

OpenApi 接口说明

概要：

此文档为第一期产品需求所需接口。主要包括电桩状态信息，空置电桩的启动/停止，以及用户充电消费。

接口调用方式：

为保证数据安全，所有交互均采用 HTTPS(POST)方式调用。

注意:所有参数名均采用小写方式，忽略文档中格式导致的大写。所有参数采用 ASCII 码编码，不可使用别的编码方式。

签名说明：

采用签名的方式对数据合法性进行校验，开发平台为接入商提供 `AppId,Token,EncodingAESKey`。

`AppId`:为固定 24 位，从 a-z,A-Z,0-9 字符组成，为运营商唯一标识,在 api 调用中传递。

`Token`:为固定 32 位，从 a-z,A-Z,0-9 字符组成，为签名密钥。

`EncodingAESKey`: 长度固定为 43 个字符，从 a-z,A-Z,0-9 字符出组成，为数据加密密钥。

签名说明：

签名生成分以下三步:

Step1. 构造原串：

源串构造步骤如下：

1: 将除“sig”外的所有参数按 key 进行字典升序排序。

2: 将排序后的参数(key=value)用&拼接起来。

URL 编码规则：签名验证时，要求对字符串中除了“-","_", "."之外的所有非字母数字字符都替换成百分号(%)后跟两位十六进制数。十六进制数中字母必须为大写。

Step2: 构成密钥：

在应用的 `Token`(开发平台分配)末尾添加一个字节的“&”，即 `Token&`

Step3: 声称签名

1.使用 HMAC-SHA1 加密算法，使用 step2 中得到的密钥对 step1 得到的源串加密

2.然后将加密后的字符串经过 Base64 编码

签名构造实例如下：

(此处为了测试使用，各个语言可参考此来验证签名函数是否正确)

1.原始请求信息:

`Token=228bf094169a40a3bd188ba37ebe8723`

请求参数:

`app_id=1111111111&info=aaaa`

2.下面开始构造源串

第 1 步：将除 sig 外的所有参数按照 key 进行字典升序排列，排列结果为： `app_id, info`

第 2 步：将第 1 步得到的所有参数(key=value)用&拼接起来：
app_id=1111111111&info=aaaa

第 3 步：构造密钥：得到密钥的方法：在应用的 appkey 末尾添加一个字节的"&"，即 Token&，例如：
228bf094169a40a3bd188ba37ebe8723&

第 4 步：生成签名：使用 HMAC-SHA1 加密算法，使用第 3 步得到的密钥对第 2 步的源串进行加密，然后将加密后的算法进行 Base64 编码：
P8B2OK/f/HK6Wlcb3cSpsP7kfO8=

不同语言实现方式：

```
python: hmac.new(token,data,hashlib.sha1).digest().encode('base64').rstrip()
php: base64_encode(hash_hmac("SHA1",clientStr,Token , true))
c++(openssl): HMAC( EVP_sha1(),
    /*key data*/ strKey.data(),
    /*key len*/ strKey.size(),
    /*data */(unsigned char*) strRandom.data(),
    /*data len*/ strRandom.size(), digest, &digest_len)
```

消息加密说明：

1. 开放平台为每一个接入者分配唯一的 EncodingAESKey。长度固定为 43 个字符，从 a-z,A-Z,0-9 共 62 个字符中选取。
2. AES 密钥：AESKey=Base64_Decompose(EncodingAESKey + "=")，EncodingAESKey 尾部填充一个字符的"="，用 Base64_Decompose 生成 32 个字节的 AESKey；
3. AES 采用 CBC 模式，密钥长度为 32 个字节，数据采用 PKCS#7 填充；PKCS#7：K 为密钥字节数（采用 32），buf 为待加密的内容，N 为其字节数。Buf 需要被填充为 K 的整数倍。在 buf 的尾部填充(K-N%K)个字节，每个字节的内容是(K- N%K)；具体参考：<http://tools.ietf.org/html/rfc2315>
4. 所有接口中的 info 字段均需要进行加密。开放平台会提供 c++/python/php/java/c#封装接口。

