**T项目接口对接方案**

修订记录

|  |  |  |
| --- | --- | --- |
| 修订时间 | 主要修订内容 | 修订人 |
| 2021-12-15 | 初稿，按照M1需求定义接口 | 吴进 |

### 概述

本文档基于业务流程定义了T项目服务端、设备端及APP端接口数据交互内容，作为开发人员自测和联调验证基础。

### 名词定义

本文将针对通用的简称、名词术语进行名词定义，方便大家沟通。

|  |  |
| --- | --- |
| 设备端 | 即充电桩设备终端，执行汽车充电和于服务端进行交互 |
| 服务端 | 即云端服务器，与设备端、APP端进行通讯与交互 |
| APP | 即APP端，使用用户的智能手机终端 |

### 接口类型

为降低系统复杂度，减轻开发量。设备端于服务端仅通过OCPP报文进行通讯；服务端与APP端使用HTTP进行通讯；服务端推送状态消息给APP使用MQTT消息；

### 接口安全

服务器HTTP端：采用JWT对HTTP Header进行校验，可增加部署SSL加密。

MQTT端：使用MQTT的用户密码进行权限管控，仅APP端使用，且APP端只用于接收消息，不进行发送消息。为了实现防篡改，后续可以在消息里加上验签机制例如DES、SHA。

OCPP端：暂使用HTTP协议的websocket，未进行加密，后续可考虑增加https加密或http basic校验

### 测试工具

#### 在线HTTP API

服务器端在线swagger API，可以在线模拟数据请求<http://charge.higtek.cn/swagger-ui.html>

**参数**

Select a spec请选择MemberAPI以测试前端接口

#### 在线OCPP测试

在线websocket测试工具<http://coolaf.com/tool/chattest>

**参数**

服务器地址：ws://chargews.higtek.cn

#### 在线MQTT测试

在线mqtt测试工具<http://api.easylink.io/tools/mqtt/>

本接口使用mosquitto的websock代理网页测试，和直接连接1883端口的消息一致

可用于观察和测试MQTT消息

**参数**

Host：charge.higtek.cn  
 Port：1983

UserName：charge  
 Password：abcd1234

Subscriptions：/chargePoint/2/<设备SN>

其他默认即可

需要监视哪个设备消息推送，Subscriptions就填写哪个设备号

### 主要流程

#### 设备上电交互流程

##### 流程图



##### 流程描述

1. 设备在上电启动或重启时，会发送BootNotification报文，服务端收到报文后，将对报文里chargePointSerialNumber标记的设备进行注册，并绑定当前会话到标记的设备；
2. 设备注册完毕后，设备端收到返回，后续可发送设备的状态通知，服务端更新对应设备的状态为可用；
3. 设备端发送BootNotification后，定义发送设备心跳请求，以保持TCP长连接有效。

#### 用户短信登录/注册流程

##### 流程图



##### 流程描述

1. APP首次登录或登录有效期过期，输入手机号请求登录/注册，跳转到登录界面；
2. 登录界面发送鉴权短信，服务端发送短信给请求用户的手机；
3. APP界面输入验证码提交；
4. 如果验证码正确，则返回用户的JWT Token，后续用户请求的header均需携带此token，服务端对于未注册用户自动进行注册，用户密码为随机密码；
5. 设备端跳转至主界面。

