}\",
    \"FRouteType\":0,
    },
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae742\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"UpdateRoutePlan\",
    \"FT_RoutePlan\":
    {
        \"FGUID\": \"09cf2885-b998-4917-bb29-fc9dbbb054c8\",

```

```

        \ "FName\": \ "000154336\ ",
        \ "FCode\": \ "000154336\ ",
        \ "FStartFence\": \ "5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\ ",
        \ "FEndFence\": \ "5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\ ",
        \ "FPassFenceGUIDs\": \ "5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d\ ",
        \ "FRoadPoints\": \ "[{\ "lat\":22.67009126, \ "lng\":113.92249915},{
\ "lat\":22.67014155, \ "lng\":113.9224293},{ \ "lat\":22.67027207, \ "lng\":113.9
2198938},{ \ "lat\":22.67034136, \ "lng\":113.92202024},{ \ "lat\":22.6708117, \ "l
ng\":113.92216942},{ \ "lat\":22.67093116, \ "lng\":113.92221918}]\ ",
        \ "FRouteType\":0,
    } ,
    \ "FTokenID\": \ "ec51d691-d9a7-4f74-95a8-1856be2ae742\ ",
    \ "FTimeDifferent\": 28800,
    \ "FLanguage\": \ "1\ "
}";
Encoding encoding = Encoding.UTF8;
HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpResponse response = (HttpResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'UpdateRoutePlan',
    'FT_RoutePlan':
        {
            'FGUID': '09cf2885-b998-4917-bb29-fc9dbbb054c8',
            'FName': '000154336',
            'FCode': '000154336',
            'FStartFence': '5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d',
            'FEndFence': '5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d',

```

```
'FPassFenceGUIDs': '5ae45d8e-18bd-4eaa-ae4b-017ab7c3f36d',
'FRoadPoints': '[{"lat":22.67009126,"lng":113.92249915},{"lat":22.67014155,"lng":113.9224293},{"lat":22.67027207,"lng":113.92198938},{"lat":22.67034136,"lng":113.92202024},{"lat":22.6708117,"lng":113.92216942},{"lat":22.67093116,"lng":113.92221918}]',
'FRouteType':0,
}
,
'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
'FTimeDifferent': 28800,
'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 围栏管理

# 批量添加围栏

## 简要描述：

- 批量添加围栏

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (AddGISFenceBatch )
FAgentGUID	是	string	公司唯一标识
FT_GISFenceList	是	List<MT_GISFence>	围栏信息列表

## MT\_GISFence 对象说明：

--	--	--	--

参数名	是否必须	类型	说明
FName	是	string	围栏名称
FFenceTypeName	是	string	围栏分组名称
FNumber	否	string	围栏编号
FRemark	否	string	围栏备注
FArea	否	double	区域
FFormType	否	int	围栏类型（1：多边形；2：圆形）
FFenceTypeGUID	否	string	围栏分组唯一标识
FPath	否	string	围栏经纬度集合
FRadius	否	double	半径
FCenterLon	否	double	中心点经度
FCenterLat	否	double	中心点纬度
FAddress	否	string	地址

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的对象为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FT_GISFenceList is null",
  "FObject": []
}
//表示有异常报错
```

```

{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
//表示传入的围栏名称为空报错
{
  "Result": 106,
  "Message": "",
  "FObject": []
}

//表示传入的围栏分组名称为空报错
{
  "Result": 107,
  "Message": "",
  "FObject": []
}
//表示传入的围栏信息有重复的
{
  "Result": 108,
  "Message": "重复围栏名称",
  "FObject": []
}

```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错
- 106: 表示传入的围栏名称为空
- 107: 表示传入的围栏分组的名称为空
- 108: 表示传入的围栏信息有重复的

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{

```

```

    \"FAction\": \"AddGISFenceBatch\",
    \"FAgentGUID\": \"c4b7730e-f74b-4dee-833b-0128bf5dd7f4\",
    \"FT_GISFenceList\": [
        {
            \"FName\": \"我的围栏1\",
            \"FFenceTypeName\": \"我的围栏类型1\"
        },
        {
            \"FName\": \"我的围栏2\",
            \"FFenceTypeName\": \"我的围栏类型2\"
        }
    ],
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"AddGISFenceBatch\",
    \"FAgentGUID\": \"c4b7730e-f74b-4dee-833b-0128bf5dd7f4\",
    \"FT_GISFenceList\": [
        {
            \"FName\": \"我的围栏1\",
            \"FFenceTypeName\": \"我的围栏类型1\"
        },
        {
            \"FName\": \"我的围栏2\",
            \"FFenceTypeName\": \"我的围栏类型2\"
        }
    ],
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'AddGISFenceBatch',
    'FAgentGUID': 'c4b7730e-f74b-4dee-833b-0128bf5dd7f4',
    'FT_GISFenceList': '[

