



# Cassia RESTful API使用说明

V1.1

发布日期：2022年6月6日

版本	更新内容	发布日期
V1.0	初版发布	20200219
V1.1	调整格式，增加容器内调用	20220606

# 目录

一、概述	2
二、使用方式	2
1、区别介绍	2
2、云端专有API	3
3、通用API	3
4、开启本地API	3
三、SSE介绍	4
四、云端专有API	5
1、获取token API	5
2、获取AC上单个蓝牙网关的状态	6
3、获取AC上所有蓝牙网关的状态	7
4、监控AC上所有蓝牙网关状态变化	9
5、其他API	9
五、通用API介绍	10
1、扫描BLE终端设备	10
2、连接/断开连接BLE终端设备	10
3、对BLE终端设备进行写指令	12
4、对BLE终端设备进行读指令	12
5、打开/关闭BLE终端设备通知	13
6、接收BLE终端设备数据	13
7、监控BLE终端设备连接状态	13
8、获取蓝牙网关连接设备列表	14
9、获取BLE终端设备服务列表	15
10、远程重启蓝牙网关	18
11、通过蓝牙网关发送广播数据	18
12、取 BLE 连接的 RSSI (v2.0 及更高版本)	20
六、蓝牙网关自动选择	21
1、功能开启	21
2、连接BLE终端设备	22
3、断连BLE终端设备	23
七、组合SSE API	24
1、创建组合SSE	24
2、开启/关闭扫描	26
3、开启/关闭通知	28
4、开启/关闭连接状态变化报告	29
5、开启/关闭蓝牙网关状态变化报告	30
八、定位API	32
九、蓝牙调试工具和在线文档	33

## 一、概述

Cassia物联网控制器（AC）和蓝牙网关中，均集成了一套Cassia RESTful API（Application Programming Interface，应用程序编程接口）接口，供开发使用。第三方开发人员和设备制造商可以将Cassia RESTful API集成到其应用平台或者服务器中，与Cassia AC和Cassia 蓝牙网关进行无缝结合。Cassia RESTful API兼容多种主流开发形式（例如Android，IOS，Windows和Linux等）和编程语言（例如Java，C#和node.js等）。

本文档帮助您使用常见的 Cassia RESTful API 及其相关服务。所有的Cassia RESTful API介绍请参考：

<https://github.com/CassiaNetworks/CassiaSDKGuide/wiki/RESTful-API>

Cassia RESTful API主要功能如下：

- 扫描低功耗蓝牙（BLE）设备广播包
- 连接低功耗蓝牙（BLE）设备
- 执行低功耗蓝牙（BLE）设备的读/写指令
- 接收低功耗蓝牙（BLE）设备数据（notification/indication）
- 监控低功耗蓝牙（BLE）设备连接状态变化
- 监控低功耗蓝牙（BLE）设备连接信号强度变化

说明：本文档中，会用Cassia AC或AC代指物联网控制器，蓝牙AP或AP代指蓝牙网关，API代指RESTful API，API接口中，出现的gateway 和 router 描述，均为蓝牙网关。

## 二、使用方式

Cassia 提供两种形式的 RESTful API调用方式。

本地调用API：调用蓝牙网关中集成的 API（应用程序和蓝牙网关通常位于同一网络中）。

云端调用API：调用物联网控制器（Cassia AC）中集成的API（应用程序可以放置在云端）。

### 1、区别介绍

- 云端调用API，需要在{ac-ip}之后包含“/api”，本地调用API 不需要此参数。
- 云端调用API，需要包含参数“&mac=<gateway mac>”用以识别使用指定网关，本地调用API不需要此参数。
- 云端调用API，需要包含参数“&access\_token=”作为安全验证，本地API不需要此参数。

云端调用示例：

```
http://82.157.174.197/api/gap/nodes/?active=1&event=1&chip=0&mac=CC:1B:E0:E0:DC:88&access_token=xxxxxxxxx
```

本地调用示例：

```
http://192.168.31.108/gap/nodes/?active=1&event=1&chip=0
```

## 2、云端专有API

以下 API只在Cassia AC中集成，只能通过Cassia AC使用。其他API在Cassia AC和蓝牙网关中均有集成，并且可以得到相同的调用结果。

- 获取token API
- SSE组合API
- 获取AC上单个蓝牙网关的状态
- 获取AC上所有蓝牙网关的状态
- 监控AC上所有蓝牙网关状态变化
- 网关优选功能
- BLE终端设备定位

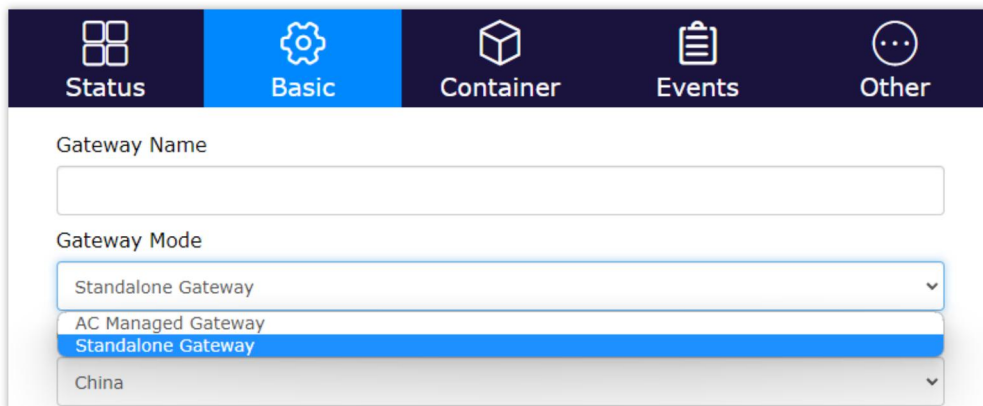
## 3、通用API

- 扫描BLE终端设备
- 连接/断开连接BLE终端设备
- 对BLE终端设备进行写操作
- 对BLE终端设备进行读操作
- 打开/关闭BLE终端设备通知
- 接收BLE终端设备数据
- 监控BLE终端设备连接状态
- 获取蓝牙网关连接设备列表
- 获取BLE终端设备服务列表
- 远程重启蓝牙网关
- 通过蓝牙网关发送广播数据
- 获取BLE终端设备连接的RSSI
- 配对

## 4、开启本地API

Cassia AC上的API默认为开启状态，可直接使用。

蓝牙网关上的API，工作在本地模式（standaloneGateway）下的蓝牙网关会默认开启，工作在AC模式（AC Managed Gateway）下的蓝牙网关，需要在AC上开启。



