

# 上海城市轨道交通通讯系统

## PA/PIS 音视频统一信息服务系统 与综合监控系统

# 接口规格书

项目号	SP20P0015			
文档编号				
工程编号				
版本号	1.0			
保密级别	一般	<u>内部公开</u>	秘密	机密

上海地铁电子科技有限公司

# 上海城市轨道交通通讯系统

## PA/PIS 音视频统一信息服务系统 与综合监控系统

# 接口规格书

编制:	孙旭东	编制日期:	2020-12-02
审核:	_____	审核日期:	_____
批准:	_____	批准日期:	_____

上海地铁电子科技有限公司

## 修改记录

序号	版本	责任人	时间	修改说明
1、	0.1	孙旭东	2020.12.02	初稿
2、	0.2	孙旭东	2020.12.04	增加车站广播状态接口：章节 3.5.8~3.5.12.
3、				

## 审核记录

序号	版本	责任人	时间	修改说明
1、	1.0			审核
2、				
3、				
4、				
5、				

## 发放记录

序号	版本	责任人	时间	修改说明
1、	1.0			文件发布
2、				
3、				
4、				
5、				

# 目录

<b>1 引言</b> .....	<b>6</b>
1.1 RESTFUL 介绍 .....	6
<b>2 PA-PIS 系统与综合监控系统通讯协议</b> .....	<b>6</b>
2.1 概述 .....	6
2.2 协议约定 .....	6
2.3 JWT 鉴权 .....	6
2.3.1 申请 appid .....	7
2.3.2 获取 token .....	8
2.3.3 接口 API 调用说明 .....	9
<b>3 接口规范</b> .....	<b>11</b>
3.1 排版约定 .....	11
3.2 编码 .....	11
3.3 消息格式 .....	11
3.4 状态码 .....	11
3.4.1 1xx 状态码 .....	12
3.4.2 2xx 状态码 .....	12
3.4.3 3xx 状态码 .....	12
3.4.4 4xx 状态码 .....	13
3.4.5 5xx 状态码 .....	14
3.5 中央通讯接口 .....	14
3.5.1 获取中央广播控制终端（呼叫站）列表 .....	14
3.5.2 获取中央广播节点列表 .....	15
3.5.3 获取中央广播分区列表 .....	16
3.5.4 发起中央呼叫站广播 .....	18
3.5.5 撤销中央呼叫站广播 .....	19

---

3.5.6 发起中央呼叫站监听.....	20
3.5.7 撤销中央呼叫站监听.....	21
3.5.8 获取车站广播分区列表.....	22
3.5.9 获取车站广播线路状态.....	25
3.5.10 获取车站广播功放列表.....	27
3.5.11 获取车站广播控制终端（呼叫站）列表.....	28
3.5.12 获取车站广播其他设备状态列表（含控制器）.....	30
<b>4 配置表.....</b>	<b>31</b>
4.1 中央席位.....	31

# 1 引言

## 1.1 Restful介绍

Representational State Transfer(REST)是一种架构原则，其中将 web 服务视为资源，可以由 URL 唯一标识。RESTful Web 服务的关键特点是明确使用 HTTP 方法来表示不同的操作的调用。REST 的基本设计原则对典型 CRUD 操作使用 HTTP 协议方法：

POST – 创建资源

GET – 检索资源

PUT – 更新资源

DELETE – 删除资源

REST 服务的主要优势在于：它们是跨平台高度可重用的，应为它们都依赖基本 HTTP 协议。它们使用基本的 XML/JSON，而不是复杂的 SOAP XML，使用非常方便。

## 2 PA-PIS系统与综合监控系统通讯协议

### 2.1 概述

PA-PIS 音视频统一信息服务系统与综合监控系统间的通信采用 restful 通信协议框架，PA-PIS 音视频统一信息服务系统作为 restful 服务器，综合监控系统作为 restful 客户端。