```

```
{
    {"FName": "我的围栏1",
      "FFenceTypeName": "我的围栏类型1"},
    {"FName": "我的围栏2",
      "FFenceTypeName": "我的围栏类型2"}
  ],
  'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
  'FTimeDifferent': 28800,
  'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                  r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 查询公司下所有围栏及围栏分组信息

## 简要描述：

- 查询公司下所有围栏及围栏分组信息

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryGISFenceByFAgentGUID )
FAgentGUID	是	string	公司唯一标识

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": [
    {
      "FGUID": "ed86c973-07e3-4f0a-afed-00ad8c176db5",
      "FName": "官田地铁站",
      "FNumber": "04",
      "FRemark": "测试",
      "FArea": 87313.54065700735,
      "FFormType": 1,
      "FFenceTypeGUID": "df0f8ab2-742e-448b-9c5e-088f88e996d5",
      "FPath": "[{\"lat\":22.68714446,\"lng\":113.94369331},{\"lat\":22.68683927,\"lng\":113.94710827},{\"lat\":22.68504754,\"lng\":113.94646312},{\"lat\":22.68561273,\"lng\":113.94355122},{\"lat\":22.6858369,\"lng\":113.94067669}]",
      "FRadius": 0,
      "FCenterLon": 113.94450032,
      "FCenterLat": 22.68618174,
      "FAddress": "广东省深圳市宝安区北环路18新永丰官田工业区附近47米",
      "FCreateTime": "2023-02-23T02:42:04.87",
      "FAgentGUID": "661e7209-c77e-4a54-afb8-6d956f962bdf",
      "FAgentName": "软件测试部",
      "FFenceTypeName": "运输路线"
    }
  ]
}
```

### 错误时返回:

```
//表示传入的公司唯一标识为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FAgentGUID is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

### 返回参数说明:

参数名	类型	说明
FGUID	string	围栏唯一标识
FName	string	围栏名称
FNumber	string	围栏编号
FRemark	string	备注
FArea	string	面积
FFormType	int	类型（1：多边形；2：圆形）
FFenceTypeGUID	string	围栏分组唯一标识
FPath	string	点集合
FRadius	int	半径
FCenterLon	double	中心点经度
FCenterLat	double	中心点纬度
FAddress	string	地址
FCreateTime	string	创建时间
FAgentGUID	string	公司唯一标识
FAgentName	string	公司名称
FFenceTypeName	string	围栏分组名称

## 备注:

- 更多返回错误代码如下：
- 102：请求参数错误
- 104：token过期
- 105：表示有异常报错

## 请求示例:

### Java:

```
String result = "";
```

```

//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
    \"FAction\": \"QueryGISFenceByFAgentGUID\",
    \"FAgentGUID\": \"661e7209-c77e-4a54-afb8-6d956f962bdf\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"QueryGISFenceByFAgentGUID\",

```

```

    \"FAgentGUID\": \"661e7209-c77e-4a54-afb8-6d956f962bdf\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryGISFenceByFAgentGUID',
    'FAgentGUID': '661e7209-c77e-4a54-afb8-6d956f962bdf',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
    'FTimeDifferent': 28800,
    'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
    r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;

```



# 修改围栏

## 简要描述：

- 修改围栏

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateGISFence )

|FT\_GISFence |是 | FT\_GISFence | 围栏信息 |

## FT\_GISFence 对象说明：

参数名	是否必须	类型	说明

FGUID	是	string	围栏唯一标识
FAgentGUID	是	string	公司唯一标识
FName	是	string	围栏名称
FNumber	否	string	围栏编号
FRemark	否	string	围栏备注
FArea	否	double	区域
FFormType	否	int	围栏类型（1：多边形；2：圆形）
FFenceTypeGUID	否	string	围栏分组唯一标识
FPath	否	string	围栏经纬度集合
FRadius	否	double	半径
FCenterLon	否	double	中心点经度
FCenterLat	否	double	中心点纬度
FAddress	否	string	地址

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的对象为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FT_GISFence is null",
  "FObject": []
}
//表示有异常报错
```

```

{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}

//表示传入的围栏不存在
{
  "Result": 107,
  "Message": "exists",
  "FObject": []
}

```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错
- 107: 表示传入的围栏不存在

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
  \"FAction\": \"UpdateGISFence\",
  \"FT_GISFence\": {
    \"FGUID\": \"BB6999E9-8571-4D64-9C35-5D51C88F1133\",
    \"FName\": \"我的围栏3\",
    \"FNumber\": \"\",
    \"FRemark\": \"\",
    \"FArea\": 0,
    \"FFormType\": 1,
    \"FFenceTypeGUID\": \"\",
    \"FPath\": \"\",
    \"FRadius\": 0,
    \"FCenterLon\": 0,
    \"FCenterLat\": 0,
    \"FAddress\": \"\",

```

```

        \"FAgentGUID\": \"661E7209-C77E-4A54-AFB8-6D956F962BDF\"
    },
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"UpdateGISFence\",
    \"FT_GISFence\": {
        \"FGUID\": \"BB6999E9-8571-4D64-9C35-5D51C88F1133\",
        \"FName\": \"我的围栏3\",
        \"FNumber\": \"\",

```

```

        "FRemark": "",
        "FArea": 0,
        "FFormType": 1,
        "FFenceTypeGUID": "",
        "FPath": "",
        "FRadius": 0,
        "FCenterLon": 0,
        "FCenterLat": 0,
        "FAddress": "",
        "FAgentGUID": "661E7209-C77E-4A54-AFB8-6D956F962BDF"
    },
    "FTokenID": "ec51d691-d9a7-4f74-95a8-1856be2ae745",
    "FTimeDifferent": 28800,
    "FLanguage": "1"
}";
Encoding encoding = Encoding.UTF8;
HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpResponse response = (HttpResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'UpdateGISFence',
    'FT_GISFence': {
        'FGUID': 'BB6999E9-8571-4D64-9C35-5D51C88F1133',
        'FName': '我的围栏3',
        'FNumber': '',
        'FRemark': '',
        'FArea': 0,
        'FFormType': 1,
    }
}

```