蓝牙网关工作在Standalone Gateway模式下，本地API自动开启



通过AC为蓝牙网关开启API（设置为ON并保存）

### 三、SSE介绍

Server Sent Events简称SSE，是一种浏览器通过HTTP连接从服务器接收自动更新的技术。SSE API被W3C标准化为HTML5的一部分。SSE用于向浏览器客户端发送消息更新或连续的数据流。

Cassia RESTful API 中，有六个API使用了 SSE：

- 扫描低功耗蓝牙（BLE）设备广播包
- 接收低功耗蓝牙（BLE）设备数据（notification/indication）
- 监控低功耗蓝牙（BLE）设备连接状态变化
- 监控低功耗蓝牙（BLE）设备连接信号强度变化
- 监控蓝牙网关状态变化（AC专有）
- 使用组合SSE（AC专有）

每个SSE响应都以“data:”开头。调试时，您可以将SSE的URL输入到Web浏览器中，然后您将看到Web浏览器中的SSE输出。

在程序中，如果像普通的HTTP请求一样调用接口，SSE请求不会返回任何数据，因为普通的HTTP请求只有在完成后才会返回输出。

SSE需要手动终止，否则它将继续运行，直到出现错误。如keep-alive丢失、socket 错误、网络问题等。此外，在调用SSE时应该监视该线程，如果它被错误或任何意外事件中中断，您需要重新启动它。Nodejs 库 'eventsouce' 自动处理SSE 重新连接。对于其他语言，重新连接可能需要由用户应用程序处理。

注意：SSE API永远不会结束并以流的形式发送数据，因此它会挂起页面。您应该添加以下代码段（以扫描为例，PHP代码）：

```
if ($stream = fopen($url_for_scan, 'r')) {
    while(($line=fgets($stream))!== false){
        echo $line;
    }
}
```

## 四、云端专有API

### 1、获取token API

云端调用API需要事先获取token，token是AC的一种验证机制。此接口为使用AC的API时，通过开发者账号和开发者密码进行认证，获取access\_token，后续每次调用AC的API都需要添加此token。

**注意：**此 API 仅 Cassia AC 支持。

获取token接口为HTTP协议的POST方法。

获取的access\_token有效期为1小时，所以每隔小于1小时都需要重新获取。

API URL:

http://{ac-ip}/api/oauth2/token

URL参数

ac-ip	AC的IP地址或者域名
-------	-------------

headers参数

Authorization	Basic Y2Fzc2lhOmNhc3NpYQ==
Content-Type	application/x-www-form-urlencoded

body参数

grant_type	client_credentials
------------	--------------------

**注意：**Authorization参数为开发者账号和开发者密码（AC中配置）使用Basic64加密得到。例如：



AC中配置的开发者账号和开发者密码均为cassia

将{开发者账号:开发者密码}进行basic64加密，得到 Y2Fzc2lhOmNhc3NpYQ==



之后，请在Y2Fzc2lhOmNhc3NpYQ==前，添加basic+空格，得到

basic Y2Fzc2lhOmNhc3NpYQ==

即为Authorization的value值。

调用示例: python

```
import requests
url = "http://62.234.110.11/api/oauth2/token"
payload='grant_type=client_credentials'
headers = {
    'Content-Type': 'application/x-www-form-urlencoded',
    'Authorization': 'Basic Y2Fzc2lhOmNhc3NpYQ=='
}
response = requests.request("POST", url, headers=headers, data=payload)
print(response.text)
```

## 2、获取AC上单个蓝牙网关的状态

您可以使用下面的 API 来获取AC上的蓝牙网关的状态，无论在线还是离线。

**注意：**此 API 仅 Cassia AC 支持。

获取蓝牙网关状态接口为HTTP协议的GET方法。

API URL:

[http://{ac-ip}/api/cassia/hubs/<gateway-mac>?&access\\_token=<token>](http://{ac-ip}/api/cassia/hubs/<gateway-mac>?&access_token=<token>)

该接口返回的结果为JSON 格式。

蓝牙网关在线返回示例:

```
{
  "id":10,
  "ip":"124.193.83.244",
  "ap":{"uplink\":"wired\""},
  "mac":"CC:1B:E0:E1:8D:A0",
  "name":"Cassia Gateway",
  "time":1654577823,
  "model":"S2000",
  "grp":"OFFICE",
  "uptime":0,
  "status":"online",
  "localip":"10.100.6.104",
  "version":"2.1.1.2201200330",
  "position":"",
  "container":{"status_code":2,"status":"not_support"},
  "reserved0":"mqtt",
  "reserved1":"",
  "reserved2":null,
  "reserved3":null,
  "reserved4":null,
  "groupcolor":"",
  "online_time":1652641156,
  "offline_time":0,
  "update_status":"update_ok",
  "update_reason":"",
  "update_version":"",
  "update_progress":0
}
```

蓝牙网关离线返回示例:

```
{
  "id":184,
  "ip":"221.218.212.91",
  "ap":{"}",
  "mac":"CC:1B:E0:E2:1E:A0",
  "name":"Cassia Gateway",
  "time":1654578414,
  "model":"S2000",
  "grp": "",
  "uptime":0,
  "status":"offline",
  "localip":"192.168.31.108",
  "version":"2.1.1.2201030330",
  "position": "",
  "container": {},
  "reserved0": "capwap",
  "reserved1": null,
  "reserved2": null,
  "reserved3": null,
  "reserved4": null,
  "groupcolor": "",
  "online_time":1654578285,
  "offline_time":1654578414,
  "update_status": "update_ok",
  "update_reason": "",
  "update_version": "",
  "update_progress":0
}
```

蓝牙网关未在AC注册返回示例:

```
Cannot read property 'container' of null
```

### 3、获取AC上所有蓝牙网关的状态

您可以使用下面的 API 来获取AC上的蓝牙网关的状态，无论在线还是离线。

**注意：**此 API 仅 Cassia AC 支持。

获取蓝牙网关状态接口为HTTP协议的GET方法。

**API URL:**

`http://{ac-ip}/api/cassia/hubs?&access_token=<token>`

返回结果是一个 JSON 对象数组。

返回数据示例:

