h1. BroadLink Android/IOS SDK 开放接口用户手册(20140703 版本)　h1. BroadLink Android/IOS SDK Open API User Manual (Publish date of this reference: July 2014)

h2. *简介*　h2. *Overview*

BroadLink 开放 API 可以让开发者直接将 BroadLink 的 easyconfig 技术，BroadLink 现有产品的设备发现和控制集成到自己的应用中。　BroadLink's open API allows developers to directly access the easyconfig technique of Broadlink and provides integration support for their applications of the Broadlink devices discovery and control.

h2. *SDK 使用流程*　h2. *How to use the SDK*

# 库初始化　# Library Initialization

在调用各个接口之前，首先使用 network_init 命令对库进行初始化。　Before calling a API, use network_init to initialize the library at first.

# 配置设备上网　# Devices Network Configuration

调用 easyconfig 命令，开始配置设备联网参数。　Use easyconfig to configure the device network parameters.

# 查找设备　# Device Discovery

调用 probe_list 命令后，可以搜索到所有局域网内的设备。　After using probe_list, all broadlink devices can be detected within the local area network (LAN).

# 对设备进行初始化　# Device Initialization

在 probe_list 命令返回后，立刻调用 device_add 对设备进行初始化，device_add 内的参数必须填写为 probe_list 返回的内容。　As soon as probe_list returns that the command finished successfully, use device_add to initialize the device. Ensure the parameters of device_add are the same as the contents returned by probe_list.

# 控制设备　#　Device Control

调用不同的设备控制命令进行控制。　Use different device control commands to control devices.

# 设备信息保存　# Device Information Storage

应用程序发现设备后，需要将设备信息（probe_list 返回的内容）保存到自己的数据库或者文件内。APP 每次重新启动调用 network_init 命令后，　When devices are detected by applications, APP need to save device information (returned by probe_list) to their own database or file.After APP reboots and uses the network_init each time,
必须重新从数据库或者文件内，将保存的设备信息通过 device_add 命令传递给底层库。

it is required to send the saved device information from the database or file to underlying libraries using device_add.

h2. *接口说明*　h2. *API Specification*

下表列出了所有请求，所有请求均为同步模式。　All requests listed in the

following table are in synchronous mode.

|_.Request ID|_.Function|_.Comment|_.Command| |_.Request ID|_.Function|_.Comment|_.Command|
|1|网络线程的初始化|网络操作的前提，必须先执行该方法|network_init| |1|Network threads initialization|Using this method is the premise of the network operation|network_init|
|2|获取 SDK 版本信息|获取当前 SDK 的版本信息与更新内容|SDK_version| |2|Obtain SDK version information|Obtain current SDK version information and updated contents|SDK_version|
|11|查找新设备列表|通过该接口，可以查找到新设备列表，或者有内容更新的设备，用户需要自己判断该设备是否已经保存|probe_list| ″|11|Search for new ″″device list″″|Through this interface, the list of new devices or the devices with updated content can be found. It requires users to determine whether devices are saved|probe_list|″
|12|添加设备到网络线程|通过该接口，可以将设备加入网络线程中，后续对该设备进行的操作都以该步为前提|device_add| |12|Add devices to network threads|Through this interface, devices can be added to the network threads, and this step is the premise of further operations of these devices|device_add|
|13|修改设备信息|通过该接口，可以更新设备的名称与锁定状态|device_update| |13|Modify device information|Through this interface, the name and locking status of devices can be updated|device_update|
|14|将设备从网络线程中删除|通过该接口，可以将设备从网络线程中删除|device_delete| |14|Delete devices from network threads|Through this interface, devices can be deleted from network threads|device_delete|
|15|获取设备局域网 IP 地址|通过该接口，可以取得设备在本地网络中的 IP 地址|device_lan_ip| |15|Obtain LAN IP address of the devices|Through this interface, IP address of the device in local network can be obtained|device_lan_ip|
|16|获取设备网络状态|通过该接口，可以取得设备的当前网络状态|device_state| |16|Obtain network status of devices|Through this interface, current network status of the device can be obtained|device_state|
|41|SP1 的登录|通过该接口，对 SP1 进行认证，只有认证过后的设备才能控制|sp1_auth| |41|SP1 login|Authenticate SP1 through this interface. Only the authenticated devices can be controlled|sp1_auth|
|42|SP1 的控制|通过该接口，可以对 SP1 进行开启/关闭操作|sp1_control| |42|SP1 control|Through this interface, SP1 can be switched on or off|sp1_control|
|43|SP1 的刷新|通过该接口，可以查看 SP1 当前的最新状态|sp1_refresh| |43|SP1 refresh|Through this interface, the current status of SP1 can be looked up|sp1_refresh|
|44|SP1 的任务|通过该接口，可以设置 SP1 的周期任务|sp1_task| |44|SP1 tasks|Through this interface, the periodic tasks of SP1 can be set up|sp1_task|
|71|SP2 的刷新|通过该接口，可以直接获取 SP2 的当前状态，无需登录|sp2_refresh| |71|SP2 refresh|Through this interface, the current state of the SP2 can be obtained directly, without the need of login|sp2_refresh|

|72|SP2 的控制|通过该接口，可以对 SP2 进行开启/关闭操作|sp2_control| |72|SP2 control|Through this interface, SP2 can be switched on or off|sp2_control|
|73|SP2 的任务|通过该接口，可以设置 SP2 的周期任务|sp2_task| |73|SP2 tasks|Through this interface, the periodic tasks of SP2 can be set up|sp2_task|
|74|SP2 的实时功率|通过该接口，可以获取 SP2 的当前实时功率|sp2_current_power| |74|SP2 Real-Time power|Through this interface, current Real-Time power of SP2 can be obtained|sp2_current_power|
|75|SP2 的最近 24 小时功率|通过该接口，可以获取 SP2 的最近 24 小时功率曲线|sp2_24_power| |75|SP2 power of last 24 hours|Through this interface, power curve of SP2 power of last 24 hours can be obtained|sp2_24_power|
|76|SP2 的一周能耗统计|通过该接口，可以获取 SP2 的指定周的能耗统计|sp2_week_power| |76|SP2 measured energy consumption of a week|Through this interface, SP2 measured energy consumption of the specified week can be obtained|sp2_week_power|
|77|SP2 的一月能耗统计|通过该接口，可以获取 SP2 的指定月份的能耗统计|sp2_month_power| |77|SP2 measured energy consumption of a month|Through this interface, SP2 measured energy consumption of the specified month can be obtained|sp2_month_power|
|78|SP2 的一年能耗统计|通过该接口，可以获取 SP2 的指定年份的能耗统计|sp2_year_power| |78|SP2 measured energy consumption of a year|Through this interface, SP2 measured energy consumption of the specified year can be obtained|sp2_year_power|
|79|SP2 获取待机功率|通过该接口，可以获取 SP2 的待机功率|sp2_get_standby_power| |79|SP2 standby power obtainment|Through this interface, SP2 standby power can be obtained|sp2_get_standby_power|
|80|SP2 设置待机功率|通过该接口，可以设置 SP2 的待机功率|sp2_set_standby_power| |80|SP2 standby power setting|Through this interface, SP2 standby power can be set|sp2_set_standby_power|
|101|RM1 的登录|通过该接口，对 RM1 进行认证，只有认证过后的设备才能控制|rm1_auth| |101|RM1 login|Through this interface, Authenticate RM1 through this interface. Only the authenticated devices can be controlled|rm1_auth|
|102|RM1 学习模式|通过该接口，让 RM1 进入学习模式|rm1_study| |102|RM1 Learning Mode|Through this interface, RM1 can be set to Learning Mode|rm1_study|
|103|RM1 查询 code|通过该接口，获取 RM1 学习到的控制码|rm1_code| |103|RM1 control code query|Through this interface, the control codes learned by RM1 can be obtained|rm1_code|
|104|RM1 发送 code|通过该接口，让 RM1 发送指定的控制码|rm1_send| |104|RM1 control code sent|Through this interface, the specified control codes can be sent by RM1|rm1_send|
|131|RM2 的刷新|通过该接口，可以查看 RM2 所处环境的温度值|rm2_refresh| |131|RM2 refresh|Through this interface, ambient temperature value can be obtained|rm2_refresh|
|132|RM2 学习模式|通过该接口，让 RM2 进入学习模式|rm2_study| |132|RM2 Learning Mode||Through this interface, RM2 can be set to Learning Mode|rm2_study|
|133|RM2 查询 code|通过该接口，获取 RM2 学习到的控制码|rm2_code| |133|RM2 control