```
'FFenceTypeGUID': '',
'FPath': '',
'FRadius': 0,
'FCenterLon': 0,
'FCenterLat': 0,
'FAddress': '',
'FAgentGUID': '661E7209-C77E-4A54-AFB8-6D956F962BDF'
},
'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
'FTimeDifferent': 28800,
'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 批量删除围栏

## 简要描述：

- 批量删除围栏

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (DeleteGISFences )
FGUIDs	是	string	围栏唯一标识列表, 多个围栏标识,用逗号隔开 (例:“BB6999E9-8571-4D64-9C35-5D51C88F1133,BB6999E9-8571-4D64-9C35-5D51C88F1133”)

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的围栏唯一标识报错,应该是数据格式不对
```

```
{
  "Result": 102,
  "Message": "FGUIDs is null",
  "FObject": []
}
```

```
//表示有异常报错
```

```
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

```
//表示传入的围栏设置了路线的起始点
```

```
{
  "Result": 125,
  "Message": "围栏名称",
  "FObject": []
}
```

```
//表示传入的围栏设置了路线的结束点
```

```
{
  "Result": 126,
  "Message": "围栏名称",
  "FObject": []
}
```

```
//表示传入的围栏设置了路线的途径点
```

```
{
  "Result": 127,
  "Message": "围栏名称",
  "FObject": []
}
```

```
//表示传入的围栏绑定了车辆
```

```
{
```

```

"Result": 124,
"Message": "围栏名称",
"FObject": []
}

```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错
- 124: 表示传入的围栏绑定了车辆
- 125: 表示传入的围栏设置了路线的起始点
- 126: 表示传入的围栏设置了路线的结束点
- 127: 表示传入的围栏设置了路线的途径点

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
    \"FAction\": \"DeleteGISFences\",
    \"FGUIDs\": \"BB6999E9-8571-4D64-9C35-5D51C88F1133\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数

```

```

pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"DeleteGISFences\",
    \"FGUIDs\": \"BB6999E9-8571-4D64-9C35-5D51C88F1133\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'DeleteGISFences',
    'FGUIDs': 'BB6999E9-8571-4D64-9C35-5D51C88F1133',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
    'FTimeDifferent': 28800,
    'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 查询公司下所有围栏分组信息

## 简要描述：

- 查询公司下所有围栏分组信息

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryGISFenceType )
FAgentGUID	是	string	公司唯一标识

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "success",
  "FObject": [
    {
      "FGUID": "df0f8ab2-742e-448b-9c5e-088f88e996d5",
      "FIcon": "6",
      "FName": "运输路线",
      "FStrokeColor": "rgba(53, 106, 195, 0.8)",
      "FFillColor": "rgba(53, 106, 195, 0.4)"
    }
  ]
}
```

### 错误时返回:

```
//表示传入的公司唯一标识为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FAgentGUID is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

### 返回参数说明:

参数名	类型	说明
FGUID	string	围栏分组唯一标识
FIcon	string	图标
FName	string	围栏分组名称
FStrokeColor	string	边框色
FFillColor	string	填充色

### 备注:

- 更多返回错误代码如下：
- 102：请求参数错误
- 104：token过期
- 105：表示有异常报错

## 请求示例：

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{
    \"FAction\": \"QueryGISFenceType\",
    \"FAgentGUID\": \"661e7209-c77e-4a54-afb8-6d956f962bdf\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
}
```

```
//返回的是json字符串
return result;
```

**C# :**

```
//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{
    \"FAction\": \"QueryGISFenceType\",
    \"FAgentGUID\": \"661e7209-c77e-4a54-afb8-6d956f962bdf\",
    \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\",
    \"FTimeDifferent\": 28800,
    \"FLanguage\": \"1\"
}";
Encoding encoding = Encoding.UTF8;
HttpRequest request = (HttpRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpResponse response = (HttpResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryGISFenceType',
    'FAgentGUID': '661e7209-c77e-4a54-afb8-6d956f962bdf',
    'FTokenID': 'ec51d691-d9a7-4f74-95a8-1856be2ae745',
    'FTimeDifferent': 28800,
    'FLanguage': '1'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
```

```
'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 围栏绑定车辆

## 简要描述：

- 围栏绑定车辆

## 接口版本：

版本号	制定人	制定日期	修订日期
1.0.0	lenny	2022-09-22	2020-09-22

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/GISServer>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/GISServer>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateGISVihicleBindFence)
FT_GISVihicleBindFence	是	FT_GISVihicleBindFence	车辆围栏信息列表

## FT\_GISVihicleBindFence 对象说明:

参数名	是否必须	类型	说明
FVehicleGUID	是	string	车辆唯一标识
FFenceGUID	是	string	围栏唯一标识,多个围栏以英文逗号隔开
FInAlarm	否	int	进围栏报警
FOutAlarm	否	int	出围栏报警
FStopAlarm	否	int	围栏内超时停留报警
FStopTime	否	int	围栏超时停留门限值
FOutUnLockAlarm	否	int	出围栏未上锁报警
FUnLockFence	否	int	围栏内开锁（开启后只能在这些特定围栏内才能开锁）

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "success",
  "FObject": []
}
```

### 错误时返回:

