OpenApi(V2.7)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。

接口调用注意事项：

1、所有接口均采用 HTTPS(POST)方式调用（测试环境可能为 HTTP 服务器而非 HTTPS）。

2、所有参数名均采用小写方式，忽略文档中格式导致的大写。

3、所有参数采用 ASIC 码编码，正常情况不应包含中文等非 ASCII 字符。

4、http 请求的所有参数需要进行 URL 编码。例如签名中的“+”符号 URL 编码后为%2B，如不进行编码可能导致签名校验失败。

5、接口中涉及到的时间参数，如无特殊说明则均为 Unix 时间戳形式，秒格式（而不是毫秒格式），10 位数字。如 2016 年 12 月 2 日 12:00:00 转为为时间戳为 1480564800。可使用以下工具在线转换和测试：http://tool.chinaz.com/Tools/unixtime.aspx

签名说明：

采用签名的方式对数据合法性进行校验，e 充网在线下为接入的充电桩运营企业（以下简称运营商）提供 app_id,app_key。

app_id:为 24 位的字符串，由 a-z,A-Z,0-9 字符组成，为运营商唯一标识,在 api 调用时传递。

app_key:为 32 位的字符串，由 a-z,A-Z,0-9 字符组成，用以产生签名密钥。

签名生成：

签名生成分以下三步:

1. 构造源串：

源串构造步骤如下：

1.1: 将除“sig”外的所有参数按 key 进行字典升序排序。

1.2: 将排序后的参数(key=value)用&拼接起来。

2: 产生密钥：

在应用的 app_key(e 充网分配的)末尾添加一个“&”符号, 即 app_key&。

3: 生成签名

3.1.使用 HMAC-SHA1 哈希算法，将 step2 中得到的密钥和 step1 得到的源串输入算法。

3.2.将哈希算法运算的结果进行 Base64 编码，即为签名。

签名构造实例如下：

（此处为了测试使用，各个语言可参考此结果来验证签名函数是否正确）

1.e 充网分配的 app_key:

app_key=VjaOvn7zkukdmbeG8oplUj3WoJSm4TN1

请求参数：

app_id=qjx6TcPLpgHZv4ZH5mJ2K6qj

info={"pile_code":"1110108217001001"}

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=qjx6TcPLpgHZv4ZH5mJ2K6qj&info={"pile_code":"1110108217001001"}

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”&”符号，即 app_key&,例如：
VjaOvn7zkukdmbeG8op1Uj3WoJSm4TN1&

第 4 步：生成签名：使用 HMAC-SHA1 哈希算法，将第 3 步得到的密钥和第 2 步的源串输入算法进行运算，然后将生成的结果进行 Base64 编码，得到：
zD2MxjTOaHh/bqmwgbUxl7s9m5M=

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))
另外可直接使用压缩包中提供的签名工具代码。

数据加密：

接口直接采用 https 方式进行调用，所以数据加密通过 ssl 在网络层和应用层之间进行，来保证数据的安全性。

业务流程：

e 充网提供 4 个接口：电桩状态上报、启动/停止回调、充电数据上报、账单上报；
充电桩运营企业需要提供 1 个接口：电桩启动/停止操作。

电桩接入 e 充网后，在平时需要通过电桩状态上报接口推送电桩状态，推送的条件：

- 在电桩的状态变化（如充电枪被插拔、工作状态变为待机或者充电中）时应该立即推送一次状态；
- 如果电桩状态一直没有变化，则应该每 15 分钟推送一次当前状态；（否则在超过 30 分钟电桩仍没有推送状态时，e 充网会认为该电桩掉线）

充电流程：
1、未充电时，电桩持续向 e 充网推送电桩状态；
2、在开始充电流程时，用户先将充电枪连接汽车。此时由于状态变化，充电桩需要向 e 充网推送当前状态；
3、用户通过 APP 向 e 充网发起开始充电指令，e 充网判断，如果电桩的状态是充电枪已连接（电桩状态中 inter_conn_state=3）、待机状态（inter_work_state=2），则向运营商发起启动电桩的请求；