#### 设备充电交互流程

##### 流程图



##### 流程描述

1. 用户进行充电，首先由登录用户在APP进行发起；
2. 服务器检查设备状态，如果为可用，则发送OCPP报文RemoteStartTransaction给设备端；
3. 设备端发起OCPP报文请求授权Authorize；
4. 服务器端检查用户，返回授权成功；
5. 设备端准备好充电，发起StartTransaction报文给服务器；
6. 服务端推送MQTT消息：准备充电，此时APP端跳连接MQTT服务器接收推送，并跳转至充电界面。服务器将给每个设备分配一个topic，地址为：/chargePoint/2/<设备SN>,APP订阅此topic以接收当前正在充电设备的各种通知推送；
7. 后续连接器闭合/暂停等消息通过StatusNotification报文发送给服务器，并由服务器处理并中转至MQTT；
8. APP收到MQTT订阅的报文将更新界面显示的设备状态；
9. 后续仪表值更新的消息通过StatusNotification报文发送给服务器，并由服务器处理并中转至MQTT；
10. APP收到MQTT订阅的报文将更新界面显示的设备仪表值；
11. 用户需要停止充电时，在APP端点击发起“停止”命令给服务器；
12. 服务器发送OCPP报文RemoteStopTransaction给设备端；
13. 设备端收到请求后，如果可以停止，则发送StopTransaction报文给服务器；
14. 服务器通过MQTT推送设备停止的状态给APP（此处需要讨论，如果有充电完成状态通过MQTT中转，也可以不用此推送），APP界面显示为“充电完成”，但是此时还需等待计费信息；
15. 设备端停止后，会再发送一条DataTransfer报文给服务器，类型是currentrecord。承载了交易结账的信息内容；
16. 服务器端收到消息后，推送类型为currentrecord的交易结账MQTT信息给APP，APP界面跳转到交易结账界面显示。

### 接口功能协议

#### 服务端与设备端接口

参考《OCPP协议1.6》规范的定义，交互协议为websocket，请求数据为json。

#### 服务端HTTP接口

服务端HTTP接口请求数据为application/x-www-form-urlencoded（用于传递简单结构数据）或application/json（用于POST传递复杂结构数据）。返回的格式为application/json。

详细以在线API接口<http://charge.higtek.cn/swagger-ui.html> 为准

##### 页面唯一key

本项目由于是采用JWT的验证机制，因此不使用传统的java servlet api session保存会话数据。对于会话的场景，我们采用标记会话（标记页面）的方式来记录会话消息，所有的会话将持久到redis缓存，并通过唯一的会话ID来标记。此ID的产生可由设备端UUID生成，但是更可靠的是由服务端返回唯一的标记，此接口可以请求服务器，来生成唯一的页面会话标记ID

METHOD：GET

格式：application/x-www-form-urlencoded

URI：/api/common/uuid

参数信息

|  |  |
| --- | --- |
| **Name** | **Description** |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |

返回：22位压缩UUID，用于标记页面会话，在短信请求和验证时需要

业务异常：

##### 给会员发送鉴权短信

METHOD：POST

格式：application/x-www-form-urlencoded

URI：/api/common/sendAuthCode

参数信息

|  |  |
| --- | --- |
| **Name** | **Description** |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |
| **sid \*** | 页面会话ID，建议由uuid请求生成或者页面随机算法生成uuid，需要与验证码短信请求的一致 |
| **string** |
| ***(formData)*** |
| **userMobile \*** | 手机号码 [maxlength=11] |
| **string** |
| ***(formData)*** |

返回：无

业务异常：

##### 手机短信方式会员登录

METHOD：POST

格式：application/x-www-form-urlencoded

URI：/api/common/mobileLogin

参数

|  |  |
| --- | --- |
| **Name** | **Description** |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |
| **sid \*** | 页面会话ID，建议由uuid请求生成或者页面随机算法生成uuid，需要与验证码短信请求的一致 |
| **string** |
| ***(formData)*** |
| **userMobile \*** | 手机号码 [maxlength=11] |
| **string** |
| ***(formData)*** |
| **validateCode \*** | 短信验证码 [maxlength=6] |
| **string** |
| ***(formData)*** |

返回：会员的JWT Token字符串

业务异常：短信验证码错误或已过期，请重新获取 | 用户名/密码输入不正确 | 用户{0}已冻结，请联系管理员

##### 绑定设备到当前用户，支持直接绑定和分享扫码绑定

METHOD：POST

格式：application/x-www-form-urlencoded

URI：/api/charge/bindUser

参数

|  |  |
| --- | --- |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |
| **chargePointSerialNumber \*** | 设备编号 [maxlength=38] |
| **string** |
| ***(formData)*** |
| sourceUserMobile | 分享源手机 [maxlength=11] |
| **string** |
| ***(formData)*** |

