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 | SP19P0039 | | | |
| 文档编号 |  | | | |
| 工程编号 |  | | | |
| 版 本 号 | 1.0 | | | |
| 保密级别 | 一般 | **内部公开** | 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： | 2019-11-08 |
| 审核： | 王凯 | 审核日期： | 2020-07-28 |
| 批准： | 贺志龙 | 批准日期： | 2020-07-28 |

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** |
|  | 0.1 | 孙旭东 | 2019.11.08 | 初稿 |
|  | 0.2 | 孙旭东 | 2019.11.18 | 增加JWT鉴权、发布车厢拥堵信息 |
|  | 0.3 | 孙旭东 | 2019.12.03 | 中央接口：增加发布文本信息、撤销文本信息  车站接口：增加开关分区屏幕、修改发布车厢拥堵信息参数、修改发起预录音、背景音乐广播参数type的类型 |
|  | 0.4 | 孙旭东 | 2019.12.12 | 车站接口：修改获取广播分区列表中的回应参数caller的值域、调整广播音量值域（1～100）  配置表：增加中央席位、14号线车站、18号线车站 |
|  | 0.5 | 孙旭东 | 2019.12.24 | 中央接口：增加发布车厢拥堵信息 |
|  | 0.6 | 孙旭东 | 2019.12.27 | 章节2.3 JWT鉴权说明修改  中央接口：修改发布车厢拥堵信息参数（TODO） |
|  | 0.7 | 孙旭东 | 2020.01.03 | 中央接口：修改发布车厢拥堵信息 |
|  | 0.8 | 孙旭东 | 2020.01.02 | 车站接口：删除车厢拥堵信息 |
|  | 0.9 | 孙旭东 | 2020.02.26 | 中央接口：修改发布车厢拥堵信息的参数说明及示例 |
|  | 0.91 | 孙旭东 | 2020.02.28 | 以下接口的优先级删除（PAPIS系统判定）  车站接口：  发起预录音、背景音乐广播  发起合成语音广播  发布文本信息(滚动)  撤销文本信息(滚动)  发布重要公告  撤销重要公告 |
|  | 0.92 | 孙旭东 | 2020.03.18 | 车站接口：PAPIS系统自检的请求API路径修改 |
|  | 0.93 | 孙旭东 | 2020.05.06 | 车站接口：发布车厢拥堵信息增加整车列车温度信息 |
|  | 0.94 | 孙旭东 | 2020.05.09 | 车站接口：发布车厢拥堵信息中整车列车温度信息改为每节车厢温度 |
|  | 0.95 | 孙旭东 | 2020.06.16 | 车站接口：获取广播分区列表中增加TTS广播、  发布重要公告的参数“轮播次数”删除 |
|  | 0.96 | 孙旭东 | 2020.07.08 | 车站接口：获取PIS分区状态和获取通道LED屏幕状态列表的监视url增加备注：url连接无效 |
|  | 0.97 | 孙旭东 | 2020.07.28 | 14号线车站名变更 |
|  | 1.01 | 孙旭东 | 2020.11.04 | 获取广播分区列表中呼叫者“总调”改为“调度长” |
|  |  |  |  |  |

审核记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **版本** | **责任人** | **时间** | **修改说明** |
|  | 1.0 | 王凯 | 2020.07.28 | 审核 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

发放记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **版本** | **责任人** | **时间** | **修改说明** |
|  | 1.0 | 贺志龙 | 2020.07.28 | 文件发布 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 引言 8](#_Toc47511585)

[1.1 Restful介绍 8](#_Toc47511586)

[2 PA-PIS系统与综合监控系统通讯协议 8](#_Toc47511587)

[2.1 概述 8](#_Toc47511588)

[2.2 协议约定 8](#_Toc47511589)

[2.3 JWT鉴权 8](#_Toc47511590)

[2.3.1 申请appid 9](#_Toc47511591)

[2.3.2 获取token 10](#_Toc47511592)

[2.3.3 接口API调用说明 11](#_Toc47511593)

[3 接口规范 14](#_Toc47511594)

[3.1 排版约定 14](#_Toc47511595)

[3.2 编码 14](#_Toc47511596)

[3.3 消息格式 14](#_Toc47511597)

[3.4 状态码 14](#_Toc47511598)

[3.4.1 1xx状态码 15](#_Toc47511599)

[3.4.2 2xx状态码 15](#_Toc47511600)

[3.4.3 3xx状态码 15](#_Toc47511601)

[3.4.4 4xx状态码 16](#_Toc47511602)

[3.4.5 5xx状态码 17](#_Toc47511603)

[3.5 中央通讯接口 17](#_Toc47511604)