code query|Through this interface,  the control codes learned by RM2 can be obtained|rm2_code|
|134|RM2 发送 code|通过该接口，让 RM2 发送指定的控制码|rm2_send|  |134|RM2 control code sent|Through this interface, the specified control codes can be sent by RM2|rm2_send|
|161|A1 的刷新|通过该接口，可以查看 A1 所处环境的温湿度、光照、空气质量、噪声信息|a1_refresh|  |161|A1 refresh|Through this interface, the temperature, humidity, illumination, air quality and noise can be detected|a1_refresh|
|162|A1 的当前任务列表|通过该接口，可以获取到 A1 当前所有的任务列表|a1_task_list|  |162|Check Current Task List of A1|Through this interface, Current Task List of A1 can be obtained|a1_task_list|
|163|A1 添加任务|通过该接口，可以添加新的任务到 A1 中|a1_add_task|  |163|Add new tasks into A1|Through this interface, new tasks can be added into A1|a1_add_task|
|164|A1 删除任务|通过该接口，可以从 A1 中删除指定的任务|a1_del_task|  |164|Delete tasks from A1|Through this interface, the specified tasks can be deleted from A1|a1_del_task|
|10000|easyconfig|通过该接口，可以将使用本公司模块的未联网设备配置到局域网中|easyconfig|  |10000|Easyconfig|When the devices using broadlink modules are not connected to a network, they can be deployed to a local area network through this interface|easyconfig|
|10001|取消 easyconfig|通过该接口，可以停止正在进行中的easyconfig|cancel_easyconfig|  |10001|Easyconfig cancellation|Through this interface, easyconfig processing can be canceled|cancel_easyconfig|

*针对所有命令的返回数据中的 code 字段，0 表示成功，非 0 表示失败。所有数据都以 json 格式传输*  *For the the code field of the return data, 0 means success, while non-zero means failure. All data are transmitted in json format*

h3. 1.0 network_init 请求参数说明  h3.  1.0  network_init  Request  Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
  "api_id"     : 1,       : 1,
  "command"    : "network_init",    ,
  "license" : "00000000000000000000"  00000000000000000000
}  }
api_id: 命令 ID。    api_id: Command ID。
command: 命令。(可选)  command: Command.(optional)
license: 用户申请 api 时获得的 license.  license: The license obtained by users when they apply for the API.
</pre>  </pre>
<pre>Retval:    <pre>Retval:
{  {

"code":0, :0,
"msg":"network init success" network init success
} }
</pre> </pre>

h3. 1.1 SDK_version 请求参数说明 h3. 1.1 SDK_version Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id" : 2, : 2,
"command" : "SDK_version", ,
} }
api_id: 命令 ID。 api_id: Command ID。
command: 命令。(可选) command: Command.(optional)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"version":"V1.1.0-201406091100", ,
"update":"V1.1.0-201406091100 update contents: ....." V1.1.0-201406091100 update contents: .....
} }

code:返回码 code: Return code
version:当前 SDK 的版本 version: Current SDK version
update:当前 SDK 的更新内容 update: Updated contents of current SDK
</pre> </pre>

h3. 1.2 probe_list 请求参数说明 h3. 1.2 probe_list Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id" : 11, : 11,
"command" : "probe_list" probe_list
} }

code: 返回码 code: Return code
msg : 返回消息 msg : Return message
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,

```
    "msg":"Execute success!",   ,
    "list":[    :[
               {                    {
                    "mac":"00:11:22:33:44:55",  ,
                    "type":"SP2",   ,
                    "name":"智能插座",  ,
                    "lock":0,   :0,
                    "password":987961777,   :987961777,
                    "id":0, :0,
                    "subdevice":0,  :0,
                    "key":"097628343fe99e23765c1513accf8b02"
    097628343fe99e23765c1513accf8b02
               },                   },
               {                    {
                    "mac":"00:11:22:33:44:55",  ,
                    "type":"SP2",   ,
                    "name":"Example",   ,
                    "lock":0,   :0,
                    "password":912023642,   :912023642,
                    "id":0, :0,
                    "subdevice":0,  :0,
                    "key":"097628343fe99e23765c1513accf8b02"
    097628343fe99e23765c1513accf8b02
               }                    }
          ]              ]
}   }
```


code: 返回码　code: Return code
msg : 返回消息　msg :Return message
list: 查询到的设备属性信息数组　list: Device property information array from the query
mac: 设备的 mac 地址 mac: MAC address of the device
type: 设备类型，目前支持的设备类型，SP1/SP2/RM1/RM2（注:必须大写）. type: Device type, the current supported device types, SP1/SP2/RM1/RM2 (Note: should be capitalized).
name: 设备名称，UTF8 编码。 Name: Device name, UTF8 code.
lock: 设备当前锁定状态　lock: Current locking state of the device
password: SP1/RM1 认证所需密码，该值由设备生成，不能修改。　password: Password required for SP1/RM1 authentication, which is generated by device and cannot be modified.
id: SP2/RM2 通信所需，该值由设备生成，不能修改。 id: ID used in SP2/RM2 communication, which is generated by device and cannot be modified.
subdevice: 保留字段。置 0　subdevice: Reserved field.Set to 0
key: SP2/RM2 通信密钥，由设备产生，不能修改。　key: Key used in SP2/RM2

communication, which is generated by device and cannot be modified.
</pre> </pre>

h3. 1.3 device_add 请求参数说明 h3. 1.3 device_add Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":12, :12,
"command":"device_add", ,
"mac":"00:11:22:33:44:55", ,
"type":"RM2", ,
"name":"智能插座", ,
"lock":0, :0,
"password":1028000492, :1028000492,
"id":0, :0,
"subdevice":0, :0,
"key":"00000000000000000000000000000000" 00000000000000000000000000000000
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令（可选）。 command: Command (optional).