返回：无

业务异常：

##### 列出当前会员绑定的设备列表

METHOD：GET

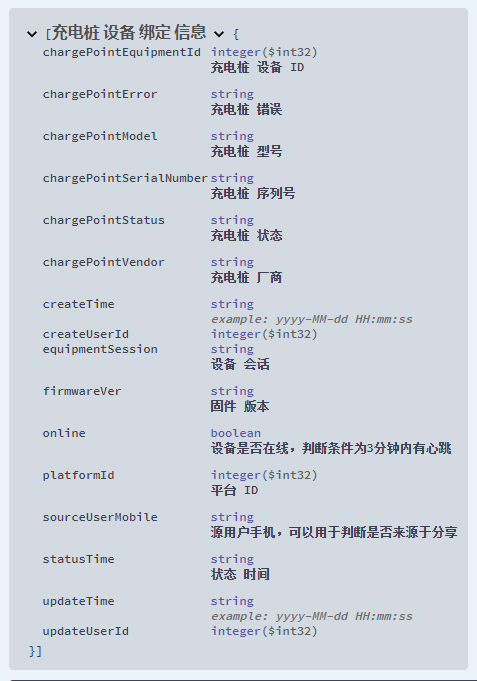
格式：application/x-www-form-urlencoded

URI：/api/common/mobileLogin

参数

|  |  |
| --- | --- |
| **Name** | **Description** |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |

返回：充电桩设备绑定信息列表，列表数据格式如下



业务异常：

##### 开始充电

METHOD：POST

格式：application/x-www-form-urlencoded

URI：/api/charge/remoteStartTransaction

参数

|  |  |
| --- | --- |
| **Name** | **Description** |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |
| **chargePointSerialNumber \*** | 设备编号 [maxlength=38] |
| **string** |
| ***(formData)*** |

返回：无

业务异常：设备不存在 | 设备故障 | 启动充电失败

##### 停止充电

METHOD：POST

格式：application/x-www-form-urlencoded

URI：/api/charge/remoteStopTransaction

参数

|  |  |
| --- | --- |
| **Name** | **Description** |
| apiToken | apiToken 前台Token |
| **string** |
| ***(header)*** |
| **chargePointSerialNumber \*** | 设备编号 [maxlength=38] |
| **string** |
| ***(formData)*** |

返回：无

业务异常：设备不存在 | 设备故障 | 停止充电失败

#### 服务端MQTT接口

##### 总体规范

服务端MQTT报文格式为json格式。类型里使用type标识不同的消息体类型。

目前推送的消息类型有3种，包括设备状态推送、充电仪表值推送、结账账单推送。

相关序列化类如下：

@Getter

@Setter

@ToString

**public** **class** ChargeMessage {

**public** **static** **final** String ***START\_CHAGE\_NOTIFY*** = "startCharge|noError"; //充电开始

**public** **static** **final** String ***STOP\_CHAGE\_NOTIFY*** = "stopCharge|noError"; //充电停止

**public** **static** **enum** Type {

***status***, //设备状态推送 格式为：主状态|子状态，除了标准的几个状态外，还增加2个状态：充电开始 和 充电停止。用于引导APP界面跳转到充电状态及充电完成跳转到结账状态

***meterValue***, //电量状态推送，状态推送时为状态代码，仪表值时为List<MeterValue>

***currentrecord*** //结账账单

}

@JsonFormat (shape = JsonFormat.Shape.***STRING***, pattern = GlobalEx.***STANDERD\_DATE\_FORMAT***)

**private** Date date;

**private** Type type;

**private** JsonNode data; //根据具体type来解析

}

##### 设备状态推送

当设备状态改变时，推送设备状态到对应设备的topic，APP通过订阅此topic获得设备状态的变更。Data格式为string，包括主状态和子状态（错误状态），状态间使用|符号间隔。例如准备工作报文，data为：”Preparing|NoError”。

##### 账单结账推送

当设备发送DataTransfer OCPP报文（类型currentrecord）给服务端时，服务端转发对应的报文到MQTT对应topic通道，APP通过订阅此topic获得仪表值的变更，相关序列化java对象定义如下（参考测试报文定义结构）：