```
//表示传入的对象为空，应该是数据格式不对
{
  "Result": 102,
  "Message": "FT_GISFenceList is null",
  "FObject": []
}
//表示有异常报错
{
  "Result": 105,
  "Message": "fail",
  "FObject": []
}
```

## 备注:

- 更多返回错误代码如下:
- 102: 请求参数错误
- 104: token过期
- 105: 表示有异常报错

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\r\n  \"FAction\": \"UpdateGISVehicleBindFence\", \r\n  \"FT_GISVehicleBindFence\": {\r\n    \"FVehicleGUID\": \"36f13b9b-62f9-4de2-aa82-9c065cb97027\", \r\n    \"FFenceGUID\": \"32f30c76-a7ac-4f68-b384-230e1ac471c9,655179b6-0cd4-46d9-a8cc-5f8b793fedd4,46e70dbf-8e6b-431d-84c7-7f1eac1cb603,6db5a322-b1c9-4fc0-b5eb-c962b0d8d610,fbdb37c1-bc4b-4433-8110-e1eb9bea8f5e\", \r\n    \"FInAlarm\": 0, \r\n    \"FOutAlarm\": 0, \r\n    \"FStopAlarm\": 0, \r\n    \"FStopTime\": 0, \r\n    \"FOverspeedAlarm\": 0, \r\n    \"FOverspeedValue\": 0, \r\n    \"FOutUnlockAlarm\": 0, \r\n    \"FUnlockFence\": 1, \r\n    \"FOffline\": 0, \r\n  }, \r\n  \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\", \r\n  \"FTimeDifferent\": 28800, \r\n  \"FLanguage\": \"1\"\r\n}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
```

```

tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数 ,建议用对象传入
string body = "{\r\n  \"FAction\": \"UpdateGISVihicleBindFence\", \r\n  \"FT_GISVihicleBindFence\": {\r\n    \"FVehicleGUID\": \"36f13b9b-62f9-4de2-aa82-9c065cb97027\", \r\n    \"FFenceGUID\": \"32f30c76-a7ac-4f68-b384-230e1ac471c9,655179b6-0cd4-46d9-a8cc-5f8b793fedd4,46e70dbf-8e6b-431d-84c7-7f1eac1cb603,6db5a322-b1c9-4fc0-b5eb-c962b0d8d610,fbdb37c1-bc4b-4433-8110-e1eb9bea8f5e\", \r\n    \"FInAlarm\": 0, \r\n    \"FOutAlarm\": 0, \r\n    \"FStopAlarm\": 0, \r\n    \"FStopTime\": 0, \r\n    \"FOverspeedAlarm\": 0, \r\n    \"FOverspeedValue\": 0, \r\n    \"FOutUnLockAlarm\": 0, \r\n    \"FUnLockFence\": 1, \r\n    \"FOffline\": 0, \r\n  }, \r\n  \"FTokenID\": \"ec51d691-d9a7-4f74-95a8-1856be2ae745\", \r\n  \"FTimeDifferent\": 28800, \r\n  \"FLanguage\": \"1\"\r\n}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = '{ "FAction": "UpdateGISVihicleBindFence", "FT_GISVihicleBindFence"
: { "FVehicleGUID": "36f13b9b-62f9-4de2-aa82-9c065cb97027", "FFenceGUID": "
32f30c76-a7ac-4f68-b384-230e1ac471c9,655179b6-0cd4-46d9-a8cc-5f8b793fedd4,4
6e70dbf-8e6b-431d-84c7-7f1eac1cb603,6db5a322-b1c9-4fc0-b5eb-c962b0d8d610,fb
db37c1-bc4b-4433-8110-e1eb9bea8f5e", "FInAlarm": 0, "FOutAlarm": 0, "FStopA
larm": 0, "FStopTime": 0, "FOverspeedAlarm": 0, "FOverspeedValue": 0, "FOut
UnLockAlarm": 0, "FUnLockFence": 1, "FOffline": 0, }, "FTokenID": "ec51d691
-d9a7-4f74-95a8-1856be2ae745", "FTimeDifferent": 28800, "FLanguage": "1" }'
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 公司信息

## 简要描述：

- 获取公司基础信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2024-03-26	

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名(QueryAllSuperAdminAgentList)

## 返回示例:

### 正确时返回:

```
{
```

```
"Result": 200,
"Message": "check token success",
"FObject":
  [{
    "FAgentGUID": "37c1b3fc-119e-4ed1-b87d-47f3a6fa73bc",
    "FName": "你猜",
    "FCode": "你猜",
    "FContacts": "",
    "FTelephone": "",
    "FEmailAddress": "",
    "FCountry": "China",
    "FDescribe": "",
    "FStatus": 1,
    "FCreateTime": "2021-06-28T09:43:17.117",
    "FFatherAgentGUID": "9edd97dd-6ab2-4b0e-ab11-731847271a6b",
    "FFatherAgentName": "特种业务",
    "FFatherAgentCode": "1111",
    "FTimeDifferent": 28800,
    "RowNo": 1
  },
  {
    "FAgentGUID": "d5456af1-ffa2-416b-a335-116436479998",
    "FName": "紫金矿业集团黄金冶炼有限公司",
    "FCode": "紫金矿业集",
    "FContacts": "",
    "FTelephone": "",
    "FEmailAddress": "",
    "FCountry": "China",
    "FDescribe": "",
    "FStatus": 1,
    "FCreateTime": "2021-06-04T03:45:42.38",
    "FFatherAgentGUID": "9edd97dd-6ab2-4b0e-ab11-731847271a6b",
    "FFatherAgentName": "特种业务",
    "FFatherAgentCode": "1111",
    "FTimeDifferent": 28800,
    "RowNo": 2
  },
  {
    "FAgentGUID": "82c5ee8e-8f57-4021-ad27-6a1daa6a4637",
    "FName": "茂名市",
    "FCode": "茂名市",
    "FContacts": "",
    "FTelephone": "",
    "FEmailAddress": "",
    "FCountry": "China",
    "FDescribe": "",
```