以下字段从 probe_list 中获取到，不能修改。 The following fields obtained from probe_list cannot be modified.
mac: 设备的 mac 地址. mac: MAC address of the device.
type: 设备类型，目前支持的设备类型，SP1/SP2/RM1/RM2。(注:必须大写) type: Device type, the current supported device types, SP1/SP2/RM1/RM2. (Note: should be capitalized)
name: 设备名称，UTF8 编码。 name: Device name, UTF8 code.
lock: 设备当前锁定状态 lock: Current locking state of the device
password: SP1/RM1 认证所需密码，该值由设备生成，不能修改。 password: Password required for SP1/RM1 authentication, which is generated by device and cannot be modified.
id: SP2/RM2 通信所需，该值由设备生成，不能修改。 id: ID used in SP2/RM2 communication, which is generated by device and cannot be modified.
subdevice: 保留字段。置 0 subdevice: Reserved field.Set to 0
key: SP2/RM2 通信密钥，由设备产生，不能修改。 key: Key used in SP2/RM2 communication, which is generated by device and cannot be modified.
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"Add device success" Add device success

}  }

code: 返回码  code: return code
msg : 返回消息  msg : return message
</pre>  </pre>

h3. 1.4 device_update 请求参数说明  h3.  1.4  device_update  Request  Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
    "api_id":13,  :13,
    "command":"device_update",  ,
    "mac":"00:11:22:33:44:55",  ,
    "name":"智能插座",  ,
    "lock":0, :0,
}  }

api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)
mac: 要修改的设备的 mac 地址。 （必填）  mac:  Device  MAC  address  needs  to  be modified. (mandatory)
name: 要修改的名称, UTF8 编码。（可选）  name: Device name needs to be modified, UTF8 code.(Optional)
lock: 是否锁定。 （可选）  lock: Whether or not to lock. (Optional)
</pre>  </pre>
<pre>Retval:  <pre>Retval:
{  {
    "code":0,  :0,
    "msg":"Update device info success" Update device info success
}  }

code: 返回码  code: return code
msg : 返回消息  msg : return message
</pre>  </pre>

h3. 1.5 device_delete 请求参数说明  h3.  1.5  device_delete  Request  Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
    "api_id":14,  :14,
    "command":"device_delete",  ,
    "mac":"00:11:22:33:44:55" 00:11:22:33:44:55

}  }

api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)
mac: 要删除的设备的 MAC 地址。（必填）  mac: Device MAC address needs to be deleted.(mandatory)
</pre> </pre>
<pre>Retval:  <pre>Retval:
{  {
 "code":0, :0,
 "msg":"Delete device success" Delete device success
}  }

code: 返回码  code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.6 device_lan_ip 请求参数说明 h3. 1.6 device_lan_ip Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
  "api_id":15, :15,
  "command":"device_lan_ip", ,
  "mac":"00:11:22:33:44:55" 00:11:22:33:44:55
}  }

api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)
mac: 要删除的设备的 MAC 地址。（必填）  mac: Device MAC address needs to be deleted.(mandatory)
</pre> </pre>
<pre>Retval:  <pre>Retval:
{  {
 "code":0, :0,
 "lan_ip":"192.168.1.113", ,
 "msg":"mac:00:11:22:33:44:55 get lan ip success" mac:00:11:22:33:44:55 get lan ip success
}  }

code: 返回码  code: return code
lan_ip: 设备当前的局域网 IP 地址.  lan_ip: Current device LAN IP address.
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.7 device_state 请求参数说明  h3. 1.7 device_state Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
  "api_id":16,  :16,
  "command":"device_state",  ,
  "mac":"00:11:22:33:44:55" 00:11:22:33:44:55
}  }

api_id: 命令 ID。  api_id: command ID。
command: 命令。（可选）  command: command. (Optional)
mac: 要删除的设备的 MAC 地址。(必填)  mac: Device MAC address needs to be deleted.(mandatory)
</pre> </pre>
<pre>Retval:  <pre>Retval:
{  {
  "code":0,  :0,
  "status":"LOCAL",  ,
  "msg":"mac:00:11:22:33:44:55 get state success"  mac:00:11:22:33:44:55 get state success
}  }

code: 返回码  code: return code
status: 设备当前的网络状态. (NOT_INIT:未初始化 LOCAL:局域网 REMOTE:远程 OFFLINE:离线)  status: current network status of the device (NOT_INIT: Not initialized Local: LAN REMOTE: Remote OFFLINE: offline)
msg : 返回消息  msg : return message
</pre> </pre>

h3. 1.8 sp1_auth 请求参数说明  h3. 1.8 sp1_auth Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
  "api_id":41,  :41,
  "command":"sp1_auth",  ,
  "mac":"00:11:22:33:44:55",  ,
  "password":1028000492 :1028000492
}  }

api_id: 命令 ID。  api_id: command ID。
command: 命令。（可选）  command: command. (Optional)
mac: 要登录的设备的 MAC 地址。(必填)  mac: MAC address of the device that needs

to login.(mandatory)
password：设备的登录密码。(必填)　password：login password of the device.(mandatory)
</pre> </pre>
<pre>Retval:　<pre>Retval:
{　{
　"code":0, :0,
　"msg":"sp1 auth success", ,
　"status":1,　:1,
　"name":"Smart Plug 1", ,
　"lock":0, :0,
　"periodic_task":[ :[
　　{　　{
　　"enable":1, :1,
　　"on_time":"08:00", ,
　　"off_time":"15:00", ,
　　"repeat":0 :0
　　},　　},
　　{　　{
　　"enable":1, :1,
　　"on_time":"--:--", ,
　　"off_time":"00:00", ,
　　"repeat":127　:127
　　}　　}
　]　　]
}　}

code: 返回码　code: return code
msg : 返回消息 msg : return message
status: 插座当前状态（1：开 0：关） status: current status of the SP1 device.(1: turn on, 0: turn off)
name: 插座的名称.　name: Name of the SP1 device.
lock: 插座是否上锁. lock: Whether SP1 device is locked or not.
periodic_task: 插座的周期任务. periodic_task: Periodic tasks of the SP1 device.
　enable: 周期任务使能。　enable: Enable periodic tasks of the SP device.
　on_time: 周期任务开启时间,若没有设置,则为"--:--". .
　off_time: 周期任务关闭时间,若没有设置,则为"--:--". .
　repeat: 周期任务的重复周期，以周为一个周期，0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。　repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday).
</pre> </pre>

h3. 1.9 sp1_control 请求参数说明 h3. 1.9 sp1_control Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":42, :42,
"command":"sp1_control", ,
"mac":"00:11:22:33:44:55", ,
"status": 0 : 0
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填） mac: MAC address of the device needs to be manipulated.(mandatory)
status：要设置的状态(0:关闭 1：开启)。(必填） status: the status to be set (0: off; 1: on).(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"Set sp1 status success" Set sp1 status success
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.10 sp1_refresh 请求参数说明 h3. 1.10 sp1_refresh Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":43, :43,
"command":"sp1_refresh", ,
"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:

```
{   {
     "code":0,  :0,
     "msg":"Refresh sp1 status success" Refresh  1 status success
     "status": 0    : 0
     "name":"Smart Plug 1", ,
     "lock":0,  :0,
     "periodic_task":[  :[
                                {                                   {
                                      "enable":1, :1,
                                      "on_time":"08:00",  ,
                                      "off_time":"15:00", ,
                                      "repeat":0  :0
                                },                                      },
                                {                                   {
                                      "enable":1, :1,
                                      "on_time":"--:--",  ,
                                      "off_time":"00:00", ,
                                      "repeat":127    :127
                                }                                   }
                           ]                              ]
}   }
```