@Getter

@Setter

@ToString

**public** **class** CurrentrecordData {

**private** Integer id;

**private** Integer connectorId;

**private** Integer chargemode;

**private** Double startengry;

**private** Double endengry;

**private** String starttime;

**private** String endtime;

**private** Double costenergy;

**private** Double costmoney;

**private** Integer transactionId;

**private** String stopReason;

}

### 附件一 OCPP测试参考报文

//登陆

[2021-11-09 15:54:38:613] Socket Connect Server Successful

Send Handshake:GET /2103252121090001 HTTP/1.1

Host:120.79.1.235:9004

Origin: http://2103252121090001

Authorization: Basic MjEwMzI1MjEyMTA5MDAwMToxMjM0NTY3OA==

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: UWR0SXNETGFNRjM0dVJMNg==

Sec-WebSocket-Protocol: ocpp1.6

Sec-WebSocket-Version: 13

WebSocket Send Handshake

receive data: HTTP/1.1 101 Web Socket Protocol Handshake

Connection: Upgrade

Date: Tue, 09 Nov 2021 07:54:39 GMT

Sec-WebSocket-Accept: G1E6fpVvRQdSnjV5XTRQU97GYTY=

Sec-WebSocket-Protocol: ocpp1.6

Server: TooTallNate Java-WebSocket

Upgrade: websocket

//启动通知

[2,"100","BootNotification",{"chargePointVendor":"TONLY","chargePointModel":"TonlyCharger\_48A","chargePointSerialNumber":"2103252121090001","firmwareVersion":"AC48A\_1PV100"}]

[3,"100",{"currentTime":"2021-11-09T15:54:39.397","interval":60,"status":"Accepted"}]

//状态通知 空闲

[2,"101","StatusNotification",{"connectorId":1,"errorCode":"NoError","status":"Available"}]

receive data: ?[3,"101",{}]

//状态通知 就绪

[2,"105","StatusNotification",{"connectorId":1,"errorCode":"NoError","status":"Preparing"}]

[2021-11-09 15:58:40:440] receive data: ?[3,"105",{}]

//状态通知 故障

[2,"103","StatusNotification",{"connectorId":1,"errorCode":"ReaderFailure","status":"Faulted"}]

[2021-11-09 16:15:30:087] receive data: ?[3,"102",{}]

//状态通知 充电中

[2,"113","StatusNotification",{"connectorId":1,"errorCode":"NoError","status":"Charging"}]

receive data: ?[3,"113",{}]

//状态通知 充电完成

[2,"135","StatusNotification",{"connectorId":1,"errorCode":"NoError","status":"Finishing"}]

receive data: ?[3,"135",{}]

//状态通知 预约功能

[2,"253","StatusNotification",{"connectorId":1,"errorCode":"NoError","status":"Reserved"}]

[2021-11-10 09:40:51:322] receive data: ?[3,"253",{}]

//心跳

[2,"104","Heartbeat",{}] (47 bytes)

[3,"104",{"currentTime":"2021-11-09T15:55:20.215"}]

//获取配置

[2,"2021119161654531","GetConfiguration",{"key":["WIFI\_SSID\_Key","WIFI\_PassWordKey"]}]

ResetData GetConfiguration = [3,"2021119161654531",{"configurationKey":[{"key":"GetConfigurationMaxKeys","value":"25","readonly":true},{"key":"Pilot\_Number","value":"2103252121090001","readonly":true},{"key":"CurrentPrice","value":"0.60","readonly":false},

{"key":"Language","value":"English","readonly":false},{"key":"MaxCurrent","value":"48.00","readonly":false},{"key":"ChargerMode","value":"1","readonly":false},{"key":"Local\_Ip","value":"192.168.1.8","readonly":false},

{"key":"HeartbeatInterval","value":"60","readonly":false},{"key":"MeterValueInterval","value":"60","readonly":false},{"key":"Over\_Temper","value":"80","readonly":false},{"key":"RemoteStartAuthorize","value":"0","readonly":true},