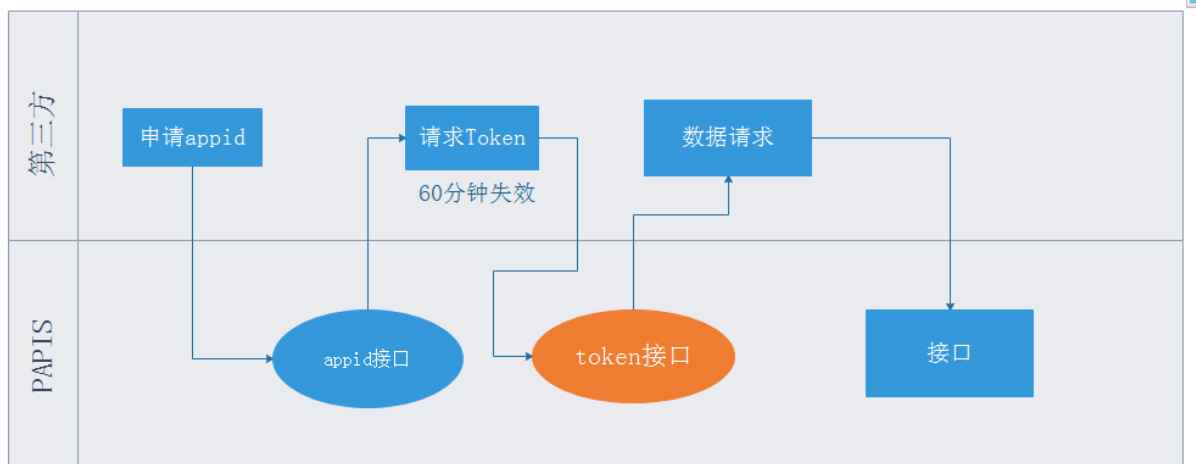
### 2.2 协议约定

中央或每个站点综合监控可以访问的 IP 为两个，状态获取采用轮询，轮询频率不小于 1 秒。授权方式采用 JWT 鉴权。

### 2.3 JWT鉴权

JSON Web Token (JWT)是一个开放标准(RFC 7519)，它定义了一种紧凑的、自包含的方式，用于作为 JSON 对象在各方之间安全地传输信息。该信息可以被验证和信任，因为它是数字签名的。

鉴权流程:



第三方根据平台提供的 appid 和与约定的时间序列获取接口通讯 Token, 此 Token 一小时失效, 第三方跟平台通讯过程中需要提交 Token, 如果 Token 错误则无权限访问接口, Token 获取限制最多每秒一次。

## 2.3.1 申请appid

每个第三方向平台申请, 平台通过后即可使用, 第三方申请的 appid 如在 15 天内未使用 (没有接口访问记录), 则需要再次申请。备注: 上线后的用户名和密码需另行约定。

### 2.3.1.1 请求API

```
POST /api/user/regappid
```

### 2.3.1.2 请求参数描述

1. authName -string 用户名
2. userNum -int 用户编号

### 2.3.1.3 请求参数示例

```
{
  "authName": "sios",
  "userNum": 1234
}
```

### 2.3.1.4 回应参数描述

1. success - boolean 是否成功
2. code - int code 200 成功 500 错误
3. message - string 失败理由
4. regappid - string 授权 appid

### 2.3.1.5 回应参数示例

```
成功
{
  "success": true,
  "code": 200,
  "message": "",
  "regappid": "ef08e274512c41d885a755edc7870e9d"
}

失败:
{
  "success": false,
  "code": 500,
  "message": "用户不存在,请检查用户名和用户编号",
  "regappid": ""
}
```

## 2.3.2 获取token

每次发送请求，请求 Headers 的 AccessTokenThird 值应为授权时返回的 token，token 有效期为 1 小时。

### 2.3.2.1 请求参数描述

```
POST /api/system/gettoken
```

### 2.3.2.2 请求参数描述

1. appid -string 授权的 appid
2. timeKey - int 时间戳（Unix 时间戳，如 2019/11/19 17:36:01 转化为时间戳等于 1574156161）



### 2.3.2.3 请求参数示例

```
{
  "appid": "2494969c3478409ca7717c2113173b1f",
  "timeKey": 1574156161
}
```

### 2.3.2.4 回应参数描述

1. success - boolean 是否成功
2. code - int code 200 成功 500 系统报错 401 未授权/授权失效
3. message - string 失败理由
4. token - string 授权 token

### 2.3.2.5 回应参数示例

成功:

```
{
  "success": true,
  "code": 200,
  "message": "",
  "token": "ThirdBearer xxx.xxx.xxx"
}
```