[3.5.1 获取广播控制终端（呼叫站）列表 17](#_Toc47511605)

[3.5.2 获取广播节点列表 18](#_Toc47511606)

[3.5.3 获取广播分区列表 19](#_Toc47511607)

[3.5.4 发起呼叫站广播 21](#_Toc47511608)

[3.5.5 撤销呼叫站广播 22](#_Toc47511609)

[3.5.6 发起呼叫站监听 23](#_Toc47511610)

[3.5.7 撤销呼叫站监听 24](#_Toc47511611)

[3.5.8 发布文本信息 25](#_Toc47511612)

[3.5.9 撤销文本信息 26](#_Toc47511613)

[3.5.10 发布车厢拥堵信息 27](#_Toc47511614)

[3.6 车站通讯接口 29](#_Toc47511615)

[3.6.1 获取广播分区列表 29](#_Toc47511616)

[3.6.2 获取广播线路状态 31](#_Toc47511617)

[3.6.3 获取广播功放列表 32](#_Toc47511618)

[3.6.4 获取广播控制终端（呼叫站）列表 33](#_Toc47511619)

[3.6.5 获取广播其他设备状态列表（含控制器） 34](#_Toc47511620)

[3.6.6 获取广播预录语音、背景音乐广播文件 34](#_Toc47511621)

[3.6.7 发起呼叫站广播 35](#_Toc47511622)

[3.6.8 撤销呼叫站广播 37](#_Toc47511623)

[3.6.9 发起呼叫站监听 37](#_Toc47511624)

[3.6.10 撤销呼叫站监听 38](#_Toc47511625)

[3.6.11 发起预录音、背景音乐广播 39](#_Toc47511626)

[3.6.12 撤销预录音、背景音乐广播 40](#_Toc47511627)

[3.6.13 发起合成语音广播 41](#_Toc47511628)

[3.6.14 撤销合成语音广播 43](#_Toc47511629)

[3.6.15 修改分区音量 43](#_Toc47511630)

[3.6.16 获取PIS分区列表 44](#_Toc47511631)

[3.6.17 获取PIS分区状态 45](#_Toc47511632)

[3.6.18 获取LCD屏幕列表 47](#_Toc47511633)

[3.6.19 发布文本信息(滚动) 48](#_Toc47511634)

[3.6.20 撤销文本信息(滚动) 49](#_Toc47511635)

[3.6.21 发布重要公告 50](#_Toc47511636)

[3.6.22 撤销重要公告 51](#_Toc47511637)

[3.6.23 开关分区屏幕 52](#_Toc47511638)

[3.6.24 调整LCD屏幕音量 53](#_Toc47511639)

[3.6.25 获取通道LED屏幕状态列表 54](#_Toc47511640)

[3.6.26 通道LED工况切换 55](#_Toc47511641)

[3.6.27 发布重要文本信息（插播） 57](#_Toc47511642)

[3.6.28 发布普通文本信息（并播） 58](#_Toc47511643)

[3.6.29 状态恢复（含工况恢复、撤销文本信息） 59](#_Toc47511644)

[3.6.30 调整LED屏幕亮度 60](#_Toc47511645)

[3.6.31 PAPIS系统自检 61](#_Toc47511646)

[4 配置表 61](#_Toc47511647)

[4.1 工况类型 61](#_Toc47511648)

[4.2 中央席位 62](#_Toc47511649)

[4.3 14号线车站 62](#_Toc47511650)

[4.4 18号线车站 63](#_Toc47511651)