code: 返回码  code: return code
msg : 返回消息  msg : return message
status: 当前开关状态（0：关闭 1：开启） status: Current switch status (0: turn off 1: turn on)
name: 插座的名称.  name: Name of SP1
lock: 插座是否上锁. lock: Whether SP1 device is locked or not.
periodic_task: 插座的周期任务.  periodic_task: Periodic task of SP1.
    enable: 周期任务使能。  enable: Enable periodic  tasks of the SP device.
    on_time: 周期任务开启时间,若没有设置,则为"--:--".  .
    off_time: 周期任务关闭时间,若没有设置,则为"--:--".  .
    repeat: 周期任务的重复周期，以周为一个周期，0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。  repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday).
</pre>  </pre>

h3. sp1_task 请求参数说明  h3. sp1_task Request Parameter Specification

<pre>Arguments: <pre>Arguments:

```
{   {
        "api_id":44,    :44,
```

```
        "command":"sp1_task",    ,
        "mac":"00:11:22:33:44:55",  ,
        "name":"Smart Plug 1",  ,
        "lock":0,    :0,
        "periodic_task":[   :[
                                {                                       {
                                      "enable":1,  :1,
                                      "on_time":"08:00",    ,
                                      "off_time":"15:00",  ,
                                      "repeat":0   :0
                                },                                      },
                                {                                       {
                                      "enable":1,  :1,
                                      "on_time":"--:--",    ,
                                      "off_time":"00:00",  ,
                                      "repeat":127 :127
                                }                                       }
                             ]                                  ]
}   }
```

api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填）  mac: MAC address of the device needs to be manipulated.(mandatory)
name: 插座的名称.  name: Name of SP1
lock: 插座是否上锁. lock: Whether SP1 device is locked or not.
periodic_task: 插座的周期任务.  periodic_task: Periodic task of SP1.
enable: 周期任务使能。  enable: Enable periodic tasks of the SP1 device.
on_time: 周期任务开启时间,若没有设置,则为"--:--".  .
off_time: 周期任务关闭时间,若没有设置,则为"--:--".  .
repeat: 周期任务的重复周期，以周为一个周期，0 表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。  repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday).

注意:设置任务需要把原来的任务也同时设置，添加与删除为同一个接口。同时还需要设置设备名称和锁定状态  Note: While setting up a task, the previous task need to be set up at the same time. Task of adding and deleting should use the same interface.The device name and locking status also need to be set.
</pre>  </pre>
<pre>Retval:  <pre>Retval:
{   {
    "code":0,  :0,
```
```

```
     "msg":"Set sp1 task success"   Set sp1 task success
}   }
```

code: 返回码    code: return code
msg : 返回消息  msg : return message
</pre>  </pre>

h3. sp2_refresh 请求参数说明 h3. sp2_refresh Request Parameter Specification

<pre>Arguments: <pre>Arguments:

```
{   {
      "api_id":71,  :71,
      "command":"sp2_refresh",  ,
      "mac":"00:11:22:33:44:55" 00:11:22:33:44:55
}   }
```

api_id: 命令 ID。    api_id: command ID。
command: 命令。 （可选）    command: command. (Optional)
mac: 要刷新的设备的 MAC 地址。（必填）    mac: MAC address of the device need to be refreshed.(mandatory)
</pre>  </pre>
<pre>Retval:    <pre>Retval:

```
{   {
     "code":0,  :0,
     "msg":"Refresh sp2 status success",    ,
     "status":0 :0
     "name":"Smart Plug 2", ,
     "lock":0,  :0,
     "periodic_task":[  :[
                                {                                   {
                                      "enable":1, :1,
                                      "on_time":"08:00",  ,
                                      "off_time":"15:00", ,
                                      "repeat":0  :0
                                },                                      },
                                {                                   {
                                      "enable":1, :1,
                                      "on_time":"--:--",  ,
                                      "off_time":"00:00", ,
                                      "repeat":127    :127
                                }                                   }
                           ]                                ]
    "timer_task":[  :[
                           {                                {
```

```
                                "on_enable":1,  :1,
                                "on_time":"2014-07-02 14:58:03" 2014-07-02
14:58:03
                                "off_enable":0, :0,
                                "off_time":"2014-07-02 15:58:03"    2014-07-
02 15:58:03
                            },                              },
                            {                               {
                                "on_enable":0,  :0,
                                "on_time":"2014-07-02 14:58:03" 2014-07-02
14:58:03
                                "off_enable":1, :1,
                                "off_time":"2014-07-03 16:58:03"    2014-07-
03 16:58:03
                            },                                  },
                            {                               {
                                "on_enable":1,  :1,
                                "on_time":"2014-07-02 14:58:03" 2014-07-02
14:58:03
                                "off_enable":1, :1,
                                "off_time":"2014-07-05 18:58:03"    2014-07-
05 18:58:03
                            }                               }
                        ]                           ]
}   }
```


code: 返回码  code: return code
msg : 返回消息  msg : return message
status: 当前设备的开关状态（0: 关闭 1：开启）  status: Current on-off status of the device (0:  turn off 1：turn on)
name: 插座的名称.  name: Name of SP1
lock: 插座是否上锁. lock: Whether SP1 device is locked or not.
periodic_task: 插座的周期任务.  periodic_task: Periodic task of SP1.
    enable: 周期任务使能。  enable: Enable periodic  tasks of the SP1 device.
    on_time: 周期任务开启时间,若没有设置,则为"--:--".  .
    off_time: 周期任务关闭时间,若没有设置,则为"--:--".  .
    repeat: 周期任务的重复周期，以周为一个周期，0表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。  repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday).
timer_task: 插座的定时任务. timer_task: Timer task of SP2.
    on_enable: 定时任务的开启使能  on_enable: Enable timer task of SP2.
    on_time: 定时任务的开启时间点  on_time: Turn on time point of timer task.

off_enable: 定时任务的关闭使能 off_enable: Timer task.enable closure.
off_time: 定时任务的关闭时间点 off_time: Turn off time point of timer task.
</pre> </pre>

h3. sp2_control 请求参数说明 h3. sp2_control Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":72, :72,
"command":"sp2_control", ,
"mac":"00:11:22:33:44:55", ,
"status":1 :1
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填） mac: MAC address of the device needs to be manipulated.(mandatory)
status: 要设置的开关状态（0:关闭 1：开启)。(必填） status: status: the status to be set (0: off; 1: on).(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"Set sp2 status success" Set sp2 status success
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. sp2_task 请求参数 h3. sp2_task Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":73, :73,
"command":"sp2_task", ,
"mac":"00:11:22:33:44:55", ,
"name":"Smart Plug 2", ,
"lock":0, :0,
"periodic_task":[ :[
{ {
"enable":1, :1,

```
                                    "on_time":"08:00",  ,
                                    "off_time":"15:00", ,
                                    "repeat":0  :0
                                },                                          },
                                {                                       {
                                    "enable":1, :1,
                                    "on_time":"--:--",  ,
                                    "off_time":"00:00", ,
                                    "repeat":127    :127
                                }                                       }
                            ]                               ]
    "timer_task":[ :[
                            {                               {
                                "on_enable":1,  :1,
                                "on_time":"2014-07-02 14:58:03" 2014-07-02
14:58:03
                                "off_enable":0, :0,
                                "off_time":"2014-07-02 15:58:03"    2014-07-
02 15:58:03
                            },                              },
                            {                               {
                                "on_enable":0,  :0,
                                "on_time":"2014-07-02 14:58:03" 2014-07-02
14:58:03
                                "off_enable":1, :1,
                                "off_time":"2014-07-03 16:58:03"    2014-07-
03 16:58:03
                            },                                  },
                            {                               {
                                "on_enable":1,  :1,
                                "on_time":"2014-07-02 14:58:03" 2014-07-02
14:58:03
                                "off_enable":1, :1,
                                "off_time":"2014-07-05 18:58:03"    2014-07-
05 18:58:03
                            }                               }
                        ]                       ]
}   }
```