E1000等支持container的网关，还会返回container和APP的相关信息，文档中不再描述，请自行测试获取或浏览在线文档。该接口可通过浏览器测试。



```

[
  {
    "id":10,
    "ip":"124.193.83.244",
    "ap":{"\uplink\":"\wired\"},
    "mac":"CC:1B:E0:E1:8D:A0",
    "name":"Office_Test_Kang",
    "time":1654579525,
    "model":"S2000",
    "grp":"OFFICE",
    "uptime":0,
    "status":"online",
    "localip":"10.100.6.104",
    "version":"2.1.1.2201200330",
    "position":"",
    "container":{"status_code":2,"status":"not_support"},
    "reserved0":"mqtt",
    "reserved1":"",
    "reserved2":null,
    "reserved3":null,
    "reserved4":null,
    "groupcolor":"undefined",
    "online_time":1654578465,
    "offline_time":1654578460,
    "update_status":"update_ok",
    "update_reason":"",
    "update_version":"",
    "update_progress":0
  },{
    "id":187,
    "ip":"221.218.212.91",
    "ap":{"\uplink\":"\wireless\"},
    "mac":"CC:1B:E0:E2:1E:A0",
    "name":"Cassia Gateway",
    "time":1654579538,
    "model":"S2000",
    "grp":"",
    "uptime":0,
    "status":"online",
    "localip":"192.168.31.108",
    "version":"2.1.1.2201030330",
    "position":"",
    "container":{"status_code":2,"status":"not_support"},
    "reserved0":"mqtt",
    "reserved1":null,
    "reserved2":null,
    "reserved3":null,
    "reserved4":null,
    "groupcolor":"",
    "online_time":1654579314,
    "offline_time":0,
    "update_status":"update_ok",
    "update_reason":"",
    "update_version":"",
    "update_progress":0
  }
]

```

#### 4、监控AC上所有蓝牙网关状态变化

您可以使用此 API 持续监控AC上的蓝牙网关的状态。此接口为SSE接口，需要保持调用。其中

- "first":"first"代表网关首次上线，为空则为离线后上线。
- "status":"online"代表上线事件，offline为离线事件。

注意：此 API 仅Cassia AC 支持。

获取蓝牙网关状态接口为HTTP协议的GET方法。

API URL:

`http://{ac-ip}/api/cassia/hubStatus?&access_token=<token>`

返回结果是一个 JSON 对象。

```
:keep-alive
data: {
  "ap": "CC:1B:E0:E0:DC:88",
  "dataType": "ap_state",
  "status": "online",
  "online_time": 1654579989207,
  "model": "E1000",
  "ip": "124.193.83.244",
  "mac": "CC:1B:E0:E0:DC:88",
  "version": "2.1.1.2112271145",
  "uptime": 0,
  "first": "first",
  "localip": "192.169.199.25"
}
:keep-alive
data: {
  "ap": "CC:1B:E0:E2:1E:A0",
  "dataType": "ap_state",
  "status": "online",
  "online_time": 1654580241817,
  "model": "S2000",
  "ip": "221.218.212.91",
  "mac": "CC:1B:E0:E2:1E:A0",
  "version": "2.1.1.2201030330",
  "uptime": 0,
  "first": " ",
  "localip": "192.168.31.108"
}
:keep-alive
data: {
  "dataType": "ap_state",
  "ap": "CC:1B:E0:E2:1E:A0",
  "mac": "CC:1B:E0:E2:1E:A0",
  "status": "offline",
  "offline_time": 1654580085918
}
```

#### 5、其他API

网关优选、组合SSE和定位API使用比较复杂，请参照后面的独立章节。

## 五、通用API介绍

除AC专有API以外，其他API在Cassia AC和蓝牙网关中均有集成，并且可以得到相同的调用结果。

其中，在容器（container）中，也可以本地调用API。容器内调用，使用固定IP地址10.10.10.254，容器的相关介绍，请参照蓝牙网关本地使用文档和容器的说明文档。

### 1、扫描BLE终端设备

使用网关通过 AC 扫描蓝牙终端（BLE）设备，可通过扫描API实现。此接口为SSE接口，需要保持调用。

获取蓝牙网关状态接口为HTTP协议的GET方法。

AC调用：

```
http://{ac-ip}/api/gap/nodes?event=1&mac=<gateway-mac>
&access_token=<token>
```

本地调用：

```
http://{gateway-ip}/gap/nodes?event=1
```

容器内调用：

```
http://10.10.10.254/gap/nodes?event=1
```

可选参数：

参数	描述
active	（可选）：0表示被动扫描，1表示主动扫描。不指定则默认执行被动扫描。
filter_duplicates	（可选）：0 关闭，1 开启。过滤重复记录。默认值为0。>1000(ms) 计时器重新启动重复过滤器。
filter_mac	（可选）：根据MAC筛选，不指定则默认扫描所有设备。新版固件支持通配符*，例如CC:1B*或*EA:0A，分别代表CC:1B开头，或EA:0A结尾的所有设备。
filter_name	（可选）：根据终端name筛选，不指定则默认扫描所有设备。新版固件支持通配符*，例如AB*或*EF，分别代表AB开头，或者EF结尾的所有设备。
filter_rssi	（可选）：根据新号强度RSSI筛选，不指定则默认扫描所有设备。例如rssi=-60，则扫描新号强度优于（大于）-60的所有设备。
timestamp	（可选）：默认为0，1 在扫描数据中添加时间戳，例如：data: {"name": "cass", "timestamp": "2021-03-08 10 :07:17.723 CST", "evtType": "0", "rssi": "-69", "adData": "02010612094341535349412D424541434F4E2D333335", "bdaddrs": [{"bdaddr": "CC:DD:EE:03:00:23 ", "bdaddrType": "random"}]}

### 2、连接/断开连接BLE终端设备

调用以下接口，可以实现指定蓝牙网关连接指定的蓝牙设备。

连接接口为HTTP协议的POST方法。

AC调用：

```
http://{ac-ip}/api/gap/nodes/<device-mac>/connection?mac=<gateway-mac>
&access_token=<token>
```

本地调用：

`http://{gateway-ip}/gap/nodes/<device-mac>/connection`

容器内调用：

`http://10.10.10.254/gap/nodes/<device-mac>/connection`

**注意：**网关不能同时处理多个连接请求。用户需要串行处理请求，即等待上一次连接的响应，然后再调用下一个连接请求。如果需要连接多个设备而不设置等待间隔，请查看批量连接 API，可以一次下发多个终端的连接请求，网关自动处理等待间隔。

参数	描述
type	(可选)：BLE终端设备的地址类型， <b>public</b> 或 <b>random</b> 。如果未指定，则默认为 <b>public</b> 。type错误会导致连接失败。
timeout	(可选)：以毫秒为单位，如果在此时间内无法完成连接，请求将超时。默认超时为 <b>5,000</b> 毫秒。取值范围为 <b>200ms - 20000ms</b> 。
discovergatt	(可选)：默认为 <b>1</b> ， <b>1</b> 表示网关会使用在之前连接时发现并缓存 <b>GATT</b> 数据库。它将节省服务发现 API 的时间，但可能不会更新信息。值为 <b>0</b> 表示网关不使用缓存的 <b>GATT</b> 数据库。当用户调用服务发现 API 时，网关应该从 <b>BLE</b> 终端设备读取新的 <b>GATT</b> 服务和特征。