# 引言

## 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RUL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。

# PA-PIS系统与综合监控系统通讯协议

## 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。

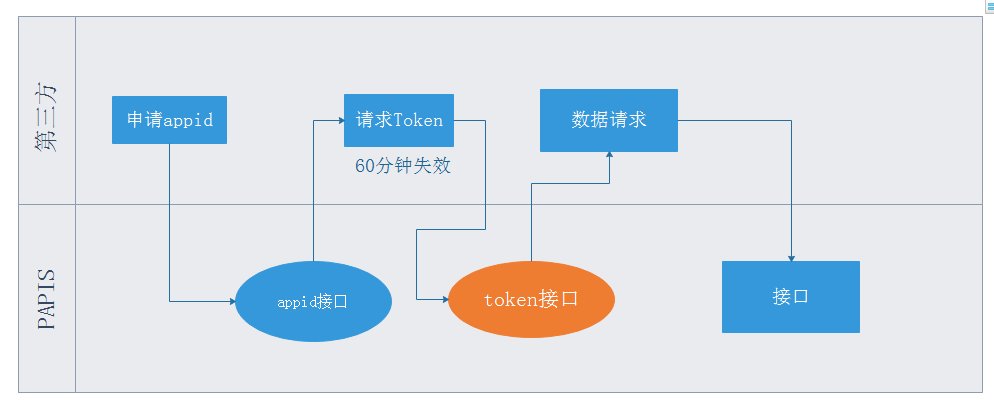
## 协议约定

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。

## 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。

### 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。

#### 请求API

POST /api/user/regappid

#### 请求参数描述

      1. authName –string  用户名

      2. userNum -int 用户编号

#### 请求参数示例

{

   "authName": "sios",

   "userNum": 1234

}

#### 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

#### 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": ""

}

### 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。

#### 请求参数描述

POST /api/system/gettoken

#### 请求参数描述

      1. appid –string  授权的appid

      2. timeKey - int 时间戳（Uinx时间戳，如2019/11/19 17:36:01转化为时间戳等于1574156161）

#### 请求参数示例

{

"appid": "2494969c3478409ca7717c2113173b1f",

"timeKey": 1574156161

}

#### 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

#### 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": ""

}

### 接口API调用说明

通用接口请求头Headers包含AccessTokenThird

   1. AccessTokenThird - string

#### 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

#### 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"

},

...

]

}

#### 章节3.5.1返回失败示例：

{

"success": false,

"code": 500,

"message":"errorMessage",

"timestamp":1577347166546,

"result":null

}

#### 章节3.6.7返回成功示例：

{

"success":true,

"code":200,

"message":"",

"timestamp":1577347166546,

"result":[

{

"id": 1,

"result": true,

"reason": "广播呼叫成功。"

},

{

"id": 2,

"result": true,

"reason": "广播呼叫成功。"

},

{

"id": 3,

"result": false,

"reason": "分区被高优先级占用。"

}

...

]

}

#### 章节3.6.7返回失败示例：

{

"success": false,

"code": 500,

"message":"errorMessage",

"timestamp":1577347166546,

"result":null

}

# 接口规范

## 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 |

## 编码

文本编码统一采用UTF-8编码。

## 消息格式

返回或发送的消息体为json格式，即Text/Json。

## 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

### 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。 |

### 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。 |

### 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。 |

### 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。 |

### 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。 |

## 中央通讯接口

### 获取广播控制终端（呼叫站）列表

#### 请求API

GET /api/pa/call\_stations

#### 请求参数描述

无

#### 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

#### 回应参数示例

[

   {

      "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": "调度长",

      "status": "on"

   }

   ...

]

### 获取广播节点列表

车站、段场、停车库等都是独立的子系统，作为不同的节点。

#### 请求API

GET /api/pa/subsystems

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   1. id - id@primary 节点id

   2. name - string 节点名

#### 回应参数示例

[

   {

      "id": 1,

      "name": "航头站"

   },

   {

      "id": 2,

      "name": "鹤立西路站"

   }

   ...

]

### 获取广播分区列表

#### 请求API

GET /api/pa/zones

#### 请求参数描述

无

#### 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)

#### 回应参数示例

[

   {

      "id": 1,

      "name\_en": "up",

      "name\_zh": "上行站台"

   },

   {

      "id": 2,

      "name\_en": "down",

      "name\_zh": "下行站台"

   },

   {

      "id": 3,

      "name\_en": "hall",

      "name\_zh": "站厅"

   },

   {

      "id": 4,

      "name\_en": "office",

      "name\_zh": "设备办公区"

   },

   {

      "id": 5,

      "name\_en": "exit",

      "name\_zh": "站车站出入口厅"

   },

   {

      "id": 6,

      "name\_en": "transfer",

      "name\_zh": "换乘通道"

   }

   ...

]

### 发起呼叫站广播

#### 请求API

POST /api/pa/start\_mic

#### 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

#### 请求参数示例

{

   "call\_station\_id": 2001,

   "subsystem\_ids": [1,2,3,4,5...],

   "zone\_ids": [1,2,3...]

}

#### 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

#### 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。"

   }

   ...

]

### 撤销呼叫站广播

#### 请求API

DELETE /api/pa/stop\_mic

#### 请求参数描述

      1. call\_station\_id - id 呼叫站id

#### 请求参数示例

{

   "call\_station\_id": 2001

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "撤销呼叫站广播成功。"

}

### 发起呼叫站监听

#### 请求API

POST /api/pa/start\_monitor

#### 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

#### 请求参数示例

{

   "call\_station\_id": 2001,

   "subsystem\_id": 1,

   "zone\_id": 2

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "呼叫站监听成功。"

}

### 撤销呼叫站监听

#### 请求API

DELETE /api/pa/stop\_monitor

#### 请求参数描述

      1. call\_station\_id - id 呼叫站id

#### 请求参数示例

{

   "call\_station\_id": 2001

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "撤销呼叫站监听成功。"