api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填）  mac: MAC address of the device needs to be manipulated.(mandatory)
name: 插座的名称.  name: Name of SP2

lock: 插座是否上锁. lock: Whether SP2 device is locked or not.
periodic_task: 插座的周期任务. periodic_task: Periodic task of SP2.
enable: 周期任务使能。 enable: Enable periodic tasks of the SP1 device.
on_time: 周期任务开启时间,若没有设置,则为"--:--". .
off_time: 周期任务关闭时间,若没有设置,则为"--:--". .
repeat: 周期任务的重复周期，以周为一个周期，0 表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。 repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday).
timer_task: 插座的定时任务. timer_task: Timer task of SP2.
on_enable: 定时任务的开启使能 on_enable: Enable timer task of SP2.
on_time: 定时任务的开启时间点 on_time: Turn on time point of timer task.
off_enable: 定时任务的关闭使能 off_enable: Timer task.enable closure.
off_time: 定时任务的关闭时间点 off_time: Turn off time point of timer task.

注意:设置任务需要把原来的任务也同时设置，添加与删除为同一个接口。同时还需要设置设备名称和锁定状态 Note: While setting up a task, the previous task need to be set up at the same time. Task of adding and deleting should use the same interface.The device name and locking status also need to be set.
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"Set sp2 task success" Set sp2 task success
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. sp2_current_power 请求参数说明 h3. sp2_current_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":74, :74,
"command":"sp2_current_power", ,
"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)

mac: 要操作的设备的 MAC 地址。(必填)　　mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval:　　<pre>Retval:
{　{
　　"code":0,　:0,
　　"current_power":121.000000,　　:121.000000,
　　"msg":"Get sp2 current power success"　Get sp2 current power success
}　}

code: 返回码　　code: return code
current_power: 设备当前的实时功率。 current_power: Current real time power of device
msg : 返回消息　msg : return message
</pre> </pre>

h3. sp2_24_power 请求参数说明　　h3. sp2_24_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{　{
　　"api_id":75,　:75,
　　"command":"sp2_24_power",　　,
　　"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
}　}

api_id: 命令 ID。　　api_id: command ID。
command: 命令。 (可选)　　command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填)　　mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval:　　<pre>Retval:
{　{
　　"code":0,　:0,
　　"current_power":121.000000,　　:121.000000,
　　"24hour_power":[0.000000, 60.000000, 80.000000, ...],　:[0.000000, 60.000000, 80.000000, ...],
　　"msg":"Get sp2 last 24hour power success"　Get sp2 last 24hour power success
}　}

code: 返回码　　code: return code
current_power: 设备当前的实时功率。 current_power: Current real time power of device
24hour_power: 设备最近 24 小时的功率，每 5 分钟一个数据点，总共 288 个。
　　24hour_power: SP2 power of last 24 hours, power data collected each 5 minutes

and 288 data points in total.
msg : 返回消息 msg : return message
</pre> </pre>

h3. sp2_week_power 请求参数说明 h3. sp2_week_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":76, :76,
"command":"sp2_week_power", ,
"mac":"00:11:22:33:44:55", ,
"week_index":2 :2
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
week_index: 当前年的第几周（从 1 开始） week_index: A specific week of current year (Beginning from 1)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"week_energy":[ :[
{ {
"peak_energy":0.250000, :0.250000,
"low_energy":0.750000, :0.750000,
"on_time":50, :50,
"standby_energy":0.180000 :0.180000
}, },
{ {
"peak_energy":0.450000, :0.450000,
"low_energy":1.250000, :1.250000,
"on_time":300, :300,
"standby_energy":0.580000 :0.580000
} }
] ]
"msg":"Get sp2 week's energy info success." Get sp2 week's energy info success.
} }

code: 返回码 code: return code

week_energy: 设备指定周的能耗统计，每天一个数据点，最多 7 个点。 week_energy: Device energy consumption statistics of a specific week, consumption data collected each day and 7 data points in total.

peak_energy: 峰电能耗统计 peak_energy: Energy consumption statistic of peak electricity

low_energy: 谷电能耗统计 low_energy: Energy consumption statistic of valley electricity

on_time: 运行时间 on_time: Run time

standby_energy: 待机能耗统计 standby_energy: Standby energy consumption

msg : 返回消息 msg : return message

</pre> </pre>

h3. sp2_month_power 请求参数说明 h3. sp2_month_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:

```
{   {
      "api_id":77,  :77,
      "command":"sp2_month_power",  ,
      "mac":"00:11:22:33:44:55",    ,
      "year":2014,  :2014,
      "month_index":3   :3
}   }
```

api_id: 命令 ID。 api_id: command ID。

command: 命令。 （可选） command: command. (Optional)

mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)

year: 指定年份 year: A specific year

month_index: 指定月份 month_index: A specific month

</pre> </pre>

<pre>Retval: <pre>Retval:

```
{   {
     "code":0,  :0,
     "month_energy":[   :[
                                  {                                          {
                                       "peak_energy":0.250000,     :0.250000,
                                       "low_energy":0.750000, :0.750000,
                                       "on_time":50,  :50,
                                       "standby_energy":0.180000  :0.180000
                                  },                                         },
                                  {                                          {
                                       "peak_energy":0.450000,     :0.450000,
```

"low_energy":1.250000, :1.250000,
"on_time":300, :300,
"standby_energy":0.580000 :0.580000
} }
] ]
"msg":"Get sp2 month's energy info success." Get sp2 month's energy info success.
} }

code: 返回码 code: return code
month_energy: 设备指定年月的能耗统计，每天一个数据点，最多 31 个点。month_energy: Device energy consumption statistics of the specific year and month, consumption data collected each day and 31 data points in total.
peak_energy: 峰电能耗统计 peak_energy: Energy consumption statistic of peak electricity
low_energy: 谷电能耗统计 low_energy: Energy consumption statistic of valley electricity
on_time: 运行时间 on_time: Run time
standby_energy: 待机能耗统计 standby_energy: Standby energy consumption
msg : 返回消息 msg : return message
</pre> </pre>

h3. sp2_year_power 请求参数说明 h3. sp2_year_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":78, :78,
"command":"sp2_year_power", ,
"mac":"00:11:22:33:44:55", ,
"year":2014 :2014
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填） mac: MAC address of the device needs to be manipulated.(mandatory)
year: 指定年份） year: Specific year)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"year_energy":[ :[