本地调用示例（没有“/api”和“mac=”）：

```
curl -X POST -H "content-type: application/json" -d
'{"timeout":"10000","type":"public"}'
'http://172.16.10.6/gap/nodes/CC:1B:E0:E8:09:2B/connection'
```

返回值示例：

```
Status-Line : HTTP/1.1 200 OK/r/n
Header : (general-header)
Message-body: text/plain
OK
```

调用以下接口，可以实现指定蓝牙网关断开已经连接的蓝牙设备。

断开连接接口为HTTP协议的DELETE方法。

AC调用：

`http://{ac-ip}/api/gap/nodes/<device-mac>/connection?mac=<gateway-mac>
&access_token=<token>`

本地调用：

`http://{gateway-ip}/gap/nodes/<device-mac>/connection`

容器内调用：

`http://10.10.10.254/gap/nodes/<device-mac>/connection`

### 3、对BLE终端设备进行写指令

调用以下接口，可以向已连接的BLE终端设备写入指令。

需要根据蓝牙通信协议，知晓写入指令的UUID，并根据获取的服务列表找到对应的HANDLE，一般为服务列表中的handle +1，可以通过蓝牙调试工具下发指令，并查看接口日志中的write指令，确认正确的handle。

写入指令接口为HTTP协议的GET方法。

AC调用：

```
http://{ac-ip}/api/gatt/nodes/<device-mac>/handle/<handle>/value/<value>?  
mac=<gateway-mac>&access_token=<token>
```

本地调用：

```
http://{gateway-ip}/gatt/nodes/<device-mac>/handle/<handle>/value/<value>
```

容器内调用：

```
http://10.10.10.254/gatt/nodes/<device-mac>/handle/<handle>/value/<value>
```

参数	描述
<handle>	写入指令的UUID对应的HANDLE
<value>	写入的指令
noresponse	(可选)：0或1，1为写入指令而不返回结果，默认为0。

本地调用的示例：

```
curl --location --request GET  
'http://192.168.31.108/gatt/nodes/CB:8A:DA:CC:5F:9C/handle/20/value/0100  
?noresponse=1'
```

返回值示例：默认，noresponse=0时

```
{"deviceMac": "CB:8A:DA:CC:5F:9C", "handle": "20", "value": "0100", "noresponse": "true"}
```

### 4、对BLE终端设备进行读指令

调用以下接口，可以读取已连接的BLE终端设备中，特定UUID的值。

需要根据蓝牙通信协议，知晓需要读取的UUID，并根据获取的服务列表找到对应的HANDLE，一般为服务列表中的handle +1，可以通过蓝牙调试工具读取，并查看接口日志中的read指令，确认正确的handle。

读取指令接口为HTTP协议的GET方法。

AC调用：

```
http://{ac-ip}/api/gatt/nodes/<device-mac>/handle/<handle>/value?  
mac=<gateway-mac>&access_token=<token>
```

本地调用：

http://{ac-ip}/gatt/nodes/<device-mac>/handle/<handle>/value

容器内调用:

http://10.10.10.254/gatt/nodes/<device-mac>/handle/<handle>/value

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/nHeader : (general-header)Message-body: application/json
{
  "handle": "36",
  "value": "56312e362e31"
}
```

## 5、打开/关闭BLE终端设备通知

通过此接口, 可以打开/关闭BLE终端设备, 并使BLE终端设备开始或者停止向蓝牙网关发送数据。

使用方法同写指令, 区别为写入的值为固定值:

开启: notification: 写入 0100

indication: 写入 0200

关闭: notification: 写入 0000

indication: 写入 0000

## 6、接收BLE终端设备数据

通过此接口, 使用户可以读取到BLE终端设备向蓝牙网关发送的数据 (5中开启的开关)。此接口为SSE接口, 需要保持调用。

接收BLE终端设备数据接口为HTTP协议的GET方法。

AC调用:

http://{ac-ip}/api/gatt/nodes?event=1&mac=<gateway-mac>  
&access\_token=<token>

本地调用:

http://{gateway-ip}/gatt/nodes?event=1

容器内调用:

http://10.10.10.254/gatt/nodes?event=1

## 7、监控BLE终端设备连接状态

通过以下API, 可以实时获取到指定蓝牙网关上, 所有BLE终端设备到的连接状态变化。此接口为SSE接口, 需要保持调用。

获取蓝牙网关状态接口为HTTP协议的GET方法。

AC调用:

http://{ac-ip}/api/management/nodes/connection-state?mac=<gateway-mac>

&access\_token=<token>

本地调用:

http://{gateway-ip}/management/nodes/connection-state

容器内调用:

http://10.10.10.254/management/nodes/connection-state

当BLE终端设备的状态从断开连接到连接, 或从连接到断开连接时, 您会收到相应的响应。例如:

```
data: {"handle":"CC:1B:E0:E8:0D:F2","connectionState":"connected"}data: {"handle":"88:C6:26:92:58:77","connectionState":"disconnected"}
```

## 8、获取蓝牙网关连接设备列表

通过以下接口, 用户可以获取到指定蓝牙网关上, 当前已经连接的BLE终端设备列表。

获取蓝牙网关连接设备列表接口为HTTP协议的GET方法。

AC调用:

http://{ac-ip}/api/gap/nodes?connection\_state=connected&mac=<gateway-mac>  
&access\_token=<token>

本地调用:

http://{gateway-ip}/gap/nodes?connection\_state=connected

容器内调用:

http://10.10.10.254/gap/nodes?connection\_state=connected

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/nHeader : (general-header)Message-body: application/json
{
  "nodes": [{
    "type": "random",
    "bdaddrs": {
      "bdaddr": "EF:A3:E6:94:CD:2D",
      "bdaddType": "random"
    },
    "chipId": 0,
    "handle": "",
    "name": "",
    "connectionState": "connected",
    "id": "EF:A3:E6:94:CD:2D",
    "pairStatus": "paired"
  }]
}
```