全局返回码说明：

所有接口调用返回值(ret)均使用该规范

返回码(4 个字节)	说明
-1	系统繁忙，此时请第三方稍后再试
0	请求成功
4001	签名错误，此时需要验证是否按照规定进行签名的生成
4002	不合法的 AppID,请到开放平台申请 AppId 和签名 Key

4003	POST 参数不合法,请确认必须参数是否缺失
4004	POST 参数类型不合法, 请确认参数类型
4005	桩编码不合法
6001	系统错误

1. 充电桩实时状态上报

- URL: 联系技术对接人获取
- (智能模块和充电桩协议新 20150626)对应充电机协议:ID=0x28. 当收到 0x28 协议时需要调用该接口上报桩状态信息。主动召唤时通过此回调。
- HTTP 请求方式: POST
- 输入参数说明(各参数均进行 URL 编码)

参数名称	是否必须	类型	描述
app_id	是	char(24)	应用 ID(开放平台分配)
info	是	varchar(json)	桩状态信息(json 格式)
sig	是	char	签名

info 桩状态信息:

字段名	是否必须	字段类型	说明
pile_code	是	char(16)	桩编码 id
Inter_no	是	int4(4 个字节)	充电接口编码(1 表示 A 口, 2 表示 B 口)
Inter_type	是	int4(4 个字节)	充电接口类型(1 交流, 2 直流)
Inter_conn_state	是	int4(4 个字节)	接口连接状态(3:车连接, 1:空置, 2:未知)
Inter_work_state	是	int4(4 个字节)	接口工作状态(1:充电, 2:待机, 3:故障, 4:充电结束,5: 被预约, 6:暂停充电)
Inter_order_state	是	int4(4 个字节)	预约状态(1:无预约, 2:有预约)
voltage	是	float(4 个字节)	输出电压
current	是	float(4 个字节)	输出电流
soc	是	int4(4 个字节)	当前的 SOC
elect_type	否	int4(4 个字节)	电表类型(1:直流, 2:交流)

<u>elect_address</u>	查	<u>char(32)</u>	电表地址(ASCII 码)
elect_rate	否	int4(4 个字节)	电表倍率
active_power	否	float(4 个字节)	有功功率
reactive_power	否	float(4 个字节)	无功功率
active_energy	否	float(4 个字节)	电表有功电能
reactive_energy	否	float(4 个字节)	电表无功电能
fault_code	是	int4(4 个字节)	故障码(0:急停故障, 1:电表故障, 2:接触器故障, 3:读卡器故障, 4:内部过温故障, 5:连接器故障, 6:绝缘故障, 7:其他)注:需要将电桩内部故障码转换成以上故障码
err_code	是	int4(4 个字节)	错误码(0:电流异常, 1:电压异常, 2:其他)
res_time	是	int4(4 个字节)	剩余充电时间
parking_state	否	int4(4 个字节)	车位状态:0:未知 1:空闲 2:占用 3:故障
time	是	int4(4 个字节)	上报时时间戳

info 举例:

```
{
  'pile_code':'xxxxx',
  'Inter_no':'xxxx,
  .....
}
```

- 返回结果(JSON 格式)

参数名称	说明
ret	参见全局返回码
msg	如果错误, 返回错误信息

2. 召唤桩状态信息[第三方提供] :

- URL:[域名]/xxxxx
- 对应充电协议: 待定
- HTTP 请求方式: POST
- 输入参数

参数名称	是否必须	参数类型	说明
------	------	------	----

app_id	是	char(32)	应用 id
info	是	json	桩信息
sig	是	varchar	签名

info 具体信息:

字段名	是否必须	参数类型	说明
pile_code	是	char(16)	桩统一编码
inter_no	是	int4(4 个字节)	桩接口

• 返回结果[JSON]

ret	参见全局返回码
msg	错误时返回错误信息

3. 电桩启动/停止操作[第三方提供]

- URL: [域名]/xxx
- 对应充电机协议:ID=0x44
- HTTP 请求方式: POST
- 输入参数

参数名称	是否必须	参数类型	说明
app_id	是	char(32)	用户 ID
info	是	json	请求参数
sig	是	varchar	签名

info 参数信息 (可批量上传):