{ {
"peak_energy":100.250000, :100.250000,
"low_energy":150.750000, :150.750000,
"on_time":1550, :1550,
"standby_energy":30.180000 :30.180000
}, },
{ {
"peak_energy":120.450000, :120.450000,
"low_energy":170.250000, :170.250000,
"on_time":3000, :3000,
"standby_energy":80.580000 :80.580000
} }
] ]
"msg":"Get sp2 year's energy info success." Get sp2 year's energy info success.
} }

code: 返回码 code: return code
year_energy: 设备指定年份的能耗统计，每月一个数据点，最多 12 个点。 year_energy: Device energy consumption statistics of the specific year, consumption data collected each month and 12 data points at most.
peak_energy: 峰电能耗统计 peak_energy: Energy consumption statistic of peak electricity
low_energy: 谷电能耗统计 low_energy: Energy consumption statistic of valley electricity
on_time: 运行时间 on_time: Run time
standby_energy: 待机能耗统计 standby_energy: Standby energy consumption
msg : 返回消息 msg : return message
</pre> </pre>

h3. sp2_get_standby_power 请求参数说明 h3. sp2_get_standby_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":79, :79,
"command":"sp2_get_standby_power", ,
"mac":"00:11:22:33:44:55", ,
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to

be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"enable":1, :1,
"standby_power":10 :10
"msg":"Get sp2 standby power success." Get sp2 standby power success.
} }

code: 返回码 code: return code
enable: 待机功率是否生效 enable: Whether standby power takes effect or not
standby_power: 待机功率值 standby_power: Value of standby power
msg : 返回消息 msg : return message
</pre> </pre>

h3. sp2_set_standby_power 请求参数说明 h3. sp2_set_standby_power Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":80, :80,
"command":"sp2_set_standby_power", ,
"mac":"00:11:22:33:44:55", ,
"enable":1, :1,
"standby_power":5 :5
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
enable:设置待机功率是否生效 enable: Whether enable standby power or not
standby_power: 设置待机功率 standby_power: Standby power setting
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"Set sp2 standby power success." Set sp2 standby power success.
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.13 rm1_auth 请求参数说明  h3. 1.13 rm1_auth Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
      "api_id":101, :101,
      "command":"rm1_auth",     ,
      "mac":"00:11:22:33:44:55",    ,
      "password":1028000492 :1028000492
}  }

api_id: 命令 ID。    api_id: command ID。
command: 命令。 （可选）    command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填）    mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval:    <pre>Retval:
{  {
     "code":0,  :0,
     "temperature":23.100000,   :23.100000,
     "msg":"rm1 auth success"   rm1 auth success
}  }

code: 返回码    code: return code
temperature: 设备当前所在环境的温度。   temperature: Ambient temperature value setting.
msg : 返回消息  msg : return message
</pre> </pre>

h3. 1.14 rm1_study 请求参数说明 h3.  1.14  rm1_study  Request  Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
      "api_id":102, :102,
      "command":"rm1_study",    ,
      "mac":"00:11:22:33:44:55" 00:11:22:33:44:55
}  }

api_id: 命令 ID。    api_id: command ID。
command: 命令。 （可选）    command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填）    mac: MAC address of the device needs to be manipulated.(mandatory)

</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"rm1 enter study mode success" rm1 enter study mode success
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.15 rm1_code 请求参数说明 h3. 1.15 rm1_code Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":103, :103,
"command":"rm1_code", ,
"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":-100, :-100,
"msg":"rm1 check data failed" rm1 check data failed
} }

Retval: Retval:
{ {
"code":0, :0,

"data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e", ,
"msg":"rm1 check data success" rm1 check data success
} }

code: 返回码, -100 表示没查到学习到的数据，0 表示查到学习的数据 code: Return code, -100 means no learning data found, while 0 means learning data found.
data: 学习到的控制码。 data: Learned control codes.
msg : 返回消息 msg : return message

</pre> </pre>

h3. 1.16 rm1_send 请求参数说明 h3. 1.16 rm1_send Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":104, :104,
"command":"rm1_send", ,
"mac":"00:11:22:33:44:55", ,

"data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e"
2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"rm1 send data success" rm1 send data success
} }
code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.17 rm2_refresh 请求参数说明 h3. 1.17 rm2_refresh Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":131, :131,
"command":"rm2_refresh", ,
"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to

be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"temperature":23.100000, :23.100000,
"msg":"Refresh rm2 temperature success" Refresh rm2 temperature success
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.18 rm2_study 请求参数说明 h3. 1.18 rm2_study Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":132, :132,
"command":"rm2_study", ,
"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"rm2 enter study mode success" rm2 enter study mode success
} }

code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.19 rm2_code 请求参数说明 h3. 1.19 rm2_code Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":133, :133,

"command":"rm2_code", ,
"mac":"00:11:22:33:44:55" 00:11:22:33:44:55
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":-1, :-1,
"msg":"rm2 check data failed" rm2 check data failed
} }

Retval: Retval:
{ {
"code":0, :0,

"data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e", ,
"msg":"rm2 check data success" rm2 check data success
} }

code: 返回码, -1 表示没查到学习到的数据，0 表示查到 code: Return code, -1 means no learning data found, while 0 means learning data found.
data: 学习到的控制码。 data: Learned control codes.
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.20 rm2_send 请求参数说明 h3. 1.20 rm2_send Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":134, :134,
"command":"rm2_send", ,
"mac":"00:11:22:33:44:55", ,

"data":"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e"
2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"rm2 send data success" rm2 send data success
} }
code: 返回码 code: return code
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.21 a1_refresh 请求参数说明 h3. 1.21 a1_refresh Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":161, :161,
"command":"a1_refresh", ,
"mac":"00:11:22:33:44:55", ,
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。（必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"temperature":25.0, :25.0,
"humidity":56.3, :56.3,
"light":3, :3,
"air":2, :2,
"noisy":1, :1,
"name":"A1 Eair", ,
"lock":0, :0,
"msg":"Refresh a1 sensor's value success" Refresh a1 sensor's value success
} }
code: 返回码 code: return code
temperature: 温度值, 浮点型 temperature: temperature, floating-point type

humidity: 湿度值,浮点型 humidity: humidity value, floating point type
light:光照度,整型 (0:暗 1:昏暗 2:正常 3:亮) light: light: illuminance, integer (0:dark 1:dim 2:normal 3:bright)
air: 空气质量,整形 (0:优 1:良 2:正常 3:差) air: air: air quality, integer (0:excellent 1:fine 2:normal 3:bad)
noisy: 噪声,整形 (0:寂静 1:正常 2:吵闹) noisy: noisy: noise, integer (0:quiet 1:normal 2:noisy)
name: 设备名称,字符串 name: name: device name, string
lock: 设备锁定状态，整型 (0:未锁 1:锁定) lock: lock: device locking status, integer (0:unlocked 1:locked)
msg : 返回消息 msg : return message
</pre> </pre>

h3. 1.22 a1_task_list 请求参数说明 h3. 1.22 a1_task_list Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":162, :162,
"command":"a1_task_list", ,
"mac":"00:11:22:33:44:55", ,
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填） mac: MAC address of the device needs to be manipulated.(mandatory)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
“list”:[ “list”:[
{ {
"mac": "00:11:22:33:44:55", ,
"time_enable":0, :0,
"task_enable":1, :1,
"index":0, :0,
"start_time":"10:30", ,
"end_time":"18:30", ,
"repeat":7, :7,
"sensor_type":0, :0,
"sensor_trigger": 0, : 0,
"sensor_value":25.0, :25.0,
"task_name":"打开电视机" Turn on TV.

```
                },                          },
                {                           {
                     "mac": "55:44:33:22:11:00", ,
                    "time_enable":1, :1,
                    "task_enable":1, :1,
                    "index":2,   :2,
                    "start_time":"10:30",    ,
                    "end_time":"17:30",  ,
                    "repeat":5,  :5,
                    "sensor_type":2, :2,
                    "sensor_trigger": 1, : 1,
                    "sensor_value":3.0,  :3.0,
                    "task_name":"打开音响"   Open the audio player
                }                           }
            ],                  ],
     "msg":"Get a1 task list success"   Get a1 list task list success
}   }
```