注意：“pairStatus”值可以是“paired”（当BLE终端设备与蓝牙网关配对时）或“none”（当BLE终端设备未与蓝牙网关配对时）。

## 9、获取BLE终端设备服务列表

通过以下API接口，用户可以获取到已连接的指定BLE终端设备的服务列表，或者单独获取characteristics、descriptors等。

获取BLE终端设备的服务列表接口为HTTP协议的GET方法。

**发现所有服务：**

AC调用：

```
http://{ac-ip}/api/gatt/nodes/<device-mac>/services?mac=<gateway-mac>
&access_token=<token>
```

本地调用：

```
http://{gateway-ip}/gatt/nodes/<device-mac>/services
```

容器内调用：

```
http://10.10.10.254/gatt/nodes/<device-mac>/services
```

返回值示例：

```
Status-Line : HTTP/1.1 200 OK/r/nHeader : (general-header)Message-body:
application/json

[
  {
    "uuid": "00001800-0000-1000-8000-00805f9b34fb",
    "primary": true,
    "handle": 1
  }, {
    "uuid": "00001801-0000-1000-8000-00805f9b34fb",
    "primary": true,
    "handle": 8
  }, {
    "uuid": "0000fd00-0000-1000-8000-00805f9b34fb",
    "primary": true,
    "handle": 9
  }, {
    "uuid": "0000180d-0000-1000-8000-00805f9b34fb",
    "primary": true,
    "handle": 20
  }, {
    "uuid": "0000180f-0000-1000-8000-00805f9b34fb",
    "primary": true,
    "handle": 26
  }, {
    "uuid": "0000180a-0000-1000-8000-00805f9b34fb",
    "primary": true,
    "handle": 30
  }
]
```



### 发现所有characteristics:

AC调用:

```
http://{ac-ip}/api/gatt/nodes/<device-mac>/characteristics?mac=<gateway-mac>&access_token=<token>
```

本地调用:

```
http://{gateway-ip}/gatt/nodes/<device-mac>/characteristics
```

容器内调用:

```
http://10.10.10.254/gatt/nodes/<device-mac>/characteristics
```

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/nHeader : (general-header)Message-body: application/json

[ {
  "handle": 3,
  "properties": 10,
  "uuid": "00002a00-0000-1000-8000-00805f9b34fb"
}, {
  "handle": 5,
  "properties": 2,
  "uuid": "00002a01-0000-1000-8000-00805f9b34fb"
}, {
  "handle": 7,
  "properties": 2,
  "uuid": "00002a04-0000-1000-8000-00805f9b34fb"
}, {
  "handle": 11,
  "properties": 16,
  "uuid": "0000fd09-0000-1000-8000-00805f9b34fb"
}, {
  "handle": 14,
  "properties": 4,
  "uuid": "0000fd0a-0000-1000-8000-00805f9b34fb"
} ]
```

### 发现一个服务中的所有characteristic:

AC调用:

```
http://{ac_ip}/api/gatt/nodes/<device_mac>/services/<service_uuid>/characteristics?mac=<gateway-mac>&access_token=<token>
```

本地调用:

```
http://{gateway_ip}/api/gatt/nodes/<device_mac>/services/<service_uuid>/characteristics?
```

容器内调用:

```
http://10.10.10.254/api/gatt/nodes/<device_mac>/services/<service_uuid>/characteristics?
```

返回值示例:

```
Response example:
Status-Line: HTTP/1.1 200 OK/r/n
Header: (general-header)
Message-body: text/plain
[{"handle":48,"properties":10,"uuid":"00002a06-0000-1000-8000
00805f9b34fb"}]
```

发现一个**characteristic** 中的所有**descriptors**:

AC调用:

```
http://{ac_ip}/api/gatt/nodes/<device_mac>/characteristics/<characteristic_
uuid>/descriptors? mac=<gateway_mac>&access_token=<token>
```

本地调用:

```
http://{gateway_ip}/api/gatt/nodes/<device_mac>/characteristics/<characteristic_
uuid>/descriptors?
```

容器内调用:

```
http://10.10.10.254/api/gatt/nodes/<device_mac>/characteristics/<characteristic_
uuid>/descriptors?
```

返回值示例:

```
Response example:
Status-Line: HTTP/1.1 200 OK/r/n
Header: (general-header)
Message-body: text/plain
[{"handle":48,"properties":10,"uuid":"00002a06-0000-1000-8000
00805f9b34fb"}]
```

发现所有服务、特征和描述符:

AC调用:

```
http://{ac_ip}/api/gatt/nodes/<device_mac>/services/characteristics/descriptors?
mac=<gateway_mac>&access_token=<token>
```

本地调用:

```
http://{gateway_ip}/api/gatt/nodes/<device_mac>/services/characteristics/descrip
tors?
```

容器内调用:

```
http://10.10.10.254/api/gatt/nodes/<device_mac>/services/characteristics/descrip
tors?
```

返回值示例:

```
Response example:
Status-Line: HTTP/1.1 200 OK/r/n
Header: (general-header)
Message-body: text/plain
[{"uuid":"00001800-0000-1000-8000-00805f9b34fb","primary":true,"characteristics":[{"descriptors":[{"handle":3,"uuid":"00002a00-0000-1000-800000805f9b34fb"}],"handle":3,"properties":10,"uuid":"00002a00-0000-1000-8000-00805f9b34fb"}, {"descriptors":[{"handle":5,"uuid":"00002a01-00001000-8000-00805f9b34fb"}],"handle":5,"properties":2,"uuid":"00002a01-00001000-8000-00805f9b34fb"}, {"descriptors":[{"handle":7,"uuid":"00002a040000-1000-8000-00805f9b34fb"}],"handle":7,"properties":2,"uuid":"00002a040000-1000-8000-00805f9b34fb"}],"handle":1}]
```

更详细的使用方法，请参照在线RESTful-API文档。

<https://github.com/CassiaNetworks/CassiaSDKGuide/wiki/RESTful-API>

## 10、远程重启蓝牙网关

通过以下API，用户可以对蓝牙网关进行远程重启。

远程重启蓝牙网关接口为HTTP协议的GET方法。

AC调用：

`http://{ac-ip}/api/cassia/reboot?mac=<gateway-mac>&access_token=<token>`

本地调用：

`http://{gateway-ip}/cassia/reboot`

容器内调用：

`http://10.10.10.254/cassia/reboot`

## 11、通过蓝牙网关发送广播数据

通过以下API，用户可以控制蓝牙网关，向外发送广播包，用户可以自定义广播包格式和内容。

发送广播数据接口为HTTP协议中的GET方法。

开始广播数据：

AC调用：

`http://{ac-ip}/api/advertise/start?mac=<mac>&interval=<interval>&ad_data=<ad_data>&resp_data=<resp_data>&access_token=<token>`

本地调用：

`http://{gateway-ip}/advertise/start?interval=<interval>&ad_data=<ad_data>`

&resp\_data=<resp\_data>

容器内调用:

http://10.10.10.254/advertise/start?interval=<interval>&ad\_data=<ad\_data>  
&resp\_data=<resp\_data>

以下是参数:

参数	描述
interval	(必选): 以毫秒为单位的广播间隔。范围是20-10000毫秒。默认为500毫秒。
ad_type	(可选): 广播类型(见下表)。默认值3。
ad_data	(必选): 广播包内容, 数据类型为字符串。
resp_data	(必选): 扫描响应包。数据类型为字符串。当您要发送resp_data时, 请设置ad_type=0。

ad\_type内容:

Value	广播类型	注释
0	ADV_IND	可连接的无方向广播
1	ADV_DIRECT_IND	可连接定向广播
2	ADV_SCAN_IND	可扫描的非定向广播
3	ADV_NONCONN_IND	不可连接的无向广播
4	SCAN_RSP	扫描响应

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header : (general-header)
Message-body: text/plain

OK
```