输入参数	是否必须	参数类型	说明
session_id	是	longlong(8 个字节)	此次操作的 session_id(可作为交易流水号)
pile_code	是	char(16)	充电桩 ID 编码
inter_no	是	int4(4 个字节)	充电桩接口
action	是	int4(4 个字节)	1:启动 2:停止
user_id	是	int4(4 个字节)	用户 ID(消费单回调时需要带上该参数, 标示哪个用户消费)
voltage	是	int4(4 个字节)	开关电压
elect	是	int4(4 个字节)	开关电流

time	是	int4(4 个字节)	上报时间戳
------	---	-------------	-------

info 举例:

```
{
'pile_code':'xxxx',
'session_id':yyy,
.....
}
```

- 返回结果(JSON 格式)

ret	参见全局返回码
msg	如果错误, 返回错误信息

4. 电桩启动/停止回调接口

- URL: 联系技术对接人获取
- HTTP 请求方式:POST

- 输入参数

参数名称	是否必须	参数类型	说明
app_id	是	char(32)	应用 ID
info	是	json	结果信息
sig	是	varchar	签名

info 信息:

参数名称	是否必须	参数类型	说明
session_id	是	longlong(8 个字节)	此次操作对应的 session_id
pile_code	是	char(16)	充电桩编码 id
inter_no	是	int4(4 个字节)	电桩接口
user_id	是	int4(4 个字节)	调用启动/停止接口传递的用户 id
action	是	int4(4 个字节)	电桩操作: 1:启动 2:停止
result	是	int4(4 个字节)	1:设置成功 2:设置失败
ecode	是	int4(4 个字节)	错误码:0x00:无错误 0x01: 已经开机 0x02: 未开机 0x03: 枪未连接 0x04:其他错误
soc	是	int4(4 个字节)	车的 soc

time	是	int4(4 个字节)	上报时间戳
------	---	-------------	-------

info 举例:

```
{
  'pile_code':'xxxx',
  'session_id':yyy,
  .....
}
```

- 返回结果(JSON 格式)

ret	参见全局返回码
msg	如果错误, 返回错误信息

5. 充电桩实时充电数据

- URL: 联系技术对接人获取
- 对应充电机协议: ID=0x2C. 消费金额由电桩计算. 停机原因需要合并到此接口中, 如果是停机则 stop=1 否则为 0.

- HTTP 请求方式: POST

- 输入参数

参数名称	是否必须	参数类型	说明
app_id	是	char(32)	应用 ID
info	是	json	数据信息
sig	是	varchar	签名

info 信息:

参数名称	是否必须	参数类型	说明
session_id	是	longlong(8 个字节)	启动时对应的 session_id
pile_code	是	char(16)	电桩编码 id
inter_no	是	int4(4 个字节)	充电接口编码
user_id	是	int4(4 个字节)	启动电桩时传递的 user_id
cur_elect	是	float(4 个字节)	当前充电总电量
cur_money	是	float(4 个字节)	当前总消费金额(电费+服务费)
cur_elect_money	是	float(4 个字节)	当前充电电费金额
cur_service_money	是	float(4 个字节)	当前服务费金额
cur_time	是	int4(4 个字节)	当前充电时间

soc	是	int4(4 个字节)	当前 soc
stop	是	int4(4 个字节)	1:未停机 2:停机
stop_reason	是	int4(4 个字节)	停机原因,当 stop=2 时返回 1: 故障 2 充满 3 刷卡 4 其他
time	是	int4(4 个字节)	上报时间戳

info 格式举例:

```
{
  'pile_code':'xxx',
  'session_id':'yyy',
  .....
}
```

- 返回结果(JSON 格式)

ret	参见全局返回码
msg	如果错误, 返回错误信息

6. 桩实时充电数据召唤接口 [第三方接口]