}

### 发布文本信息

#### 请求API

POST /api/pis/post\_text

#### 请求参数描述

      1. user\_id – int 席位编号

      2. subsystem\_ids - array[id](int) 所选节点(车站等)

      3. zone\_ids - array[int] 所选分区

      4. message - string 文本

      5. times - int 轮播次数 - 0为无限轮播 （命令类型为滚动时有效）

#### 请求参数示例

{

   "user\_id": 1,

   "subsystem\_ids": [1,2,3,4,5...],

   "zone\_ids": [1,2],

   "message": "欢迎乘坐上海地铁18号线。",

   "times": 0

}

#### 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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "播放器未开启。"

   }

]

### 撤销文本信息

#### 请求API

DELETE /api/pis/cancel\_text

#### 请求参数描述

      1. user\_id – int 席位编号

      2. subsystem\_ids - array[id](int) 所选节点(车站等)

      3. zone\_ids - array[int] 所选分区

#### 请求参数示例

{

   "user\_id": 1,

   "subsystem\_ids": [1,2,3,4,5...],

   "zone\_ids": [1,2]

}

#### 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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "播放器未开启。"

   }

]

### 发布车厢拥堵信息

#### 请求API

POST /api/pis/post\_jam\_infos

#### 请求参数描述

      1. jam\_infos - array[struct] 车厢拥堵信息

         1. train\_id - int 列车号(需要和ATS的列车号匹配，或者提供对应关系)

         2. direction – enum(string) 列车方向

            1. up – 上行

            2. down – 下行

         3. speed - int 速度（km/h，0：静止）

         4. jam\_tube - array[enum(string)] 车厢拥堵信息（车厢编组按顺序固定）

            1. free – 空载

            2. comfortable - 舒适

            3. normal - 轻度拥挤

            4. jam - 重度拥挤

         5. weight - array[int] 车厢载重（单位：KG，车厢编组按顺序固定）

         6. temperature - array[int] 列车车厢温度（单位：℃，车厢编组按顺序固定）

#### 请求参数示例

[

   {

      "train\_id": 1801,

      "direction": "up",

      "speed": 0,

      "jam\_tube": ["free","jam","jam","normal","normal","comfortable"]，

      "weight": [0,8888,9999,6666,5555,2356],

      "temperature": [20,22,23,23,22,21]

   },

   {

      "train\_id": 1802,

      "direction": "up",

      "speed": 32,

      "jam\_tube": ["free","free","comfortable","comfortable","free","free"],

      "weight": [0,0,2356,2255,0,0],

      "temperature": [21,22,23,24,22,21]

   },

...

]

#### 回应参数描述

      1. result - array[struct] 发布拥堵信息(列车等)设置反馈

         1. train\_id - int 所发布拥堵信息((列车等)id

         2. result - bool 发布是否成功

            1. true - 成功

            2. false - 失败

         3. reason - string 失败理由

#### 回应参数示例

[

   {

      "id": 1801,

      "result": true,

      "reason": "列车拥堵信息发布成功。"

   },

   {

      "id": 1802,

      "result": true,

      "reason": 列车拥堵信息发布成功。"

   },

   ...

]

## 车站通讯接口

### 获取广播分区列表

#### 请求API

GET /api/pa/zones

#### 请求参数描述

无

#### 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)

   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 20200616增加

      7. other - 其他广播

   5. caller - enum(string) 呼叫者（待定，20191110增加）

      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

      7. train\_dispatcher4 - 行调4（增加20191212）

      8. multiple\_dispatcher1 - 综合调1（增加20191212）

      9. multiple\_dispatcher2 - 综合调2（增加20191212）

      10. chief\_dispatcher - 调度长（修改20201104）

   6. volume - int 音量（0～100）

#### 回应参数示例

[

   {

      "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

   }

   ...

]

### 获取广播线路状态

#### 请求API

GET /api/pa/broadcast\_lines

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   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

#### 回应参数示例

[

   {

      "name": "上行站台1",

      "status": "normal"

   },

   {

      "name": "上行站台2",

      "status": "normal"

   },

   {

      "name": "下行站台1",

      "status": "normal"

   }

   ...

]

### 获取广播功放列表

#### 请求API

GET /api/pa/amplifiers

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   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

#### 回应参数示例

[

   {

      "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"

   }

   ...