停止发送广播数据:

AC调用:

http://{ac-ip}/api/advertise/stop?mac=<gateway\_mac>&access\_token=<token>

本地调用:

http://{gateway-ip}/advertise/stop

容器内调用:

http://10.10.10.254/advertise/stop

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header : (general-header)
Message-body: text/plain

OK
```

## 12、取 BLE 连接的 RSSI（v2.0 及更高版本）

用户可以通过此 API 获取指定蓝牙网关上连接的指定 BLE 终端设备，连接的当前 RSSI。

获取 BLE 连接 RSSI 接口为 HTTP 方法的 GET 方法。

AC 调用：

```
http://{ac-ip}/api/gap/nodes/<device-mac>/rssi?mac=<gateway-mac>
&access_token=<token>
```

本地调用：

```
http://{gateway-ip}/gap/nodes/<device-mac>/rssi?mac=<gateway-mac>
```

容器内调用：

```
http://10.10.10.254/gap/nodes/<device-mac>/rssi?mac=<gateway-mac>
```

返回值示例：

```
{"id": "C0:00:5B:D1:A9:20", "rssi": -33}
```

如果用户想要获得指定蓝牙网关连接的所有 BLE 终端设备的连续 RSSI 报告，可以使用以下 API。此接口为 SSE 接口，需要保持调用。

AC 调用：

```
http://{ac-ip}/api/gap/rssi?mac=<gateway-mac>&access_token=<token>
```

本地调用：

```
http://{gateway-ip}/gap/rssi?mac=<gateway-mac>
```

容器内调用：

```
http://10.10.10.254/gap/rssi?mac=<gateway-mac>
```

以下是参数：

参数	描述
rssi	（可选）：仅当 RSSI 低于此阈值时才报告 RSSI。默认值为 127。
rssi_interval	（可选）：RSSI 报告间隔。单位是毫秒。默认值为 1000。
filter_mac	（可选）：如果用户只想获取特定 BLE 终端设备的 RSSI 值，可以使用此参数。格式与根据设备 MAC、RSSI、NAME 和 UUID 过滤扫描数据中的参数 filter_mac 相同。

返回值示例：

```
{"id": "C0:00:5B:D1:A9:20", "rssi": -34}
```

## 六、蓝牙网关自动选择

Cassia AC可以从候选蓝牙网关列表中自动选择一个网关，然后使用该网关连接BLE终端设备。自动选择的蓝牙网关，基于BLE终端设备的RSSI、蓝牙网关当前负载和蓝牙网关是否支持该功能。

如果用户想要将BLE终端设备与指定蓝牙网关连接，或者他们想要使用自定义网关选择算法，请参照连接/断开连接BLE终端设备。

**注意：**此 API 仅Cassia AC 支持。

蓝牙网关自动选择接口为HTTP协议中的POST方法。

### 1、功能开启

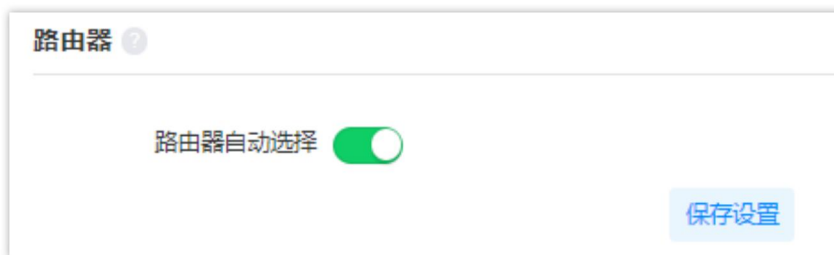
通过此API，用户可以启用或者禁用蓝牙网关自动选择功能。如果flag为1，则启用网关自动选择功能。如果flag为0，蓝牙网关自动选择功能将被关闭。

通过在调用该接口时指定scan\_params参数，用户可以根据设备MAC、RSSI、NAME、UUID过滤扫描数据。这是被动扫描，因此它只能过滤扫描到的BLE终端设备的adData。

scan\_params包括以下选项：

参数	描述
filter_mac	（可选）：BLE终端设备MAC列表。只有这些BLE终端设备的广播数据被发送到AC。
filter_rssi	（可选）：RSSI值。低于此RSSI值的BLE终端设备的广播包会被忽略。
其他	请参照扫描接口中的其他参数

**注意：**使用此 API 前，请现在AC的设置页面中，开启此开关。



调用示例：

`http://{ac-ip}/api/aps/ap-select-switch?access_token=<token>`

body参数示例1（application/json）：

仅开启功能

```
{"flag": 1}
```

body参数示例2（application/json）：

自定义筛选条件，指定BLE终端设备列表，和信号强度。

```
{
  "flag":1,
  "scan_params":{
    "filter_mac":[
      "C7:ED:44:24:3E:55",
      "C7:ED:44:24:3E:56",
      "C7:ED:44:24:3E:57"
    ],
    "filter_rssi":"-70"
  }
}
```

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/nHeader : (general-header)Message-body:
application/json

{ "status": "success", "flag": 1 }
```

## 2、连接BLE终端设备

调用此 API 将自动从指定的蓝牙网关列表中选择一个蓝牙网关，并使用它来连接 BLE 终端设备。

API URL:

[http://{ac-ip}/api/aps/connections/connect?access\\_token=<token>](http://{ac-ip}/api/aps/connections/connect?access_token=<token>)

body参数 (application/json) :