code: 返回码  code: return code

list: 任务列表  list: Task list

mac: 执行任务的设备 mac 地址。字符串  mac: Mac address of the device to execute tasks.String

time_enable: 时间使能。 整型  time_enable: Time enable Integer

task_enable: 任务使能。 整形  task_enable: Task enable Integer

index: 任务编号,(删除任务时使用)。 整形  index: Task Number, (used when deleting task) Integer

start_time: 任务执行的开始时间。 字符串  start_time: Start time of task executing String

end_time: 任务执行的结束时间。 字符串  end_time: End time of task executing String

repeat: 任务的重复周期，以周为一个周期，0 表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。 整型  repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday). Integer

sensor_type: 任务执行条件的传感器类型, (0: 温度  1:湿度  2:光照  3:空气质量  4:噪声)。 整型  sensor_type: Sensor types of the task execution condition (0: temperature  1:humidity  2:illuminance  3:air quality  4:noise). Integer

sensor_trigger: 任务执行条件传感器值发生变化的趋势, (0:上升  1:下降)。 整型  sensor_trigger: The change in the trend of the sensor value of the task execution condition (0: rise 1: fall) Integer

sensor_value: 任务执行条件传感器的值. 浮点型  sensor_value: The sensor value of the task execution condition. floating point type

task_name: 执行任务的名称。 字符串  task_name: Name of the executed task. String

msg : 返回消息  msg : return message
</pre>  </pre>

h3. 1.23 a1_add_task 请求参数说明  h3.  1.23  a1_add_task  Request  Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
      "api_id":163, :163,
      "command":"a1_add_task",  ,
      "mac":"00:11:22:33:44:55",    ,
      "task_name": "关闭空调",  ,
      "time_enable": 1, : 1,
      "task_enable": 1,     : 1,
      "start_time": "2014-06-06 10:30:00",  ,
      "end_time": "2014-06-06 18:30:00",    ,
      "repeat": 7,  : 7,
      "sensor_type": 0, : 0,
      "sensor_trigger": 1,  : 1,
      "sensor_value" : 20.0,     : 20.0,
      "device_mac": "aa:bb:cc:dd:ee:ff",    ,
      "device_id": 25,  : 25,
      "device_key": "097628343fe99e23765c1513accf8b02", ,
      "device_type": "RM2", ,
      "task_data":
"2600e6006f390e2a0e290d100d100e0f0e290e0f0e0f0e2a0e290d100e2a0e0e0f0e0e2a0d2a0e0f0e2a0e",  ,
      "status": 0   : 0
}  }

api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填）  mac: MAC address of the device needs to be manipulated.(mandatory)
task_name: 添加的任务的名称。  task_name: Name of the added task
time_enable: 任务时间使能。 time_enable: Task time enable.
task_enable: 任务使能。 task_enable: Task enable
start_time: 任务开始时间。必须按照该格式传入时间参数，其中年月日只在 time_enable 为 0 时生效(年月日必须为当天的准确时间)。  start_time: Task start time.must be passed into time parameter in this format, especially day, month and year are only available when tiem_enalbe is set to be 0 (day, month and year must be the exact time of the day)
end_time: 任务结束时间。必须按照该格式传入时间参数, 其中年月日只在 time_enable 为 0 时生效(年月日必须为当天的准确时间)。  end_time: Task end time.must be passed

into time parameter in this format, especially day, month and year are only available when tiem_enalbe is set to be 0 (day, month and year must be the exact time of the day)
repeat: 任务重复周期。 在 time_enable 为 1 时有效。 repeat: Repetitive cycles of periodic tasks. is available when time_enable is 1.
sensor_type: 任务执行条件的传感器类型。 sensor_type: Sensor types of the task execution condition.
sensor_trigger: 任务执行条件的传感器值变化趋势. sensor_trigger: The change in the trend of the sensor value of the task execution condition.
sensor_value: 任务执行条件的传感器值。 sensor_value: The sensor value of the task execution condition.
device_mac: 执行任务的设备的 mac 地址。 device_mac: Mac address of the device to execute tasks.
device_id: 执行任务的设备的 ID。(该值从 probe_list 中返回) device_id: ID of the device to execute tasks.(the value returned from probe_list)
device_key: 执行任务的设备的 key. (该值从 probe_list 中返回) device_key: Key of the device to execute tasks. (the value returned from probe_list)
device_type: 执行任务的设备的类型. (该值从 probe_list 中返回) device_type: Type of the device to execute tasks. (the value returned from probe_list)
task_data: 执行任务的命令。(当执行任务设备为SP2时无效, 若是RM2, 则设置为rm2_code 中返回的 data) task_data: Command to execute tasks.(It is invalid when device to execute tasks is SP2. If the device is RM2, it is set to be the returned data of rm2_code.)
status: 执行任务的设备要设置的状态. (仅对执行任务设备为 SP2 时有效，非 SP2 设备可取消该字段) status: Status of the device to execute tasks. (Only is valid when the device is SP2, otherwise the field should be canceled.)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
“list”:[ “list”:[
{ {
"mac": "00:11:22:33:44:55", ,
"time_enable":0, :0,
"task_enable":1, :1,
"index":0, :0,
"start_time":"10:30", ,
"end_time":"18:30", ,
"repeat":7, :7,
"sensor_type":0, :0,
"sensor_trigger": 0, : 0,
"sensor_value":25.0, :25.0,
"task_name":"打开电视机" Turn on TV.
}, },

```
                {                   {
                     "mac": "55:44:33:22:11:00", ,
                    "time_enable":1, :1,
                    "task_enable":1, :1,
                    "index":2,   :2,
                    "start_time":"10:30",    ,
                    "end_time":"17:30",  ,
                    "repeat":5,  :5,
                    "sensor_type":2, :2,
                    "sensor_trigger": 1, : 1,
                    "sensor_value":3.0,  :3.0,
                    "task_name":"打开音响"  Open the audio player
                }                   }
            ],              ],
     "msg":"Get a1 task list success"  Get a1 task list success
}   }
```


code: 返回码  code: return code

list: 任务列表  list: Task list

mac: 执行任务的设备 mac 地址。字符串  mac: Mac address of the device to execute tasksString

time_enable: 时间使能。 整型  time_enable: Time enable. Integer

task_enable: 任务使能。 整形  task_enable: Task enable. Integer

index: 任务编号,(删除任务时使用)。 整形  index: Task Number, (used when deleting task) Integer

start_time: 任务执行的开始时间。 字符串  start_time: Start time of task executing String

end_time: 任务执行的结束时间。 字符串  end_time: End time of task executing String

repeat: 任务的重复周期，以周为一个周期，0 表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。 整型  repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday). Integer

sensor_type: 任务执行条件的传感器类型, (0: 温度  1:湿度  2:光照  3:空气质量  4:噪声)。 整型  sensor_type: Sensor types of the task execution condition (0: temperature  1:humidity  2:illuminance  3:air quality  4:noise). Integer

sensor_trigger: 任务执行条件传感器值发生变化的趋势, (0:上升  1:下降)。 整型  sensor_trigger: The change in the trend of the sensor value of the task execution condition (0: rise 1: fall) Integer

sensor_value: 任务执行条件传感器的值. 浮点型  sensor_value: The sensor value of the task execution condition. floating point type