]

### 获取广播控制终端（呼叫站）列表

#### 请求API

GET /api/pa/call\_stations

#### 请求参数描述

无

#### 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

#### 回应参数示例

[

   {

      "id": 1069,

      "name": "大世界站车控室",

      "status": "on"

   }

]

### 获取广播其他设备状态列表（含控制器）

#### 请求API

GET /api/pa/child\_devices

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   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

#### 回应参数示例

[

   {

      "name": "大世界站主机",

      "status": "on"

   }

]

### 获取广播预录语音、背景音乐广播文件

#### 请求API

GET /api/pa/audio\_files

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   1. id - id@primary(int) 文件id

   2. name - string 文件名

   3. text - string 描述

#### 回应参数示例

[

   {

      "id": 1,

      "name": "富锦路终点站.mp3",

      "text": "富锦路终点站"

   },

   {

      "id": 2,

      "name": "火车站终点站.mp3",

      "text": "火车站终点站"

   },

   {

      "id": 3,

      "name": "莘庄终点站.mp3",

      "text": "莘庄终点站"

   }

   ...

]

### 发起呼叫站广播

#### 请求API

POST /api/pa/start\_mic

#### 请求参数描述

      1. call\_station\_id - id(int) 呼叫站id

      2. zone\_ids - array[id] 所选分区

#### 请求参数示例

{

   "call\_station\_id": 1069,

   "zone\_ids": [1,2,3]

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "广播呼叫成功。"

   },

   {

      "id": 2,

      "result": true,

      "reason": "广播呼叫成功。"

   },

   {

      "id": 3,

      "result": false,

      "reason": "分区被高优先级占用。"

   }

]

### 撤销呼叫站广播

#### 请求API

DELETE /api/pa/stop\_mic

#### 请求参数描述

      1. call\_station\_id - id(int) 呼叫站id

#### 请求参数示例

{

   "call\_station\_id": 1069

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "撤销呼叫站广播成功。"

}

### 发起呼叫站监听

#### 请求API

POST /api/pa/start\_monitor

#### 请求参数描述

      1. call\_station\_id - id(int) 呼叫站id

      2. zone\_id - id 所选分区

#### 请求参数示例

{

   "call\_station\_id": 1069,

   "zone\_id": 2

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "呼叫站监听成功。"

}

### 撤销呼叫站监听

#### 请求API

DELETE /api/pa/stop\_monitor

#### 请求参数描述

      1. call\_station\_id - id 呼叫站id

#### 请求参数示例

{

   "call\_station\_id": 1069

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "撤销呼叫站监听成功。"

}

### 发起预录音、背景音乐广播

#### 请求API

POST /api/pa/start\_prerecord

#### 请求参数描述

      1. audio\_file\_id - id(int) 预录音文件编号id

~~2. priority - int 优先级~~

      3. times - int 播放次数 - 0为无限轮播

      4. zone\_ids - array[id] 所选分区

      5. type -enum(string) 广播音源类型 20191204修正

         1. pre - 预录语音

         2. bgm - 背景音乐

         3. ats\_up - ATS上行

         4. ats\_down - ATS下行

#### 请求参数示例

{

   "audio\_file\_id": 3,

~~"priority": 100,~~

   "times": 3,

   "zone\_ids": [1,2,3],

   "type": "pre"

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

      2. id - id 发起的id - 与“撤销的id”相同，用于标识预录语音的发起者

#### 回应参数示例

{

   "result": [

      {

         "id": 1,

         "result": true,

         "reason": "发起广播成功。"

      },

      {

         "id": 2,

         "result": true,

         "reason": "发起广播成功。"

      },

      {

         "id": 3,

         "result": false,

         "reason": "分区被高优先级占用。"

      }

   ],

   "id": 10001

}

### 撤销预录音、背景音乐广播

#### 请求API

DELETE /api/pa/stop\_prerecord

#### 请求参数描述

      1. id - id(int) 撤销的id - 与“发起的id”相同，用于标识预录语音的撤销者

#### 请求参数示例

{

   "id": 10001

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "撤销广播成功。"

}

### 发起合成语音广播

#### 请求API

POST /api/pa/start\_tts

#### 请求参数描述

      1. message - string 语音合成的文字（赛嘉可以考虑）

~~2. priority - int 优先级~~

      3. times - int 播放次数 - 0为无限轮播

      4. interval - int 播放间隔(秒)

      5. zone\_ids - array[id] 所选分区

#### 请求参数示例

{

   "message": "",

~~"priority": 102,~~

   "times": 2,

   "interval": 10,

   "zone\_ids": [1,2,3]

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

      2. id - id 发起的id - 与“撤销的id”相同，用于标识合成语音的发起者

#### 回应参数示例

{

   "result": [

      {

         "id": 1,

         "result": true,

         "reason": "发起合成语音广播成功。"

      },

      {

         "id": 2,

         "result": true,

         "reason": "发起合成语音广播成功。"

      },

      {

         "id": 3,

         "result": false,

         "reason": "分区被高优先级占用。"