参数	描述
aps	自动选择的蓝牙网关列表，用户可以定义一个或者多个蓝牙网关的MAC。支持通配符*，表示使用AC上所有的蓝牙网关
devices	待连接的BLE终端设备MAC，只能填写一个。

body参数示例 (application/json) :

```
{
  "aps":[
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F9",
    "CC:1B:E0:E7:FE:FA"
  ],
  "devices": ["F7:18:BC:18:F0:3A"]
}
```

返回值示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header : (general-header)
Message-body: text/plain
OK
```

### 3、断连BLE终端设备

调用此 API 将断开连接的BLE终端设备。body中的“devices”中只能添加一个BLE终端设备的MAC地址。

API URL:

`http://{ac-ip}/api/aps/connections/disconnect?access_token=<token>`

body参数示例（application/json）：

```
{ "devices": ["F7:18:BC:18:F0:3A"] }
```

返回值示例：

```
Status-Line : HTTP/1.1 200 OK/r/n  
Header : (general-header)  
Message-body: text/plain
```

```
OK
```



## 七、组合SSE API

用户想要通过 AC使用RESTful API 控制蓝牙网关，可以通过以下两种方式：

为每个蓝牙网关创建三个SSE隧道：扫描数据、接收notify/indicate和监控连接的BLE终端设备状态变化。

通过AC 创建一个SSE隧道，即组合SSE。组合SSE可以接收该 AC 控制的所有蓝牙网关的扫描数据、接收notify/indicate和监控连接的BLE终端设备状态变化。

通过组合SSE API，可以简化在需要处理多个蓝牙网关数据时的操作，降低资源占用，在相同AC硬件资源条件下提升AC的可扩展性和处理能力。

**注意：**此 API 仅 Cassia AC 支持。

组合SSE接口为HTTP协议中的GET方法。

### 1、创建组合SSE

此 API 将创建一个与 AC 的组合 SSE 连接。此 SSE 连接可以接收此 AC 控制的所有网关的扫描数据、通知/指示数据和连接的设备状态。

API URL:

`http://{ac-ip}/api/aps/events?access_token=<token>`

调用此 API 时，AC 会立即返回一条包含所有网关信息的信息，例如：

返回值示例：

```
data: {
  "dataType": "state",
  "aps": {
    "CC:1B:E0:E7:FE:F8": {
      "_id": "5a93755b028e6c00519ce1dc",
      "id": "CC:1B:E0:E7:FE:F8",
      "mac": "CC:1B:E0:E7:FE:F8",
      "name": "Cassia Router",
      "group": "",
      "status": "online",
      "model": "X1000",
      "version": "1.2.0.1803131043",
      "position": "",
      "time": 1519613275655,
      "ip": "192.168.1.202",
      "localip": "192.168.1.202",
      "uptime": 14873,
      "offline_time": 0,
      "online_time": 1522052125,
      "update_status": "update_ok",
      "update_reason": "",
      "update_version": "",
      "update_progress": 0,
      "notify": true,
      "connectionstate": true
    },
  },
}
```

```

"CA:79:F5:B6:1F:04": {
  "devices": {
    "CA:79:F5:B6:1F:04": {
      "bdaddrs": {
        "bdaddr": "CA:79:F5:B6:1F:04",
        "bdaddrType": "random"
      },
      "chipId": 0,
      "handle": "",
      "name": "",
      "connectionState": "connected",
      "id": "CA:79:F5:B6:1F:04"
    }
  }
}
}
}
}

```

每 30 秒将返回一条keep-live消息，以确保 SSE 链接已启动并正在运行。

如果扫描开启（打开组合SSE的开启/关闭扫描接口），这个SSE 隧道将返回扫描数据。

返回扫描数据示例：

```

data:
{
  "dataType": "scan",
  "ap": "CC:1B:E0:E7:FE:F8",
  "bdaddrs": [
    {
      "bdaddr": "CC:1B:E0:E0:98:16",
      "bdaddrType": "public"
    }
  ],
  "adData": "0201060D084361737369615F5331303030",
  "name": "Cassia_S1000",
  "rssi": -34,
  "evt_type": 3
}

```

如果读取通知开关是开启的（默认），当任意蓝牙网关有notification/indication数据到达AC时，数据将通过此SSE 隧道返回给用户。

返回通知数据示例：

```

data:
{
  "dataType": "notification",
  "ap": "CC:1B:E0:E7:FE:F8",
  "value": "FF000C000205100101010126",
  "device": "CA:79:F5:B6:1F:04",
  "handle": 16
}

```

如果连接状态变化是开启的（默认），则当任何BLE终端设备的连接状态发生变化时，将通过此SSE隧道将BLE终端设备的连接状态变化返回给用户。

返回连接状态变化示例：

```
data:
{"handle":"CA:79:F5:B6:1F:04","connectionState":"disconnected","dataType":"connection_state","ap":"CC:1B:E0:E7:FE:F8"}data:
{"handle":"CA:79:F5:B6:1F:04","connectionState":"connected","dataType":"connection_state","ap":"CC:1B:E0:E7:FE:F8"}
```

如果蓝牙网关上下线状态是开启的（默认配置），当蓝牙网关的状态在在线和离线之间发生变化时，此SSE隧道将返回相应信息。

返回值示例：

```
{
  "dataType": "ap_state",
  "ap": "CC:1B:E0:E7:FE:F8",
  "mac": "CC:1B:E0:E7:FE:F8",
  "status": "offline",
  "offline_time": 1522067273296
}
```

## 2、开启/关闭扫描

### 开启扫描

通过此API将打开蓝牙网关列表中所有蓝牙网关的扫描。SSE隧道将接收并返回扫描数据。

开启扫描接口为HTTP协议中的POST方法。

API URL：

[http://{ac-ip}/api/aps/scan/open?access\\_token=<token>](http://{ac-ip}/api/aps/scan/open?access_token=<token>)

body示例（application/json）：

```
{
  "aps": [
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8"
  ],
  "chip": 0,
  "active": 0,
  "filter_name": "cassia"
}
```

body参数：

参数	描述
aps	(必选)：一个或多个蓝牙网关的MAC地址
chip	(可选)：0 或 1。表示扫描哪个芯片
active	(可选)：0 或 1。0 表示启用被动扫描；1 表示启用主动扫描
filter_name	(可选)：根据BLE终端设备的 name筛选
filter_mac	(可选)：根据BLE终端设备的 MAC筛选
filter_uuid	(可选)：根据BLE终端设备的 UUID筛选
filter_rssi	(可选)：根据BLE终端设备的 RSSI筛选
filter_duplicates	(可选)：过滤重复设备
filter_value	(可选)：根据广播数据筛选，并可配置数据位置偏移

返回数据示例：

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain
OK
```