```

    "FStatus": 1,
    "FCreateTime": "2021-05-31T10:17:15.48",
    "FFatherAgentGUID": "9edd97dd-6ab2-4b0e-ab11-731847271a6b",
    "FFatherAgentName": "特种业务",
    "FFatherAgentCode": "1111",
    "FTimeDifferent": 28800,
    "RowNo": 3
  },
  {
    "FAgentGUID": "256c6cd1-6a19-42a4-a572-de661d1c0e61",
    "FName": "北京市",
    "FCode": "北京市",
    "FContacts": "",
    "FTelephone": "",
    "FEmailAddress": "",
    "FCountry": "China",
    "FDescribe": "",
    "FStatus": 1,
    "FCreateTime": "2021-04-25T02:39:41.69",
    "FFatherAgentGUID": "9edd97dd-6ab2-4b0e-ab11-731847271a6b",
    "FFatherAgentName": "特种业务",
    "FFatherAgentCode": "1111",
    "FTimeDifferent": 28800,
    "RowNo": 4
  },
  {
    "FAgentGUID": "769d5f45-e8eb-40eb-b0e1-379131aab871",
    "FName": "深圳市",
    "FCode": "深圳市",
    "FContacts": "",
    "FTelephone": "",
    "FEmailAddress": "",
    "FCountry": "China",
    "FDescribe": "",
    "FStatus": 1,
    "FCreateTime": "2021-04-25T02:39:41.68",
    "FFatherAgentGUID": "9edd97dd-6ab2-4b0e-ab11-731847271a6b",
    "FFatherAgentName": "特种业务",
    "FFatherAgentCode": "1111",
    "FTimeDifferent": 28800,
    "RowNo": 5
  },
  {
    "FAgentGUID": "68afd094-d2c0-4796-b685-089f71b48e47",
    "FName": "中国黄金",
    "FCode": "中国黄金",

```

```

"Contacts": "",
"Telephone": "",
"EmailAddress": "",
"Country": "China",
"Describe": "",
"Status": 1,
"CreateTime": "2021-04-25T02:39:41.643",
"FFatherAgentGUID": "9edd97dd-6ab2-4b0e-ab11-731847271a6b",
"FFatherAgentName": "特种业务",
"FFatherAgentCode": "1111",
"FTimeDifferent": 28800,
"RowNo": 6
    }
  ]
}

```

**错误时返回:**

```

{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}

```

**返回参数说明:**

参数名	类型	说明
FAgentGUID	String	公司唯一标识
FFatherAgentGUID	String	上级公司唯一标识
FName	String	公司名
FFatherAgentName	String	上级公司名
FCode	String	公司编码
FFatherAgentCode	String	上级公司编码
FVehicleCode	String	车辆编码
FContacts	String	联系人

FDescribe	String	描述
FTelephone	String	联系电话
FEmailAddress	String	邮件地址
FCountry	String	所属国家
FTimeDifferent	Int	时差
FCreateTime	DateTime	注册时间

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 102: 请求参数错误

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"QueryAllSuperAdminAgentList\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
```

```
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
InputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;
```

**C# :**

```
//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction:\"QueryAllSuperAdminAgentList\",FTokenID:\"3acef045
-d302-4032-b40a-d9ee6c1519cd\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAllSuperAdminAgentList',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
}
data = parse.urlencode(data).encode('utf-8')
headers = {
```

```
'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 视频监控

**实时视频监控**

---

**停止视频监控**

---

# 实时视频监控

## 简要描述：

- 发送拉流指令,并返回拉流地址

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL:

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Monitor>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Monitor>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数:

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (RealTimeVideoPlay)
FAssetGUID	是	string	设备唯一标识
FChannel	否	Int	播放通道,不传则默认为通道1
FInterval	否	Int	拉流时长（单位秒，默认1分钟，支持0-10分钟）

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": {
    "flv": "http://media.assetscontrols.com:8111/rtp/80425090031/1.live.flv",
    "https_flv": "https://media.assetscontrols.com:7443/rtp/80425090031/1/1.live.flv",
    "ws_flv": "ws://media.assetscontrols.com:8111/rtp/80425090031/1.live.flv",
    "wss_flv": "wss://media.assetscontrols.com:7443/rtp/80425090031/1.live.flv",
    "rtmp": "rtmp://media.assetscontrols.com:8935/rtp/80425090031/1",
    "rtsp": "rtsp://media.assetscontrols.com:8554/rtp/80425090031/1"
  }
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明(根据实际需求选择一种来进行拉流):

参数名	类型	说明
flv	string	flv格式http 拉流地址
https_flv	string	flv格式https 拉流地址
ws_flv	string	flv格式ws 拉流地址
wss_flv	string	flv格式wss 拉流地址
rtmp	string	rtmp 拉流地址
rtsp	string	rtsp 拉流地址

## 备注:

- 更多返回错误代码如下:
- 102: 参数错误
- 105: 系统异常
- 108: 指令发送异常
- 122: 设备未配置监控服务器
- 123: 设备不支持
- 111: 设备离线或休眠

## 请求示例:

### Java:

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\\"RealTimeVideoPlay\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\\",FChannel:1,FInterval:60}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
```

```

}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数 , 建议用对象传入
string body = "{FAction:\\"RealTimeVideoPlay\\",FTokenID:\\"3acef045-d302-4032-b40a-d9ee6c1519cd\\",FAssetGUID:\\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\\",FChannel:1,FInterval:60}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{
//返回的是json字符串
return reader.ReadToEnd();
}