4、运营商收到启动电桩请求，如果确认正常可以开启，则应该返回成功（ret=0），并开始启动电桩；
5、电桩启动之后，运营商需要向 e 充网的“电桩启动/停止回调”接口发送启动回调；
6、随后正常充电过程中，充电桩需要通过“电桩充电数据上报”接口向 e 充网推送充电数据，推送的频率以 2 分钟每次为宜。（在充电过程中，电桩状态仍需要正常推送）
7、用户向 e 充网发出停止充电指令，e 充网判断如果电桩在充电状态，则向运营商发起停止电桩的请求；
8、运营商收到请求，如果正常可以停止，则返回成功（ret=0），并停止电桩，在成功停止后向 e 充网发送停止回调；（类似第 4、5 步骤）
9、此时一次充电完毕，运营商需要在停止充电后通过“充电完成账单上报”接口向 e 充网发送账单。
10、如果在充电中因为充电枪被拔出、电量充满等原因导致停止充电，则认为一次充电完毕，运营商应该同第 9 步骤向 e 充网发送账单。

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(int 型)	说明
-1	系统繁忙，此时请调用方稍后再试
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 app_id,请到 e 充网开放平台申请 app_id 和 app_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 pile_code 不合法
6001	系统错误

1. 电桩状态上报

- URL: 见“e 充网 OpenApi 接口测试和正式地址 V2.7.txt”文档
- 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	string	运营商标识 ID(e 充网分配的)
info	是	string(json)	电桩状态信息(json 字符串)
sig	是	string	签名

info 桩状态信息：

字段名	是否必须	字段类型	说明
pile_code	是	string	充电桩编码
inter_no	是	int	电桩上的接口(1 表示 A 口, 2 表示 B 口)
inter_type	是	int	充电电流类型(1 交流，2 直流)
inter_conn_state	是	int	充电枪连接状态(1:空置，2:未知，3:车连接)
inter_work_state	是	int	电桩接口工作状态(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	int	预约状态(1:无预约，2:有预约)
voltage	是	float	输出电压
current	是	float	输出电流
soc	是	int	当前的 SOC（汽车电量的百分比）
elect_type	否	int	电表类型(1:直流, 2:交流)
elect_address	否	string	电表地址(ASCII 码)
elect_rate	否	int	电表倍率
active_power	否	float	有功功率
reactive_power	否	float	无功功率
active_energy	否	float	电表有功电能
reactive_energy	否	float	电表无功电能
fault_code	是	int	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。无故障则传 7
err_code	是	int	错误码(0:电流异常，1:电压异常，2:其他)无错误则传 2
res_time	是	int	剩余充电时间，单位分钟。未在充电状态则传 0
parking_state	否	int	车位状态:0:未知 1:空闲 2:占用 3:故障
time	是	int	上报时间（秒格式 Unix 时间戳）

info 参数举例见文档末尾。

- 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. 电桩启动/停止操作[需要由运营商提供]

- URL: 由运营商确定，并提供给 e 充网
- 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	string	运营商标识 ID(e 充网分配的)
info	是	string(json)	启动/停止相关信息(**注意此处为 json 数组字符串，**[{"pile_code":"1110108217001001"}]形式，而非{"pile_code":"1110108217001001"}形式，具体见文档末尾示例)
sig	是	string	签名

info 参数信息 (可批量上传，即 info json 数组包含多个启动动作)：

输入参数	是否必须	参数类型	说明
session_id	是	long	此次操作的 session_id(可作为交易流水号)
pile_code	是	string	充电桩编码
inter_no	是	int	电桩上的接口(1 表示 A 口, 2 表示 B 口)
action	是	int	1:启动 2:停止
user_id	是	int	用户 ID,后续其它数据上报时可能需要该 user_id
voltage	是	int	开关电压
elect	是	int	开关电流
time	是	int	上报时间（秒格式 Unix 时间戳）