失败:

```
{
  "success": false,
  "code": 500,
  "message": "xxx",
  "token": ""
}
```

## 2.3.3 接口API调用说明

通用接口请求头 Headers 包含 AccessTokenThird

1. AccessTokenThird - string

### 2.3.3.1 接口通用回应参数描述

1. success - boolean 是否成功

2. code - string 返回编码 200 成功 500 失败 401 未授权
3. message - string 返回错误信息
4. timestamp - long 当前时间
5. result - 内容为章节 3.5、章节 3.6 各接口回应参数

### 2.3.3.2 章节 3.5.1 返回成功示例：

```
{
  "success":true,
  "code":200,
  "message":"",
  "timestamp":1577347166546,
  "result":[
    {
      "id": 2001,
      "name": "行调 1",
      "status": "on"
    },
    {
      "id": 2002,
      "name": "行调 2",
      "status": "on"
    },
    {
      "id": 2003,
      "name": "行调 3",
      "status": "on"
    },
    {
      "id": 2004,
      "name": "总调",
      "status": "on"
    },
    ...
  ]
}
```

### 2.3.3.3 章节 3.5.1 返回失败示例：

```
{
  "success": false,
```

```
"code": 500,  
"message": "errorMessage",  
"timestamp": 1577347166546,  
"result": null  
}
```

## 3 接口规范

### 3.1 排版约定

排版格式	含义
<>	变量
[]	可选项
{}	必选项
	互斥关系
等宽字体 Courier New	屏幕输出

### 3.2 编码

文本编码统一采用 UTF-8 编码。

### 3.3 消息格式

返回或发送的消息体为 json 格式，即 Text/Json。

### 3.4 状态码

HTTP 状态码就是一个三位数，分成以下五个类别。

- 1xx: 相关信息
- 2xx: 操作成功
- 3xx: 重定向
- 4xx: 客户端错误
- 5xx: 服务器错误

### 3.4.1 1xx状态码

服务器收到请求，需要请求者继续执行操作。

状态码	状态码英文名称	中文描述
100	Continue	继续。客户端应继续其请求。
101	Switching Protocols	切换协议。服务器根据客户端的请求切换协议。只能切换到更高级的协议，例如，切换到 HTTP 的新版本协议。
102	Processing	由 WebDAV (RFC 2518) 扩展的状态码，代表处理将被继续执行。

### 3.4.2 2xx状态码

操作被成功接收并处理，但是不同的方法可以返回更精确的状态码。

状态码	状态码英文名称	中文描述
200	OK	请求成功。
201	Created	已创建。成功请求并创建了新的资源。
202	Accepted	已接受。已经接受请求，但未处理完成。
203	Non-Authoritative Information	非授权信息。请求成功。但返回的 meta 信息不在原始的服务器，而是一个副本。
204	No Content	无内容。服务器成功处理，但未返回内容。在未更新网页的情况下，可确保浏览器继续显示当前文档。
205	Reset Content	重置内容。服务器处理成功，用户终端（例如：浏览器）应重置文档视图。可通过此返回码清除浏览器的表单域。
206	Partial Content	部分内容。服务器成功处理了部分请求。
207	Multi-Status	由 WebDAV (RFC 2518) 扩展的状态码，代表之后的消息体将是一个 XML 消息，并且可能依照之前子请求数量的不同，包含一系列独立的响应代码。

### 3.4.3 3xx状态码

需要进一步的操作以完成请求。

状态码	状态码英文名称	中文描述
300	Multiple Choices	多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如：浏览器）选择。
301	Moved Permanently	永久移动。请求的资源已被永久的移动到新 URI，返回信息会包括新的 URI，浏览器会自动定向到新 URI。今后任何新的请求都应使用新的 URI 代替。
302	Move temporarily	临时移动。与 301 类似。但资源只是临时被移动。客户端应继续使用原有 URI。
303	See Other	查看其它地址。与 301 类似。使用 GET 和 POST 请求查看。