      }

   ],

   "id": 10002

}

### 撤销合成语音广播

#### 请求API

DELETE /api/pa/stop\_tts

#### 请求参数描述

      1. id - id 撤销的id - 与“发起的id”相同，用于标识合成语音的撤销者

#### 请求参数示例

{

   "id": 10002

}

#### 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

#### 回应参数示例

{

   "result": true,

   "reason": "撤销合成语音广播成功。"

}

### 修改分区音量

#### 请求API

PUT /api/pa/adjust\_volume

#### 请求参数描述

       1. zone\_id - id(int) 分区id

       2. volume - int 分区音量(1~100)

#### 请求参数示例

{

   "zone\_id": 2,

   "volume": 60

}

#### 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 失败的理由

#### 回应参数示例

{

   "result": true,

   "reason": "修改分区音量成功。"

}

### 获取PIS分区列表

#### 请求API

GET /api/pis/zones

#### 请求参数描述

无

#### 回应参数描述

   1. id - id@primary(int) 分区id

   2. name\_en - string 分区名（英文）

   3. name\_zh - string 分区名（中文）

#### 回应参数示例

array[struct]

[

   {

      "id": 1,

      "name\_en": "up",

      "name\_zh": "上行站台"

   },

   {

      "id": 2,

      "name\_en": "down",

      "name\_zh": "下行站台"

   }

]

### 获取PIS分区状态

#### 请求API

GET /api/pis/statues

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   1. id - id@primary(int) 分区id

   2. mode - enum(string) 当前工况（针对不同自定义版式）

   3. roll\_text - string 滚动文本

   4. important\_text - string 重要公告

   5. volume\_type - bool 是否静音状态

      1.  true  静音

      2.  false 非静音

   6. volume\_value - int 音量（1～100）

   7. monitor\_url - string 监视url（备注：url连接无效）

   8. monitor\_type - enum(string) 监控类型

      1. photo 图片

      2. video 视频

      3. text 文本

   9. disk\_status - bool 硬盘属性状态

      1.  true 正常

      2.  false 故障

   10. sum\_space - int 硬盘总空间（单位MB）

   11. free\_space - int 硬盘剩余空间（单位MB）

   12. temperature - int 温度

   13. sum\_memory – int（单位MB）内存总空间

   14. free\_memory – int（单位MB） 内存剩余空间

#### 回应参数示例

[

   {

      "id": 1,

      "mode": "default",

      "roll\_text": "欢迎乘坐上海地铁18号线。"

      "important\_text": "因客流较大，目前18号线杨高中路站暂时停止运营。"

      "volume\_type": false,

      "volume\_value": 55,

      "monitor\_url": "rtsp://10.18.1.12/sc",

      "monitor\_type": "video",

      "disk\_status": true,

      "sum\_space": 512000,

      "free\_space": 307200,

      "temperature": 52,

      "sum\_memory": 16384,

      "free\_memory": 10240

   }

]

### 获取LCD屏幕列表

#### 请求API

GET /api/pis/screens

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   1. id - id@primary(int) 屏幕id（屏幕序号,从1开始编号）

   2. zone\_id - id@foreign(int) 分区id

   3. name - string 屏幕名

   4. status - enum(string) 屏幕设备状态

      1. on 开启

      2. off 关闭

      3. failure 故障

#### 回应参数示例

[

   {

      "id": 1,

      "zone\_id": 1,

      "name": "上行屏幕1",

      "status": "on"

   },

   {

      "id": 2,

      "zone\_id": 1,

      "name": "上行屏幕2",

      "status": "off"

   },

   {

      "id": 3,

      "zone\_id": "1",

      "name": "上行屏幕3",

      "status": "failure"

   }

]

### 发布文本信息(滚动)

#### 请求API

POST /api/pis/post\_text

#### 请求参数描述

      1. zone\_ids - array[int] 所选分区

~~2. level - int 消息等级（等值越高，权限越大，1-10级）~~

      3. message - string 文本

      4. times - int 轮播次数 - 0为无限轮播 （命令类型为滚动时有效）

#### 请求参数示例

{

   "zone\_ids": [1,2],

~~"level": 3,~~

   "message": "欢迎乘坐上海地铁18号线。",

   "times": 0

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "分区播放器未开启。"

   }

]

### 撤销文本信息(滚动)

#### 请求API

DELETE /api/pis/cancel\_text

#### 请求参数描述

      1. zone\_ids - array[int] 所选分区

~~2. level - int 消息等级（等值越高，权限越大，1-10级）~~

#### 请求参数示例

{

   "zone\_ids": [1,2],

~~"level": 6~~

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "分区播放器未开启。"

   }

]