```

**Python :**

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Monitor'
data = {
'FAction': 'RealTimeVideoPlay',
'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd' ,
'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c' ,
'FChannel':1 ,
'FInterval':60
}
data = parse.urlencode(data).encode('utf-8')
headers = {
'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',

```

```
'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 停止视频监控

## 简要描述：

- 发送停止拉流指令

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2020-06-20	2020-06-20

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Monitor>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Monitor>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (RealTimeVideoPlayControl)
FAssetGUID	是	string	设备唯一标识
FChannel	否	Int	播放通道,不传则默认为通道1

## 返回示例：

**正确时返回:**

```
{
  "Result": 200,
  "Message": "ins send ok",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 102: 参数错误
- 105: 系统异常
- 108: 指令发送异常
- 122: 设备未配置监控服务器
- 123: 设备不支持
- 111: 设备离线或休眠

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数,建议用对象传入
String body = "{FAction:\"RealTimeVideoPlayControl\",FTokenID:\"3acef045-d302-4032-b40a-d9ee6c1519cd\",FAssetGUID:\"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c\",FChannel:1}";
URL realUrl = new URL(url);
// 设置通用请求的属性
```

```

URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Monitor";
//请求参数 ,json格式参数,建议用对象传入
string body = "{FAction: \"RealTimeVideoPlayControl\", FTokenID: \"3acef045-d3
02-4032-b40a-d9ee6c1519cd\", FAssetGUID: \"7f6b94c8-abda-48aa-9eec-9e2d8c5f6a
3c\", FChannel:1}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

```

```
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}
```

### Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Monitor'
data = {
    'FAction': 'RealTimeVideoPlayControl',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FAssetGUID': '7f6b94c8-abda-48aa-9eec-9e2d8c5f6a3c',
    'FChannel':1
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 司机管理

# 新增司机

## 简要描述：

- 新增司机信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2026-03-02	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (AddAdminDriver)
			{“FDriverName”:司机名称,“FLicenseNumber”:驾驶证编号,“FDriverPhone”:司机电话,“FIDNumber”:司机身份证号,“FDescription”:描

FT_AdminDriverParam	是	Object	述,"FCountryResidenc":城市,"FDistrictRegion":城镇,"FAgentGUID":公司唯一标识,"FHeadPortrait":头像Base64图片","FDrivingLicenseFront":驾驶证正面照Base64图片","FDrivingLicenseBack":驾驶证反面照Base64图片"}
---------------------	---	--------	--

**FT\_AdminDriverParam 对象说明:**

参数名	是否必须	类型	说明
FDriverName	是	string	司机名称
FLicenseNumber	是	string	驾驶证编号
FDriverPhone	是	string	司机电话
FIDNumber	是	string	司机身份证号
FDescription	是	string	描述
FCountryResidenc	是	string	城市
FDistrictRegion	是	string	城镇
FHeadPortrait	否	string	头像Base64图片
FDrivingLicenseFront	否	string	驾驶证正面照Base64图片
FDrivingLicenseBack	否	string	FDrivingLicenseBack
FAgentGUID	否	string	公司唯一标识,该字段不传,则默认为当前用户所在公司

**返回示例:****正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"AddAdminDriver\", \"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\", \"FT_AdminDriverParam\":{\"FDriverName\":\"12321\", \"FLicenseNumber\":\"12321333\", \"FDrivingType\":\"C\", \"FDriverPhone\":\"5481166\", \"FDescription\":\"ddb\", \"FIDNumber\":\"323432543543\", \"FCountryResidenc\":\"322\", \"FDistrictRegion\":\"211\", \"FHeadPortrait\":\"\", \"FDrivingLicenseFront\":\"\", \"FDrivingLicenseBack\":\"\"}}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
```

```

// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"AddAdminDriver\", \"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\", \"FT_AdminDriverParam\":{\"FDriverName\":\"12321\", \"FLicenseNumber\":\"12321333\", \"FDrivingType\":\"C\", \"FDriverPhone\":\"5481166\", \"FDescription\":\"ddb\", \"FIDNumber\":\"323432543543\", \"FCountryResidenc\":\"322\", \"FDistrictRegion\":\"211\", \"FHeadPortrait\":\"\", \"FDrivingLicenseFront\":\"\", \"FDrivingLicenseBack\":\"\"}}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
{

```

```
//返回的是json字符串  
return reader.ReadToEnd();  
}
```

## Python :

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'  
data = {  
    'FAction': 'AddAdminDriver',  
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',  
    'FT_AdminDriverParam': '{  
        'FDriverName': '12321',  
        'FLicenseNumber': '12321333',  
        'FDrivingType': 'C',  
        'FDriverPhone': '5481166',  
        'FDescription': 'ddb',  
        'FIDNumber': '323432543543',  
        'FCountryResidenc': '322',  
        'FDistrictRegion': '211',  
        'FHeadPortrait': '',  
        'FDrivingLicenseFront': '',  
        'FDrivingLicenseBack': ''  
    }'  
}  
data = parse.urlencode(data).encode('utf-8')  
headers = {  
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.  
36 (KHTML, like Gecko) '  
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',  
    'Connection': 'keep-alive'  
}  
req = request.Request(url, headers=headers, data=data)  
page = request.urlopen(req).read()  
page = page.decode('utf-8')  
# json_array = json.loads(page)  
return page;
```