304	Not Modified	未修改。所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。客户端通常会缓存访问过的资源，通过提供一个头信息指出客户端希望只返回在指定日期之后修改的资源。
305	Use Proxy	使用代理。所请求的资源必须通过代理访问。
306	Unused	已经被废弃的 HTTP 状态码。
307	Temporary Redirect	临时重定向。与 302 类似。使用 GET 请求重定向。

### 3.4.4 4xx状态码

客户端错误，请求包含语法错误或无法完成请求。

状态码	状态码英文名称	中文描述
400	Bad Request	客户端请求的语法错误，服务器无法理解。
401	Unauthorized	请求要求用户的身份认证。
402	Payment Required	保留，将来使用。
403	Forbidden	服务器理解请求客户端的请求，但是拒绝执行此请求。
404	Not Found	服务器无法根据客户端的请求找到资源（网页）。通过此代码，网站设计人员可设置“您所请求的资源无法找到”的个性页面。
405	Method Not Allowed	客户端请求中的方法被禁止。
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求。
407	Proxy Authentication Required	请求要求代理的身份认证，与 401 类似，但请求者应当使用代理进行授权。
408	Request Time-out	服务器等待客户端发送的请求时间过长，超时。
409	Conflict	服务器完成客户端的 PUT 请求是可能返回此代码，服务器处理请求时发生了冲突。
410	Gone	客户端请求的资源已经不存在。410 不同于 404，如果资源以前有现在被永久删除了可使用 410 代码，网站设计人员可通过 301 代码指定资源的新位置。
411	Length Required	服务器无法处理客户端发送的不带 Content-Length 的请求信息。
412	Precondition Failed	客户端请求信息的先决条件错误。
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个 Retry-After 的响应信息。
414	Request-URI Too Large	请求的 URI 过长（URI 通常为网址），服务器无法处理。

415	Unsupported Media Type	服务器无法处理请求附带的媒体格式。
416	Requested range not satisfiable	客户端请求的范围无效。
417	Expectation Failed	服务器无法满足 Expect 的请求头信息。
422	Unprocessable Entity	客户端上传的附件无法处理，导致请求失败。
429	Too Many Requests	客户端的请求次数超过限额。

### 3.4.5 5xx状态码

服务器在处理请求的过程中发生了错误。

状态码	状态码英文名称	中文描述
500	Internal Server Error	服务器内部错误，无法完成请求。
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
502	Bad Gateway	充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。
503	Service Unavailable	由于超载或系统维护，服务器暂时的无法处理客户端的请求。延时的长度可包含在服务器的 Retry-After 头信息中。
504	Gateway Time-out	充当网关或代理的服务器，未及时从远端服务器获取请求。
505	HTTP Version not supported	服务器不支持请求的 HTTP 协议的版本，无法完成处理。

## 3.5 中央通讯接口

### 3.5.1 获取中央广播控制终端（呼叫站）列表

#### 3.5.1.1 请求API

```
GET /api/pa/call_stations
```

#### 3.5.1.2 请求参数描述

无

### 3.5.1.3 回应参数描述

```
array[struct]
  1. id - id@primary(int) 呼叫站 id
  2. name - string 终端名
  3. status - enum(string) 终端是否工作正常
      1. on - 正常
      2. failure - 故障
```

### 3.5.1.4 回应参数示例

```
[
  {
    "id": 5051,
    "name": "行调 1",
    "status": "on"
  },
  {
    "id": 5052,
    "name": "行调 2",
    "status": "on"
  },
  {
    "id": 5053,
    "name": "行调 3",
    "status": "on"
  },
  {
    "id": 5054,
    "name": "综合调 1",
    "status": "on"
  }
  ...
]
```

## 3.5.2 获取中央广播节点列表

车站、段场、停车库等都是独立的子系统，作为不同的节点。

### 3.5.2.1 请求API

```
GET /api/pa/subsystems
```

### 3.5.2.2 请求参数描述

无

### 3.5.2.3 回应参数描述

```
array[struct]
  1. id - id@primary 节点 id
  2. name - string 节点名
```

### 3.5.2.4 回应参数示例

```
[
  {
    "id": 1501,
    "name": "莘庄站"
  },
  {
    "id": 1502,
    "name": "春申路站"
  }
  ...
]
```