### 发布重要公告

#### 请求API

POST /api/pis/post\_notice

#### 请求参数描述

      1. zone\_ids - array[int] 所选分区

~~2. level - int 消息等级（等值越高，权限越大，1-10级）~~

      3. message - string 文本

~~4. times - int 轮播次数 - 0为无限轮播 （命令类型为滚动时有效）~~

#### 请求参数示例

{

   "zone\_ids": [1,2],

~~"level": 5,~~

   "message": "因客流较大，目前18号线杨高中路站暂时停止运营。"

~~"times": 5~~

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "分区播放器未开启。"

   }

]

### 撤销重要公告

#### 请求API

DELETE /api/pis/cancel\_notice

#### 请求参数描述

      1. zone\_ids - array[int] 所选分区

~~2. level - int 消息等级（等值越高，权限越大，1-10级）~~

#### 请求参数示例

{

   "zone\_ids": [1,2],

~~"level": 5~~

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "分区播放器未开启。"

   }

]

### 开关分区屏幕

#### 请求API

POST /api/pis/switch\_screens

#### 请求参数描述

      1. zone\_ids - array[int] 所选分区

      2. cmd\_type - bool 命令类型

         1. true  打开

         2. false 关闭

#### 请求参数示例

{

   "zone\_ids": [1,2],

   "cmd\_type": true

}

#### 回应参数描述

      1. result - array[struct] 分区设置反馈

         1. id - int 分区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

回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "分区播放器未开启。"

   }

]

### 调整LCD屏幕音量

#### 请求API

PUT /api/pis/adjust\_volume

#### 请求参数描述

      1. zone\_id - id(int) 所选分区

      2. cmd\_type - enum(string) 命令类型

         1. mute 静音

         2. recover 音量恢复

         3. adjust 音量调整

      3. volume - int 音量（0～100）

#### 请求参数示例

{

   "zone\_id": 1,

   "cmd\_type": "adjust",

   "volume": 60

}

#### 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 失败的理由

#### 回应参数示例

回应参数示例：

{

   "result": true,

   "reason": "调整屏幕音量成功。"

}

### 获取通道LED屏幕状态列表

#### 请求API

GET /api/tunnel\_led/statues

#### 请求参数描述

无

#### 回应参数描述

array[struct]

   1. id - int@primary 屏id

   2. name\_zh - string 屏中文名

   3. mode - enum(string) 当前工况

   4. monitor\_url - string 监视url（备注：url连接无效）

   5. monitor\_type - enum(string) 监控类型

      1. photo 图片

      2. video 视频

   6. brightness - int 屏幕亮度（0～100）

   7. player\_status - enum(string) 播放器状态

      1. online 在线

      2. offline 离线

#### 回应参数示例

[

   {

      "id": 1,

      "name\_zh": "18换9通道口正面",

      "mode": "normal",

      "monitor\_url": "http://10.18.1.43:8080/TerminalInfoService.do?type=screensnapshot&mac=02-12-04-41-C1-DC",

      "monitor\_type": "photo",

      "brightness": 30,

      "player\_status": "online"

   },

   {

      "id": 2,

      "name\_zh": "18换9通道口背面",

      "mode": "normal",

      "monitor\_url": "http://10.18.1.43:8080/TerminalInfoService.do?type=screensnapshot&mac=02-12-04-41-C1-AA",

      "monitor\_type": "photo",

      "brightness": 30,

      "player\_status": "online"

   }

]

### 通道LED工况切换

#### 请求API

POST /api/tunnel\_led/switch\_mode

#### 请求参数描述

      1. ids - array[int] 所选屏幕

      2. mode - enum(string) 工况

         参考表《工况类型》

#### 请求参数示例

{

   "ids": [1,2],

   "mode": "urgent"

}

#### 回应参数描述

      1. result - array[struct] 屏幕设置反馈

         1. id - int 屏幕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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false

      "reason": "播放器不在线。"

   }

]

### 发布重要文本信息（插播）

#### 请求API

POST /api/tunnel\_led/insert\_text

#### 请求参数描述

      1. ids - array[int] 所选屏幕

      2. message - string 文本

#### 请求参数示例

{

   "ids": [1,2],

   "message": "因客流较大，目前18号线杨高中路站暂时停止运营。"

}

#### 回应参数描述

      1. result - array[struct] 屏幕设置反馈

         1. id - int 屏幕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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "播放器未开启。"

   }

]

### 发布普通文本信息（并播）

#### 请求API

POST /api/tunnel\_led/post\_text

#### 请求参数描述

      1. ids - array[int] 所选屏幕

      2. message - string 文本

#### 请求参数示例

{

   "ids": [1,2],

   "message": "欢迎乘坐上海地铁18号线。"