- URL:[域名]/xxxxx
- 对应充电协议: 待定
- HTTP 请求方式: POST
- 输入参数

参数名称	是否必选	参数类型	说明
app_id	是	char(32)	应用 id
info	是	json	具体信息
sig	是	varchar	签名

info 详细信息

参数名称	是否必须	参数类型	说明
pile_code	是	char(16)	桩统一编码
inter_no	是	int4(4 个字节)	桩接口

- 返回结果[JSON] 桩实际状态数据通过实时数据上报返回

ret	参见全局返回码
msg	如果错误，返回错误信息

7. 充电完成消费单通知

- URL: 联系技术对接人获取
- 对应充电协议: ID=0x2E
- HTTP 请求方式: POST
- 输入参数

参数名称	是否必须	参数类型	说明
app_id	是	char(32)	App 唯一标示
info	是	string(json)	订单信息
sig	是	varchar	签名

订单信息

字段名	是否必须	字段类型	
pile_code	是	char(16)	桩 ID 编码
session_id	是	longlong(8 个字节)	此次充电启动时对应的 session_id
user_id	是	int4(4 个字节)	用户 ID(调用启动/停止充电时传递的 user_id)
money	是	float(4 个字节)	本次充电消费总金额
elect_money	是	float(4 个字节)	本次充电电费总金额
service_money	是	float(4 个字节)	本次充电服务费金额
elect	是	float(4 个字节)	充电电量
start_elect	是	float(4 个字节)	开始充电电量
end_elect	是	float(4 个字节)	结束充电电量
culp_elect	是	float(4 个字节)	尖阶段电量
culp_elect_price	是	float(4 个字节)	尖电价价格
culp_service_price	是	float(4 个字节)	尖服务费单价
culp_money	是	float(4 个字节)	尖总金额
culp_elect_money	是	float(4 个字节)	尖充电金额
culp_service_money	是	float(4 个字节)	尖服务费金额
peak_elect	是	float(4 个字节)	峰阶段电量

peak_elect_price	是	float(4 个字节)	峰电价价格
peak_service_price	是	float(4 个字节)	峰服务费单价
peak_money	是	float(4 个字节)	峰总金额
peak_elect_money	是	float(4 个字节)	峰充电金额
peak_service_money	是	float(4 个字节)	峰服务费金额
flat_elect	是	float(4 个字节)	平阶段电量
flat_elect_price	是	float(4 个字节)	平阶段电价
flat_service_price	是	float(4 个字节)	平阶段服务费单价
flat_money	是	float(4 个字节)	平总金额
flat_elect_money	是	float(4 个字节)	平充电金额
flat_service_money	是	float(4 个字节)	平 服务费金额
valley_elect	是	float(4 个字节)	谷阶段电量
valley_elect_price	是	float(4 个字节)	谷阶段电价
valley_service_price	是	float(4 个字节)	谷阶段服务费单价
valley_money	是	float(4 个字节)	谷总金额
valley_elect_money	是	float(4 个字节)	谷充电金额
valley_service_money	是	float(4 个字节)	谷服务费金额
start_time	是	int(4 个字节)	充电开始时间
end_time	是	int(4 个字节)	充电结束时间
stop_model	是	int(4 个字节)	停止时充电模式
stop_reason	是	int(4 个字节)	停止充电原因
soc	是	int(4 个字节)	soc
time	是	int(4 个字节)	时间戳

info 格式举例:

```
{
  'pile_code':'xxx',
  'session_id':'yyy',
  .....
}
```

- 返回结果(JSON 格式)

ret	参见全局返回码
msg	如果错误, 返回错误信息

8. 消费记录的召唤[第三方提供]

- URL: [域名]/xxxx
- 对应充电协议: ID=0x
- HTTP 请求方式: POST
- 输入参数:

参数名称	是否必须	参数类型	说明
app_id	是	char(32)	应用 ID
info	是	json	请求信息
sig	是	varchar	签名

info 信息:

参数名称	是否必须	参数类型	说明
pile_code	是	char(16)	桩编码
session_id	是	longlong(8 个字节)	账单对应的 session_id
time	是	int(4 个字节)	时间戳

- 返回: (json 格式)

ret	参见全局返回码
msg	如果错误, 返回错误信息