## 3.5.3 获取中央广播分区列表

### 3.5.3.1 请求API

```
GET /api/pa/zones
```

### 3.5.3.2 请求参数描述

无



### 3.5.3.3 回应参数描述

```
array[struct]
  1. id - id@primary 分区 id
  2. name_en - string 分区名(英文)
  3. name_zh - string 分区名(中文)
```

### 3.5.3.4 回应参数示例

```
[
  {
    "id": 1,
    "name_en": "up",
    "name_zh": "奉贤上行"
  },
  {
    "id": 2,
    "name_en": "down",
    "name_zh": "奉贤至莘庄下行"
  },
  {
    "id": 3,
    "name_en": "up2",
    "name_zh": "闵开上行"
  },
  {
    "id": 4,
    "name_en": "down2",
    "name_zh": "闵开至莘庄下行"
  },
  {
    "id": 5,
    "name_en": "hall",
    "name_zh": "站厅"
  },
  {
    "id": 6,
    "name_en": "exit",
    "name_zh": "车站出入口"
  },
  {
    "id": 7,
```

```
"name_en": "office",  
"name_zh": "设备办公区"  
}  
...  
]
```

### 3.5.4 发起中央呼叫站广播

#### 3.5.4.1 请求API

```
POST /api/pa/start_mic
```

#### 3.5.4.2 请求参数描述

1. call\_station\_id - id 呼叫站 id
2. subsystem\_ids - array[id](int) 所选节点(车站等)
3. zone\_ids - array[id](int) 所选分区

#### 3.5.4.3 请求参数示例

```
{  
  "call_station_id": 5051,  
  "subsystem_ids": [1,2,3,4,5...],  
  "zone_ids": [1,2,3...]  
}
```

#### 3.5.4.4 回应参数描述

1. result - array[struct] 所选节点(车站等)设置反馈
  1. id - int 所选节点(车站等)id
  2. result - bool 调用是否成功
    1. true - 成功
    2. false - 失败
  3. reason - string 失败理由

#### 3.5.4.5 回应参数示例

```
[  
  {  
    "id": 1,
```

```
"result": true,
"reason": "航头站广播呼叫成功。"
},
{
  "id": 2,
  "result": true,
  "reason": "鹤立西路站广播呼叫成功。"
},
{
  "id": 3,
  "result": false,
  "reason": "下盐路站广播呼叫失败。"
},
{
  "id": 4,
  "result": true,
  "reason": "沈梅路站广播呼叫失败。"
},
{
  "id": 5,
  "result": true,
  "reason": "繁荣路站广播呼叫失败。"
}
...
]
```

### 3.5.5 撤销中央呼叫站广播

#### 3.5.5.1 请求API

```
DELETE /api/pa/stop_mic
```

#### 3.5.5.2 请求参数描述

```
1. call_station_id - id 呼叫站 id
```

#### 3.5.5.3 请求参数示例

```
{
  "call_station_id": 5051
}
```

```
}
```

#### 3.5.5.4 回应参数描述

1. result - bool 调用是否成功
  1. true - 成功
  2. false - 失败
2. reason - string 失败理由

#### 3.5.5.5 回应参数示例

```
{  
  "result": true,  
  "reason": "撤销呼叫站广播成功。"  
}
```

### 3.5.6 发起中央呼叫站监听

#### 3.5.6.1 请求API

```
POST /api/pa/start_monitor
```

#### 3.5.6.2 请求参数描述

1. call\_station\_id - id 呼叫站 id
2. subsystem\_id - id 所选节点(车站等)
3. zone\_id - id 所选分区

#### 3.5.6.3 请求参数示例

```
{  
  "call_station_id": 5051,  
  "subsystem_id": 1,  
  "zone_id": 2  
}
```

#### 3.5.6.4 回应参数描述

1. result - bool 调用是否成功
  1. true - 成功

- 2. false - 失败
- 2. reason - string 失败理由

### 3.5.6.5 回应参数示例

```
{  
  "result": true,  
  "reason": "呼叫站监听成功。"  
}
```

## 3.5.7 撤销中央呼叫站监听

### 3.5.7.1 请求API

```
DELETE /api/pa/stop_monitor
```

### 3.5.7.2 请求参数描述

- 1. call\_station\_id - id 呼叫站 id

### 3.5.7.3 请求参数示例

```
{  
  "call_station_id": 5051  
}
```

### 3.5.7.4 回应参数描述

- 1. result - bool 调用是否成功
  - 1. true - 成功
  - 2. false - 失败
- 2. reason - string 失败理由