{"key":"LocalAuthorize","value":"0","readonly":true},{"key":"LocalPreAuthorize","value":"0","readonly":true},{"key":"ConnectNumber","value":"1","readonly":true},{"key":"Wifi\_Name","value":"WIFI\_Default","readonly":false},

{"key":"Wifi\_Password","value":"12345678","readonly":false},{"key":"Rate\_Price1","value":"0.50","readonly":false},{"key":"Rate\_Time1","value":"00:00-07:00","readonly":false},{"key":"Rate\_Price2","value":"1.00","readonly":false},

{"key":"Rate\_Time2","value":"07:00-12:00","readonly":false},{"key":"Rate\_Price3","value":"0.60","readonly":false},{"key":"Rate\_Time3","value":"12:00-18:00","readonly":false},{"key":"Rate\_Price4","value":"1.50","readonly":false},

{"key":"Rate\_Time4","value":"18:00-00:00","readonly":false},{"key":"J\_RateCurrency","value":"0-00:00","readonly":false}]}]

//更改配置

[2,"20211191617369433","ChangeConfiguration",{"key":"config","value":"[{\"key\":\"Rate\_Price1\",\"value\":\"0.50\"},{\"key\":\"Rate\_Time1\",\"value\":\"00:00-07:00\"},{\"key\":\"Rate\_Price2\",\"value\":\"1.00\"},{\"key\":\"Rate\_Time2\",\"value\":\"07:00-12:00\"},{\"key\":\"Rate\_Price3\",\"value\":\"0.60\"},{\"key\":\"Rate\_Time3\",\"value\":\"12:00-18:00\"},{\"key\":\"Rate\_Price4\",\"value\":\"1.50\"},{\"key\":\"Rate\_Time4\",\"value\":\"18:00-00:00\"}]"}]

[3,"20211191617369433",{"status":"Accepted"}]

//远程启动

[2,"2021119162036620","RemoteStartTransaction",{"connectorId":1,"idTag":"13714319313","chargingProfile":{"chargingProfileId":1,

"transactionId":701,"stackLevel":1,"chargingProfilePurpose":"TxProfile","chargingProfileKind":"Recurring","chargingSchedule":

{"duration":0,"chargingRateUnit":"A","chargingSchedulePeriod":[{"startPeriod":1,"limit":32.0,"numberPhases":3}]}}}]

[3,"2021119162036620",{"status":"Accepted"}]

//授权

[2,"111","Authorize",{"idTag":"13714319313"}]

[3,"111",{"idTagInfo":{"expiryDate":"2021-11-09T16:20:36.224","status":"Accepted"}}]

//开始传输

[2,"112","StartTransaction",{"connectorId":1,"idTag":"13714319313","meterStart":0,"reservationId":0,"timestamp":"2021-11-09T16:20:35"}]

[3,"112",{"idTagInfo":{"expiryDate":"2021-11-09T16:20:36.297","parentIdTag":"13714319313","status":"Accepted"},"transactionId":701}]

//获取电表值

[2,"114","MeterValues",{"connectorId":1,"transactionId":701,"meterValue":[{"timestamp":"2021-11-09T16:20:35","sampledValue":

[{"value":"0.00","context":"Sample.Periodic","format":"Raw","measurand":"Energy.Active.Import.Register","unit":"kWh"},

{"value":"0.00","context":"Sample.Periodic","format":"Raw","measurand":"Current.Export","phase":"L1","unit":"A"},

{"value":"0.00","context":"Sample.Periodic","format":"Raw","measurand":"Current.Export","phase":"L2","unit":"A"},

{"value":"0.00","context":"Sample.Periodic","format":"Raw","measurand":"Current.Export","phase":"L3","unit":"A"},

{"value":"230.8","context":"Sample.Periodic","format":"Raw","measurand":"Voltage","phase":"L1","unit":"V"},

{"value":"230.1","context":"Sample.Periodic","format":"Raw","measurand":"Voltage","phase":"L2","unit":"V"},

{"value":"230.6","context":"Sample.Periodic","format":"Raw","measurand":"Voltage","phase":"L3","unit":"V"},

{"value":"33","context":"Sample.Periodic","format":"Raw","measurand":"Temperature","unit":"Celcius"}]}]}]

receive data: ?[3,"114",{}]