# 修改司机

## 简要描述：

- 修改司机信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2026-03-02	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (UpdateAdminDriver)
			{“FGUID”:司机信息唯一标识,“FDriverName”:司机名称,“FLicenseNumber”:驾驶证编号,“FDriverPhone”:司机电话,“FIDNumber”:司机身

FT_AdminDriverParam	是	Object	份证号,"FDescription":描述,"FCountryResidenc":城市,"FDistrictRegion":城镇,"FAgentGUID":公司唯一标识,"FHeadPortrait":"头像Base64图片","FDrivingLicenseFront":"驾驶证正面照Base64图片","FDrivingLicenseBack":"驾驶证反面照Base64图片"}
---------------------	---	--------	---

**FT\_AdminDriverParam 对象说明:**

参数名	是否必须	类型	说明
FGUID	是	string	司机信息唯一标识
FDriverName	是	string	司机名称
FLicenseNumber	是	string	驾驶证编号
FDriverPhone	是	string	司机电话
FIDNumber	是	string	司机身份证号
FDescription	是	string	描述
FCountryResidenc	是	string	城市
FDistrictRegion	是	string	城镇
FHeadPortrait	否	string	头像Base64图片,传空字符串,表示清空原来的信息,传null或者字段不传,则表示保证原来的字段内空
FDrivingLicenseFront	否	string	驾驶证正面照Base64图片,传空字符串,表示清空原来的信息,传null或者字段不传,则表示保证原来的字段内空
FDrivingLicenseBack	否	string	驾驶证反面照Base64图片,传空字符串,表示清空原来的信息,传null或者字段不传,则表示保证原来的字段内空
FAgentGUID	否	string	公司唯一标识,该字段不传,则默认为当前用户所在公司

**返回示例:**

**正确时返回:**

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"UpdateAdminDriver\",\"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\", \"FT_AdminDriverParam\":{\"FGUID\":\"4d66002d-5554-4be1-a55f-d6e9e01b6f8b\", \"FDriverName\":\"12321\", \"FLicenseNumber\":\"12321333\", \"FDrivingType\":\"C\", \"FDriverPhone\":\"5481166\", \"FDescription\":\"ddb\", \"FIDNumber\":\"323432543543\", \"FCountryResidenc\":\"322\", \"FDistrictRegion\":\"211\", \"FHeadPortrait\":\"\", \"FDrivingLicenseFront\":\"\", \"FDrivingLicenseBack\":\"\"}}";
```

```

URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-Alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.ge
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"UpdateAdminDriver\", \"FTokenID\": \"3acef045-d
302-4032-b40a-d9ee6c1519cd\", \"FT_AdminDriverParam\": { \"FGUID\": \"4d66002
d-5554-4be1-a55f-d6e9e01b6f8b\", \"FDriverName\": \"12321\", \"FLicenseNumbe
r\": \"12321333\", \"FDrivingType\": \"C\", \"FDriverPhone\": \"5481166
\", \"FDescription\": \"ddb\", \"FIDNumber\": \"323432543543\", \"FCo
untryResidenc\": \"322\", \"FDistrictRegion\": \"211\", \"FHeadPortra
it\": \"\", \"FDrivingLicenseFront\": \"\", \"FDrivingLicenseBack\": \"\"}}";

Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";

```

```

request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'UpdateAdminDriver',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FT_AdminDriverParam': '{
        'FGUID': '4d66002d-5554-4be1-a55f-d6e9e01b6f8b',
        'FDriverName': '12321',
        'FLicenseNumber': '12321333',
        'FDrivingType': 'C',
        'FDriverPhone': '5481166',
        'FDescription': 'ddb',
        'FIDNumber': '323432543543',
        'FCountryResidenc': '322',
        'FDistrictRegion': '211',
        'FHeadPortrait': '',
        'FDrivingLicenseFront': '',
        'FDrivingLicenseBack': ''
    }'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
        r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')

```

修改司机

```
# json_array = json.loads(page)
return page;
```

# 删除司机

## 简要描述：

- 删除司机信息

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2026-03-02	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (DeleteAdminDriver)
FGUIDs	是	string	司机信息唯一标识,多个以英文逗号隔开

## 返回示例：

### 正确时返回：

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": []
}
```

**错误时返回:**

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

**返回参数说明:**

参数名	类型	说明
-----	----	----

**备注:**

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误

**请求示例:****Java:**

```
String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"DeleteAdminDriver\",\"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\", \"FGUIDs\": \"0edd0b34-f99b-4ec4-b455-c3545a014d1f,a1fff4c5-4933-495c-9d06-748f566eb7c0\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; W
```

```

indows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.get
tInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"DeleteAdminDriver\", \"FTokenID\": \"3acef045-d
302-4032-b40a-d9ee6c1519cd\", \"FGUIDs\": \"0edd0b34-f99b-4ec4-b455-c3545a0
14d1f,a1fff4c5-4933-495c-9d06-748f566eb7c0\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

}

**Python :**