### 3.5.7.5 回应参数示例

```
{  
  "result": true,  
  "reason": "撤销呼叫站监听成功。"  
}
```

### 3.5.8 获取车站广播分区列表

#### 3.5.8.1 请求API

```
GET /api/pa/all_zones
```

#### 3.5.8.2 请求参数描述

无

#### 3.5.8.3 回应参数描述

```
array[struct]
  1. station_id - station_id@primary 车站节点 id
  2. zones - array[struct] 广播分区列表
    1. id - id@primary 分区 id
    2. name_en - string 分区名(英文)
    3. name_zh - string 分区名(中文)
    4. status - enum(string) 分区是否被占用
      0. free - 未被占用
      1. ats - ATS 自动广播
      2. prerecord - 预录语音广播
      3. call - 人工广播
      4. fire - 消防广播
      5. bgm - 背景音乐广播
      6. tts - TTS 广播
      7. other - 其他广播
    5. caller - enum(string) 呼叫者
      0. "" - 当分区状态为非人工广播时, 为空
      1. sios - 车站值班员
      2. other_pa - 换乘站其他线路广播系统
      3. lte - 无线手持台
      4. train_dispatcher1 - 行调 1
      5. train_dispatcher2 - 行调 2
      6. train_dispatcher3 - 行调 3
      8. multiple_dispatcher1 - 综合调 1
    6. volume - int 音量 (1~100)
```

### 3.5.8.4 回应参数示例

```
[
  {
    "station_id": 1501,
    "zones": [
      {
        "id": 1,
        "name_en": "up",
        "name_zh": "上行站台",
        "status": "free",
        "caller": "",
        "volume": 60
      },
      {
        "id": 2,
        "name_en": "down",
        "name_zh": "下行站台",
        "status": "ats",
        "caller": "",
        "volume": 60
      },
      {
        "id": 3,
        "name_en": "hall",
        "name_zh": "站厅",
        "status": "call",
        "caller": "sios",
        "volume": 60
      },
      {
        "id": 4,
        "name_en": "office",
        "name_zh": "设备办公区",
        "status": "call",
        "caller": "sios",
        "volume": 60
      },
      {
        "id": 5,
        "name_en": "exit",
        "name_zh": "站车站出入口厅",

```

```
        "status": "call",
        "caller": "sios",
        "volume": 50
    }
    ...
]
},
{
    "station_id": 1502,
    "zones": [
        {
            "id": 1,
            "name_en": "up",
            "name_zh": "上行站台",
            "status": "free",
            "caller": "",
            "volume": 60
        },
        {
            "id": 2,
            "name_en": "down",
            "name_zh": "下行站台",
            "status": "ats",
            "caller": "",
            "volume": 60
        },
        {
            "id": 3,
            "name_en": "hall",
            "name_zh": "站厅",
            "status": "call",
            "caller": "sios",
            "volume": 60
        },
        {
            "id": 4,
            "name_en": "office",
            "name_zh": "设备办公区",
            "status": "call",
            "caller": "sios",
            "volume": 60
        }
    ],
    {
```



```
[
  {
    "id": 5,
    "name_en": "exit",
    "name_zh": "站车站出入口厅",
    "status": "call",
    "caller": "sios",
    "volume": 50
  },
  {
    "id": 6,
    "name_en": "transfer",
    "name_zh": "换乘通道",
    "status": "call",
    "caller": "sios",
    "volume": 60
  },
  ...
]
},
...
]
```

### 3.5.9 获取车站广播线路状态

#### 3.5.9.1 请求API

```
GET /api/pa/all_broadcast_lines
```

#### 3.5.9.2 请求参数描述

无

#### 3.5.9.3 回应参数描述

```
array[struct]
  1. station_id - station_id@primary 车站节点 id
  2. broadcast_lines - array[struct] 广播线路状态
    1. name - string 广播线路名
    2. status - enum(string) 广播线路状态
      1. normal - 正常
```