//远程停止

[2,"20211191627106370","RemoteStopTransaction",{"transactionId":701}]

RemoteStopTransactionData Response = [3,"20211191627106370",{"status":"Accepted"}]

//停止传输

[2,"134","StopTransaction",{"meterStop":0,"timestamp":"2021-11-09T16:27:09","transactionId":701}]

receive data: [3,"134",{}]

//数据传输

[2,"136","DataTransfer",{"vendorId":"TONLY","messageId":"currentrecord","data":{"id":1,"connectorId":1,"chargemode":1,"startengry":0.00,

"endengry":0.00,"starttime":"2021-11-09 16:26:38","endtime":"2021-11-09 16:27:08","costenergy":0.00,"costmoney":0.00,"transactionId":701,

"stopReason":"Stop 3"}}]

receive data: [3,"136",{"status":"Accepted"}]

//预约状态

[2,"20211110940511291","ReserveNow",{"connectorId":1,"expiryDate":"2021-11-10T09:55:50.877","idTag":"13714319313","reservationId":1}]

[3,"20211110940511291",{"status":"Accepted"}]

//取消预约

[2,"20211110118248789","CancelReservation",{"reservationId":5}]

CancelReservation Response = [3,"20211110118248789",{"status":"Accepted"}]

//触发功能 获取诊断状态

[2,"20211110159454930","TriggerMessage",{"connectorId":1,"requestedMessage":"DiagnosticsStatusNotification"}]

[3,"20211110159454930",{"status":"Accepted"}]

//触发功能 获取启动通知

[2,"202111101634533576","TriggerMessage",{"connectorId":1,"requestedMessage":"BootNotification"}]

[3,"202111101634533576",{"status":"Accepted"}]

//触发功能 获取升级状态

[2,"20211110163601519","TriggerMessage",{"connectorId":1,"requestedMessage":"FirmwareStatusNotification"}]

[3,"100",{"currentTime":"2021-11-10T16:33:49.888","interval":60,"status":"Accepted"}]

//触发功能 获取电表值

[2,"202111101641279662","TriggerMessage",{"connectorId":1,"requestedMessage":"MeterValues"}]

//设备处于空闲所以拒绝

TriggerMessageData Response = [3,"202111101641279662",{"status":"Rejected"}]

//触发功能 获取诊断

[2,"202111101637486717","GetDiagnostics",{"location":"ftp://test:root\_123@120.79.1.235:21/diagnosticFiles"}]

[3,"202111101637486717",{"fileName":"2103252121090001-2021-11-10.txt"}]

[2,"106","DiagnosticsStatusNotification",{"status":"Uploading"}]

//FTP 上传文件 Update File data successful!提示上传成功后发送上传完成

[2,"107","DiagnosticsStatusNotification",{"status":"Uploaded"}]

//触发功能 获取状态通知

[2,"202111101642296937","TriggerMessage",{"connectorId":1,"requestedMessage":"StatusNotification"}]

[3,"202111101634533576",{"status":"Accepted"}]

//更改有效性 可用

[2,"20211110168141165","ChangeAvailability",{"connectorId":1,"type":"Operative"}]

ChangeAvailabilityData Response = [3,"20211110168141165",{"status":"Accepted"}]

//更改有效性 禁用

[2,"202111101622478979","ChangeAvailability",{"connectorId":1,"type":"Inoperative"}]

ChangeAvailabilityData Response = [3,"202111101622478979",{"status":"Accepted"}]

//升级

[2,"20211191553208096","UpdateFirmware",{"location":"120.79.1.235:21:test:root\_123:/test/AC48k01\_update.bin","retrieveDate":"2021-11-09T15:53:20.247"}]7"}]

[3,"20211191553208096",{}]

[2,"102","FirmwareStatusNotification",{"status":"Downloaded"}]

[2,"103","FirmwareStatusNotification",{"status":"Downloaded"}]

[2,"102","FirmwareStatusNotification",{"status":"Installed"}]

[2,"103","FirmwareStatusNotification",{"status":"Installed"}]