info 参数举例见文档末尾。

- 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

3. 电桩启动/停止回调接口

- URL: 见“e 充网 OpenApi 接口测试和正式地址 V2.7.txt”文档
- 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	string	运营商标识 ID(e 充网分配的)
info	是	string(json)	回调信息(json 字符串)
sig	是	string	签名

info 信息：

参数名称	是否必须	参数类型	说明
session_id	是	long	启动/停止操作对应的 session_id
pile_code	是	string	充电桩编码
inter_no	是	int	电桩上的接口(1 表示 A 口, 2 表示 B 口)
user_id	是	int	e 充网调用启动/停止接口时传递的用户 id
action	是	int	电桩操作：1:启动 2:停止
result	是	int	1:操作成功 2:操作失败
ecode	是	int	错误码:0:无错误 1: 已经开机 2: 未开机 3: 枪未连接 4:其他错误
soc	是	int	当前的 soc（汽车电量的百分比，范围 0-100）
time	是	int	上报时间（秒格式 Unix 时间戳）

info 参数举例见文档末尾。

- 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. 电桩充电数据上报

- URL: 见“e 充网 OpenApi 接口测试和正式地址 V2.7.txt”文档
- 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	string	运营商标识 ID(e 充网分配的)
info	是	string(json)	充电数据信息(json 字符串)
sig	是	string	签名

info 信息:

参数名称	是否必须	参数类型	说明

session_id	是	long	启动时对应的 session_id
pile_code	是	string	充电桩编码
inter_no	是	int	电桩上的接口(1 表示 A 口, 2 表示 B 口)
user_id	是	int	启动电桩时传递的 user_id
cur_elect	是	float	当前已经充电的电量
cur_money	是	float	当前总消费金额(电费＋服务费)
cur_elect_money	是	float	当前充电电费金额
cur_service_money	是	float	当前服务费金额
cur_time	是	int	当前已充电时间，单位分钟
soc	是	int	当前的 soc（汽车电量的百分比，范围 0-100）
stop	是	int	1:未停机 2:停机
stop_reason	是	int	停机原因,1：故障 2 充满 3 刷卡 4 其他，未停机时传 4
time	是	int	上报时间（秒格式 Unix 时间戳）

info 参数举例见文档末尾。

- 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. 充电完成账单上报

- URL: 见“e 充网 OpenApi 接口测试和正式地址 V2.7.txt”文档
- 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	string	运营商标识 ID(e 充网分配的)
info	是	string(json)	订单信息(json 字符串)
sig	是	string	签名

订单信息

字段名	是否必须	字段类型	
pile_code	是	string	充电桩编码

session_id	是	long	此次充电启动时对应的 session_id
user_id	是	int	用户 ID(调用启动/停止充电时传递的 user_id)
money	是	float	本次充电消费总金额（电费+服务费）
elect_money	是	float	本次充电电费金额
service_money	是	float	本次充电服务费金额
elect	是	float	充电电量
start_elect	是	float	开始充电电量
end_elect	是	float	结束充电电量
cusp_elect	是	float	尖阶段电量
cusp_elect_price	是	float	尖电价价格
cusp_service_price	是	float	尖服务费单价
cusp_money	是	float	尖总金额
cusp_elect_money	是	float	尖充电金额
cusp_service_money	是	float	尖服务费金额
peak_elect	是	float	峰阶段电量
peak_elect_price	是	float	峰电价价格
peak_service_price	是	float	峰服务费单价
peak_money	是	float	峰总金额
peak_elect_money	是	float	峰充电金额
peak_service_money	是	float	峰服务费金额
flat_elect	是	float	平阶段电量
flat_elect_price	是	float	平阶段电价
flat_service_price	是	float	平阶段服务费单价
flat_money	是	float	平总金额
flat_elect_money	是	float	平充电金额
flat_service_money	是	float	平 服务费金额
valley_elect	是	float	谷阶段电量
valley_elect_price	是	float	谷阶段电价
valley_service_price	是	float	谷阶段服务费单价
valley_money	是	float	谷总金额
valley_elect_money	是	float	谷充电金额
valley_service_money	是	float	谷服务费金额