task_name: 执行任务的名称。 字符串  task_name: Name of the executed task. String

msg : 返回消息  msg : return message

</pre> </pre>

h3. 1.24 a1_del_task 请求参数说明 h3. 1.24 a1_del_task Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":164, :164,
"command":"a1_del_task", ,
"mac":"00:11:22:33:44:55", ,
"index": 0 : 0
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
mac: 要操作的设备的 MAC 地址。(必填） mac: MAC address of the device needs to be manipulated.(mandatory)
index: 要删除的任务的 index.(该值从返回的列表中获取) index: The index of tasks need to be deleted. (The value is obtained from the returned list)
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
“list”:[ “list”:[
{ {
"mac": "00:11:22:33:44:55", ,
"time_enable":0, :0,
"task_enable":1, :1,
"index":0, :0,
"start_time":"10:30", ,
"end_time":"18:30", ,
"repeat":7, :7,
"sensor_type":0, :0,
"sensor_trigger": 0, : 0,
"sensor_value":25.0, :25.0,
"task_name":"打开电视机" Turn on TV.
}, },
{ {
"mac": "55:44:33:22:11:00", ,
"time_enable":1, :1,
"task_enable":1, :1,
"index":2, :2,
"start_time":"10:30", ,
"end_time":"17:30", ,

"repeat":5,  :5,
"sensor_type":2, :2,
"sensor_trigger": 1, : 1,
"sensor_value":3.0,  :3.0,
"task_name":"打开音响"  Open the audio player
}  }
],  ],
"msg":"Get a1 task list success"  Get a1 task list success
}  }
code: 返回码  code: return code
list: 任务列表  list: Task list
mac: 执行任务的设备 mac 地址。字符串  mac: Mac address of the device to execute tasks.String
time_enable: 时间使能。 整型  time_enable: Time enable Integer
task_enable: 任务使能。 整形  task_enable: Task enable Integer
index: 任务编号,(删除任务时使用)。 整形  index: Task Number, (used when deleting task) Integer
start_time: 任务执行的开始时间。 字符串  start_time: Start time of task executing String
end_time: 任务执行的结束时间。 字符串  end_time: End time of task executing String
repeat: 任务的重复周期，以周为一个周期，0 表示只执行一次 (bit0:周日, bit1:周一, bit2:周二, bit3:周三, bit4:周四, bit5:周五, bit6: 周六)。 整型  repeat: Repetitive cycles of periodic tasks in one-week iteration, 0 means task is only executed once (bit0: Sunday, bit1:Monday, bit2:Tuesday, bit3:Wednesday, bit4:Thursday, bit5:Friday, bit6: Saturday). Integer
sensor_type: 任务执行条件的传感器类型, (0: 温度  1:湿度  2:光照  3:空气质量  4:噪声)。 整型  sensor_type: Sensor types of the task execution condition (0: temperature  1:humidity  2:illuminance  3:air quality  4:noise). Integer
sensor_trigger: 任务执行条件传感器值发生变化的趋势, (0:上升  1:下降)。 整型  sensor_trigger: The change in the trend of the sensor value of the task execution condition (0: rise 1: fall) Integer
sensor_value: 任务执行条件传感器的值. 浮点型  sensor_value: The sensor value of the task execution condition. floating point type
task_name: 执行任务的名称。 字符串  task_name: Name of the executed task. String
msg : 返回消息  msg : return message
</pre>  </pre>

h1. easyconfig 请求参数说明  h1. easyconfig Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{  {
"api_id":10000,  :10000,

"command":"easyconfig", ,
"ssid":"abcdefg", ,
"password":"11111111", ,
"broadlinkv2":1, :1,
"dst":"192.168.3.1" 192.168.3.1
} }

api_id: 命令 ID。 api_id: command ID。
command: 命令。 （可选） command: command. (Optional)
ssid: 设备将要连接的网络名称。(必填) ssid: Name of network that the device will connect to.(mandatory)
password: 设备将要连接的网络密码。(必填) password: Password of network that the device will connect to.(mandatory)
broadlinkv2: 将要配置的设备是否采用二代（含）以后的 broadlink 无线模块。(选填，默认是二代模块) broadlinkv2: Whether the device need to be configured uses the second and later generation broadlink wireless module or not.(optional, the default setting is second generation module)
dst: 网关地址。(选填，填写网关地址可加快配置速度) dst: The gateway address.(optional, filling the gateway address can accelerate the configuration speed)

*注意：该接口非线程安全* *Note: the interface is not thread-safe*
</pre> </pre>
<pre>Retval: <pre>Retval:
{ {
"code":0, :0,
"msg":"easyconfig success" easyconfig success
} }

code: 返回码 0 表示成功，1 表示超时，-1 表示用户取消或者其他错误 code: return code 0 means success, 1 means timeout, -1 means users cancel it or other error.
msg : 返回消息 msg : return message
</pre> </pre>

h1. 取消 easyconfig 请求参数说明 h1. 取 消 easyconfig Request Parameter Specification

<pre>Arguments: <pre>Arguments:
{ {
"api_id":10001, :10001,
"command":"cancel_easyconfig" cancel_easyconfig
} }

api_id: 命令 ID。  api_id: command ID。
command: 命令。 （可选）  command: command. (Optional)

*注意：该接口非线程安全.*  *Note: the interface is not thread-safe*
</pre> </pre>
<pre>Retval:  <pre>Retval:
{  {
  "code":0,  :0,
  "msg":"easyconfig success" easyconfig success
}  }

code: 返回码  0 表示成功，1 表示超时，-1 表示用户取消或者其他错误 code:  return code 0 means success, 1 means timeout, -1 means users cancel it or other error.
msg : 返回消息  msg : return message
</pre> </pre>

h2.  *注意:* _返回的 code 除了已经说明的，其余返回的 code 对应的错误信息如下：_
h2.  *Note:* _Except the explained return code, the corresponding error informations of rest return code are shown below_

_0:  成功_  _0:  success_

_-1:  设备所在网络已改变或网络密码已经修改。_  _-1: The network that devices are connected to or the password of the network has been changed. _

_-2:  设备已在其他地方登录，如需继续控制，请重新登录(针对 rm1/sp1)_
_-2:  The device has logged in at another location, if there is a need to continue to control the device, please log in again (Especially for rm1/sp1)_

_-3:  设备不在线_  _-3:  The device is offline_

_-4:  不支持的操作_  _-4:  Unsupported operations_

_-5:  空间满_  _-5:  Storage space is full_

_-6:  数据结构异常_  _-6:  Data structure is abnormal_

_-7:  设备已经复位，需进入局域网重新认证。(针对 sp1/rm1 以外的设备)_
_-7:  Device has been reset, and need to enter the LAN for a new configuration. (especially for the device beyond sp1/rm1)_

_-100: 超时_ _-100: Timeout_

_-101: 网络线程找不到该设备。_ _-101: Network thread cannot find the device_

_-102: 内存不足_ _-102: There is not enough memory_

_-103: 设备未初始化_ _-103: The device is uninitialized_

_-104: 网络线程已暂停_ _-104: Network thread has been suspended_

_-105: 返回消息类型错误_ _-105: Type of return message is error_

_-106: 操作过于频繁_ _-106: Operations are too frequent_

_-107: 服务器已拒绝该 license 操作,请联系客服处理_ _-107: The server has declined the operation of this license, please contact the custmer sevice_

_-108: 设备不在局域网中_ _-108: The device is not connected to the LAN_

_一10000: 未知错误_ _一10000: Unknown error_