```
url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'DeleteAdminDriver',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FGUIDs': '0edd0b34-f99b-4ec4-b455-c3545a014d1f,a1fff4c5-4933-495c-9d
06-748f566eb7c0'
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.loads(page)
return page;
```

# 查询司机列表

## 简要描述：

- 查询司机信息列表

## 接口版本：

版本号	制定人	制定日期	修订日期
2.0.0	lenny	2026-03-02	

## 请求URL：

- 国际客户：<https://icloud.assetscontrols.com:3443/OpenApi/Admin>
- 国内客户：<https://cloud.assetscontrols.com:3443/OpenApi/Admin>

## 请求方式：

- POST

## 请求头：

参数名	是否必须	类型	说明
Content-Type	是	string	请求类型： application/json

## 请求参数：

参数名	是否必须	类型	说明
FTokenID	是	string	令牌ID
FAction	是	string	方法名 (QueryAdminDriverList)
FPageIndex	是	int	页数
FPageSize	是	int	每页数量
FKey	否	string	模糊搜索字段,按司机名或者司机手机号查
FAgentGUID	否	string	公司唯一标识,该字段不传,默认查询该账户所在公司

## 返回示例:

### 正确时返回:

```
{
  "Result": 200,
  "Message": "check token success",
  "FObject": {
    "Table": [
      {
        "FTotalCount": 3 //分页总数量
      }
    ],
    "Table1": [
      {
        "FGUID": "e7eb61fb-3e54-4f65-9748-cd6fde97086a",
        "FDriverName": "suly",
        "FLicenseNumber": "粤8542111",
        "FDrivingType": "A1",
        "FDriverPhone": "15625215199",
        "FDescription": "20240228测试23",
        "FAgentGUID": "661e7209-c77e-4a54-afb8-6d956f962bdf",
        "FCreateTime": "2024-02-20T05:51:57.733",
        "FCountryResidenc": "深圳市",
        "FDistrictRegion": "宝安区",
        "FIDNumber": "415252632122232233",
        "FDrivingLicenseFrontUrl": "http://120.25.245.20:8081/Picture/Driver/2a325bf851584f7bb1365a86acc6a5fc.jpg",
        "FHeadPortraitUrl": "",
        "FDrivingLicenseBackUrl": "",
        "RowNo": 2,
        "FAgentName": "软件测试部"
      }
    ]
  }
}
```

### 错误时返回:

```
{
  "Result": 102,
  "Message": "Action is error",
  "FObject": []
}
```

## 返回参数说明:

参数名	类型	说明
FGUID	string	司机信息唯一标识
FDriverName	string	司机名称
FLicenseNumber	string	驾驶证编号
FDriverPhone	string	司机电话
FIDNumber	string	司机身份证号
FDescription	string	描述
FCountryResidenc	string	城市
FDistrictRegion	string	城镇
FHeadPortraitUrl	string	头像图片Url
FDrivingLicenseFrontUrl	string	驾驶证正面照片Url
FDrivingLicenseBackUrl	string	驾驶证反面照片Url
FAgentGUID	string	公司唯一标识
FAgentName	string	公司名称

## 备注:

- 更多返回错误代码如下:
- 104: token错误或过期
- 105: 系统异常
- 102: 请求参数错误
- 106: 有设备号为空
- 107: 有已存在的设备ID
- 108: 有车辆已绑定设备
- 109: 有错误的设备类型

## 请求示例:

### Java:

```

String result = "";
//请求路径
String url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
String body = "{\"FAction\":\"QueryAdminDriverList\", \"FAgentGUID\":\"\", \"FKey\":\"su\", \"FPageIndex\":1, \"FPageSize\":50, \"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
URL realUrl = new URL(url);
// 设置通用请求的属性
URLConnection conn = realUrl.openConnection();
conn.setRequestProperty("accept", "*/*");
conn.setRequestProperty("connection", "keep-alive");
conn.setRequestProperty("Content-Type", "application/json");
conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)");
conn.setRequestProperty("method", "post");
// 发送POST请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
PrintWriter pw = new PrintWriter(conn.getOutputStream());
// 发送请求参数
pw.print(body);
// flush输出流的缓冲
pw.flush();
// 定义BufferedReader输入流来读取URL的响应
BufferedReader bufReader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
// 定义BufferedReader输入流来读取URL的响应
String line;
while ((line = bufReader.readLine()) != null) {
    result += line;
}
//返回的是json字符串
return result;

```

**C# :**

```

//请求路径
string url = "https://cloud.assetscontrols.com:3443/OpenApi/Admin";
//请求参数 ,json格式参数,建议用对象传入
string body = "{\"FAction\":\"QueryAdminDriverList\", \"FAgentGUID\":\"\", \"FKey\":\"su\", \"FPageIndex\":1, \"FPageSize\":50, \"FTokenID\":\"3acef045-d302-4032-b40a-d9ee6c1519cd\"}";
Encoding encoding = Encoding.UTF8;
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

```

```

//请求方式 post / get
request.Method = "post";
request.Accept = "*/*";
request.ContentType = "application/json";
request.UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)";
byte[] buffer = encoding.GetBytes(body);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
using (StreamReader reader = new StreamReader(response.GetResponseStream(),
    Encoding.UTF8))
{
    //返回的是json字符串
    return reader.ReadToEnd();
}

```

### Python :

```

url = 'https://cloud.assetscontrols.com:3443/OpenApi/Admin'
data = {
    'FAction': 'QueryAdminDriverList',
    'FTokenID': '3acef045-d302-4032-b40a-d9ee6c1519cd',
    'FKey': 'su',
    'FPageIndex': 1
    'FPageSize': 20
}
data = parse.urlencode(data).encode('utf-8')
headers = {
    'User-Agent': r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '
                r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3',
    'Connection': 'keep-alive'
}
req = request.Request(url, headers=headers, data=data)
page = request.urlopen(req).read()
page = page.decode('utf-8')
# json_array = json.Loads(page)
return page;

```