start_time	是	int	充电开始时间（秒格式 Unix 时间戳）
end_time	是	int	充电结束时间（秒格式 Unix 时间戳）
stop_model	是	int	停止时充电模式，1 表示恒压，2 表示恒流。
stop_reason	是	int	停止充电原因,1：故障 2 充满 3 刷卡 4 其他 收到 e 充网的停止请求而停止则传 4
soc	是	int	当前的 soc（汽车电量的百分比，范围 0-100）
time	是	int	订单创建时间（秒格式 Unix 时间戳）

info 参数举例见文档末尾。

- 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

info 参数 json 字符串示例：

电桩状态：


```
{
 "pile_code": "1110108217001001",
 "inter_no": 0,
 "inter_type": 2,
 "inter_conn_state": 3,
 "inter_work_state": 2,
 "inter_order_state": 1,
 "voltage": 5,
 "current": 9,
 "soc": 21,
 "fault_code": 22,
 "err_code": 0,
 "res_time": 0,
 "time": 1480417165,
 "elect_address": "none",
 "elect_type": 2,
 "elect_rate": 0,
 "active_power": 16,
 "reactive_power": 17,
 "active_energy": 18,
 "reactive_energy": 19,
 "parking_state": 2
}
```


启动/停止请求：

```
[
 {
  "session_id": 1903,
  "pile_code": "1110108217001001",
  "inter_no": 1,
  "action": 1,
  "user_id": 228,
  "voltage": 1,
  "elect": 5,
  "time": 1480508124
 }
]
```


启动停止回调：


```
{
 "pile_code": "1110108217001001",
 "inter_no": 1,
 "session_id": 1903,
 "user_id": 228,
 "action": 1,
 "result": 1,
 "soc": 57,
 "time": 1480508124,
 "ecode": 0
}
```


充电数据：


```
{
 "pile_code": "1110108217001001",
 "inter_no": 1,
 "user_id": 35829,
 "session_id": 683121,
 "cur_elect": 13.69,
 "cur_money": 20.46,
 "stop": 1,
 "soc": 97,
 "cur_time": 51,
 "stop_reason": 0,
 "time": 1480586288,
 "cur_elect_money": 0,
 "cur_service_money": 0
}
```


账单：


```
{
 "pile_code": "1110108217001001",
 "session_id": 683222,
```

```
  "user_id": 48521,
  "money": 40.6,
  "elect_money": 20.3,
  "service_money": 20.3,
  "elect": 22.51,
  "start_elect": 70359.5,
  "end_elect": 70382,
  "cusp_elect": 0,
  "cusp_elect_price": 1.0044,
  "cusp_service_price": 0.8,
  "cusp_money": 0,
  "cusp_elect_money": 0,
  "cusp_service_money": 0,
  "peak_elect": 22.51,
  "peak_elect_price": 1.0044,
  "peak_service_price": 0.8,
  "peak_money": 0,
  "peak_elect_money": 22.6,
  "peak_service_money": 18,
  "flat_elect": 0,
  "flat_elect_price": 0.6950000000000001,
  "flat_service_price": 0.8,
  "flat_money": 0,
  "flat_elect_money": 0,
  "flat_service_money": 0,
  "valley_elect": 0,
  "valley_elect_price": 0.3946,
  "valley_service_price": 0.8,
  "valley_money": 0,
  "valley_elect_money": 0,
  "valley_service_money": 0,
  "start_time": 1480588539,
  "end_time": 1480593339,
  "stop_model": 2,
  "stop_reason": 0,
  "soc": 88,
  "time": 1480593334
}
```