### 关闭扫描

此 API 将关闭对网关列表中所有蓝牙网关的扫描。

关闭扫描接口为HTTP协议中的POST方法。

API URL:

[http://{ac-ip}/api/aps/scan/close?access\\_token=<token>](http://{ac-ip}/api/aps/scan/close?access_token=<token>)

body参数:

参数	描述
aps	一个或多个网关的 MAC 地址

body示例 (application/json) :

```
{
  "aps": [
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8"
  ]
}
```

返回数据示例：

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain

ok
```

### 3、开启/关闭通知

#### 开启通知

此 API 将在组合SSE隧道上的开启通知消息。BLE终端设备的notification和indication数据数据将通过此SSE返回给用户。

开启通知接口为HTTP协议中的POST方法。

#### API URL:

http://{ac-ip}/api/aps/notify/open?access\_token=<token>

#### body参数:

参数	描述
aps	一个或多个网关的 MAC 地址

#### body示例 (application/json) :

```
{
  "aps": [
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8"
  ]
}
```

#### 返回数据示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain

ok
```

#### 关闭通知

此 API 将在组合SSE隧道上的关闭通知消息。关闭后，BLE终端设备的notification和indication数据数据将不会再通过此SSE返回给用户。

关闭通知接口为HTTP协议中的POST方法。

#### API URL:

http://{ac-ip}/api/aps/notify/close?access\_token=<token>

#### body参数:

参数	描述
aps	一个或多个网关的 MAC 地址

body示例 (application/json) :

```
{
  "aps": [
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8"
  ]
}
```

返回数据示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain

ok
```

#### 4、开启/关闭连接状态变化报告

##### 开启连接状态变化报告

此 API 将开启组合 SSE 隧道上的连接状态变化监控。当BLE终端设备的连接状态发生变化时，连接状态变化数据将通过此SSE 隧道上返回给用户。

开启连接状态变化接口为HTTP协议中的POST方法。

API URL:

http://{ac-ip}/api/aps/connection-state/open?access\_token=<token>

body参数:

参数	描述
aps	一个或多个网关的 MAC 地址

body示例 (application/json) :

```
{
  "aps": [
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8"
  ]
}
```

返回数据示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain

ok
```

## 关闭连接状态变化报告

此 API 将关闭组合 SSE 隧道上的连接状态变化监控。关闭后，当 BLE 终端设备的连接状态发生变化时，连接状态变化数据将不会通过此 SSE 隧道上返回给用户。

关闭连接状态变化接口为 HTTP 协议中的 POST 方法。

API URL:

`http://{ac-ip}/api/aps/connection-state/close?access_token=<token>`

body 参数:

参数	描述
aps	一个或多个网关的 MAC 地址

body 示例 (application/json) :

```
{
  "aps": [
    "CC:1B:E0:E7:FE:F8",
    "CC:1B:E0:E7:FE:F8"
  ]
}
```

返回数据示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain

ok
```

## 5、开启/关闭蓝牙网关状态变化报告

### 打开蓝牙网关状态变化报告

此 API 将打开 SSE 隧道上所有网关的 ap-state 监控。当网关状态在在线和离线之间发生变化时，ap-state 的数据将在此 SSE 隧道上发送给用户应用程序。

开启蓝牙网关状态变化报告接口为 HTTP 协议中的 GET 方法。

API URL:

`http://{ac-ip}/api/aps/ap-state/open?access_token=<token>`

返回数据示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain

ok
```

### 关闭蓝牙网关状态变化报告

此 API 将关闭 SSE 隧道上所有网关的 **ap-state** 监控。**ap-state** 的数据将不再发送到此 SSE 隧道上的用户应用程序。

关闭蓝牙网关状态变化报告接口为HTTP协议中的GET方法。

`http://{ac-ip}/api/aps/ap-state/close?access_token=<token>`

返回数据示例:

```
Status-Line : HTTP/1.1 200 OK/r/n
Header      : (general-header)
Message-body : text/plain
```

```
ok
```



## 八、定位API

Cassia 支持基于房间的BLE终端设备位置跟踪。通过以下API可以获取到位置信息。

**注意：**在调用定位API接口之前，请先调用相关蓝牙网关的扫描接口。此 API 仅 Cassia AC 支持。

定位API接口为HTTP协议中的GET方法。

要识别 BLE终端 设备所在的最近网关：

```
http://{ac-ip}/api/middleware/position/by-device/<device_mac>
?access_token=<token>
```

该接口将返回 {"hubMac":"hubMac1"}，例如 {"hubMac":"CC:1B:E0:E0:01:47"}。

获取 AC 可以检测到的所有 BLE 终端设备的最近的蓝牙网关列表：

```
http://{ac-ip}/api/middleware/position/by-device/*?access_token=<token>
```

其中\*为通配符，代表所有所有BLE终端设备。

它将返回一个列表：

```
{
  "11: 22: 33: 44: 55: 66": {
    "hubMac": "CC: 1 B: E0: E0: 01: 47"
  },
  "11: 22: 33: 44: 55: 77": {
    "hubMac": "CC: 1 B: E0: E0: 01: 48"
  },
}
```

要获取蓝牙网关周围的 BLE终端设备列表：

```
http://{ac-ip}/api/middleware/position/by-ap/<gateway_mac>
?access_token=<token>
```

它将返回：

```
["11:22:33:44:55:66", "11:22:33:44:55:AA", ...].
```

获取 AC 内所有网关的 BLE 设备列表：

```
http://{ac-ip}/api/middleware/position/by-ap/*?access_token=<token>
```

其中\*为通配符，代表所有所有蓝牙网关。

它将返回：

```
{  
  "CC:1B:E0:E0:11:22": ["11:22:33:44:55:66", "11:22:33:44:55:AA"...],  
  ...  
}
```

## 九、蓝牙调试工具和在线文档

用户可以使用Cassia提供的蓝牙调试工具熟悉和学习Cassia RESTful API的使用，蓝牙调试工具的使用方法，请参照本地和云端的蓝牙网关使用说明中的相关章节。

蓝牙调试工具云端地址：

<http://bluetooth.tech/debugger2/dist/>

如果遇到云端工具无法使用的问题，请联系Cassia技术支持人员，或者索取windows版本的本地版工具。

完整的Cassia RESTful API文档，请访问以下地址查看。

<https://github.com/CassiaNetworks/CassiaSDKGuide/wiki/RESTful-API>