2. short - 短路
3. open - 断路

#### 3.5.9.4 回应参数示例

```
[
  {
    "station_id": 1501,
    "broadcast_lines": [
      {
        "name": "上行站台 1",
        "status": "normal"
      },
      {
        "name": "上行站台 2",
        "status": "normal"
      },
      {
        "name": "下行站台 1",
        "status": "normal"
      },
      ...
    ]
  },
  {
    "station_id": 1502,
    "broadcast_lines": [
      {
        "name": "上行站台 1",
        "status": "normal"
      },
      {
        "name": "上行站台 2",
        "status": "normal"
      },
      {
        "name": "下行站台 1",
        "status": "normal"
      },
      ...
    ]
  },
]
```

```
...  
]
```

### 3.5.10 获取车站广播功放列表

#### 3.5.10.1 请求API

```
GET /api/pa/all_amplifiers
```

#### 3.5.10.2 请求参数描述

无

#### 3.5.10.3 回应参数描述

```
array[struct]  
  1. station_id - station_id@primary 车站节点 id  
  2. amplifiers - array[struct] 功放列表  
    1. id - id@primary 功放 id  
    2. name - string 功放名  
    3. status - enum 功放状态  
      1. on - 正常  
      2. off - 关机  
      3. failure - 故障
```

#### 3.5.10.4 回应参数示例

```
[  
  {  
    "station_id": 1501,  
    "amplifiers": [  
      {  
        "id": 10001,  
        "name": "上行站台 1",  
        "status": "on"  
      },  
      {  
        "id": 10002,  
        "name": "上行站台 2",
```

```
        "status": "on"
      },
      {
        "id": 10003,
        "name": "下行站台 1",
        "status": "on"
      }
    ]
  },
  {
    "station_id": 1502,
    "amplifiers": [
      {
        "id": 10001,
        "name": "上行站台 1",
        "status": "on"
      },
      {
        "id": 10002,
        "name": "上行站台 2",
        "status": "off"
      },
      {
        "id": 10003,
        "name": "下行站台 1",
        "status": "on"
      }
    ]
  }
  ...
]
],
...
]
```

### 3.5.11 获取车站广播控制终端（呼叫站）列表

#### 3.5.11.1 请求API

```
GET /api/pa/all_call_stations
```

### 3.5.11.2 请求参数描述

无

### 3.5.11.3 回应参数描述

array[struct]

1. station\_id - station\_id@primary 车站节点 id
2. call\_stations - array[struct] 车站广播控制终端（呼叫站）列表
  1. id - id@primary(int) 呼叫站 id
  2. name - string 终端名
  3. status - enum(string) 终端是否工作正常
    1. on - 正常
    2. failure - 故障

### 3.5.11.4 回应参数示例

```
[
  {
    "station_id": 1501,
    "call_stations": [
      {
        "id": 1539,
        "name": "东川路站呼叫站",
        "status": "on"
      }
    ]
  },
  {
    "station_id": 1502,
    "call_stations": [
      {
        "id": 1540,
        "name": "江川路站呼叫站",
        "status": "on"
      }
    ]
  },
  ...
]
```

### 3.5.12 获取车站广播其他设备状态列表（含控制器）

#### 3.5.12.1 请求API

```
GET /api/pa/all_child_devices
```

#### 3.5.12.2 请求参数描述

无

#### 3.5.12.3 回应参数描述

```
array[struct]
  1. station_id - station_id@primary 车站节点 id
  2. child_devices - array[struct] 其他设备状态列表（含控制器）
    1. name - string 设备名
    2. status - enum(string) 设备是否工作正常
      1. on - 正常
      2. failure - 故障
```

#### 3.5.12.4 回应参数示例

```
[
  {
    "station_id": 1501,
    "child_devices": [
      {
        "name": "东川路站主机",
        "status": "on"
      }
    ]
  },
  {
    "station_id": 1502,
    "child_devices": [
      {
        "name": "江川路站主机",
```

```
        "status": "on"
      }
    ]
  },
  ...
]
```

## 4 配置表

### 4.1 中央席位

编号	席位名 (英文)	席位名 (中文)
1	train_dispatcher1	行调 1
2	train_dispatcher2	行调 2
3	train_dispatcher3	行调 3
4	multiple_dispatcher1	综合调 1