}

#### 回应参数描述

      1. result - array[struct] 屏幕设置反馈

         1. id - int 屏幕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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "播放器未开启。"

   }

]

### 状态恢复（含工况恢复、撤销文本信息）

#### 请求API

POST /api/tunnel\_led/cancel

#### 请求参数描述

      1. ids - array[int] 所选屏幕

#### 请求参数示例

{

   "ids": [1,2]

}

#### 回应参数描述

      1. result - array[struct] 屏幕设置反馈

         1. id - int 屏幕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

#### 回应参数示例

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "播放器未开启。"

   }

]

### 调整LED屏幕亮度

#### 请求API

POST /api/tunnel\_led/adjust\_bright

#### 请求参数描述

      1. ids - array[int] 所选屏幕

      2. brightness - int 屏幕亮度（0～100）

#### 请求参数示例

{

   "ids": [1,2],

   "brightness": 35

}

#### 回应参数描述

      1. result - array[struct] 屏幕设置反馈

         1. id - int 屏幕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

#### 回应参数示例

[

   {

      "id": 1,

      "result": true,

      "reason": "操作成功。"

   },

   {

      "id": 2,

      "result": false,

      "reason": "屏幕未开启。"

   }

]

### PAPIS系统自检

#### 请求API

POST /api/papis/self\_check

#### 请求参数描述

无

#### 回应参数描述

      1. result - bool 自检状态

         1. true – 自检成功

         2. false – 自检失败

#### 回应参数示例

{

   "result": true

}

# 配置表

## 工况类型

|  |  |  |
| --- | --- | --- |
| **工况id** | **工况类型（英文）** | **工况类型（中文）** |
| 1 | default | 正常（默认） |
| 2 | morning | 早间启运 |
| 3 | normal | 平峰 |
| 4 | morning\_peak | 早高峰 |
| 5 | evening\_peak | 晚高峰 |
| 6 | evening | 晚间停运 |
| 7 | flow3 | 三级大客流 |
| 8 | flow2 | 二级大客流 |
| 9 | flow1 | 一级大客流 |
| 10 | urgent | 紧急 |
| 11 | other | 其他 |

## 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 | train\_dispatcher4 | 行调4 |
| 5 | multiple\_dispatcher1 | 综合调1 |
| 6 | multiple\_dispatcher2 | 综合调2 |
| 7 | chief\_dispatcher | 调度长 |

## 14号线车站

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **车站名（英文）** | **车站名（中文）** | **服务端地址** |
| 0 | occ | 控制中心 | 10.14.202.10 |
| 1 | Fengbang | 封浜站 | 10.14.101.10 |
| 2 | Lexiu Road | 乐秀路站 | 10.14.102.10 |
| 3 | Lintao Road | 临洮路站 | 10.14.103.10 |
| 4 | Jiayi Road | 嘉怡路站 | 10.14.104.10 |
| 5 | Dingbian Road | 定边路站 | 10.14.105.10 |
| 6 | Zhenxin Xincun | 真新新村站 | 10.14.106.10 |
| 7 | Zhen'guang Road | 真光路站 | 10.14.107.10 |
| 8 | Tongchuan Road | 铜川路站 | 10.14.108.10 |
| 9 | Zhenru | 真如站 | 10.14.109.10 |
| 10 | Zhongning Road | 中宁路站 | 10.14.110.10 |
| 11 | Caoyang Road | 曹杨路站 | 10.14.111.10 |
| 12 | Wuning Road | 武宁路站 | 10.14.112.10 |
| 13 | Wuding Road | 武定路站 | 10.14.113.10 |
| 14 | Jing'an Temple | 静安寺站 | 10.14.114.10 |
| 15 | South Huangpi Road | 黄陂南路站 | 10.14.115.10 |
| 16 | Dashijie | 大世界站 | 10.14.116.10 |
| 17 | Yu Garden Road | 豫园站 | 10.14.117.10 |
| 18 | Lujiazui | 陆家嘴站 | 10.14.118.10 |
| 19 | South Pudong Road | 浦东南路站 | 10.14.119.10 |
| 20 | Pudong Avenue | 浦东大道站 | 10.14.120.10 |
| 21 | Yuanshen Road | 源深路站 | 10.14.121.10 |
| 22 | Changyi Road | 昌邑路站 | 10.14.122.10 |
| 23 | Xiepu Road | 歇浦路站 | 10.14.123.10 |
| 24 | Longju Road | 龙居路站 | 10.14.124.10 |
| 25 | Yunshan Road | 云山路站 | 10.14.125.10 |
| 26 | Lantian Road | 蓝天路站 | 10.14.126.10 |
| 27 | Huangyang Road | 黄杨路站 | 10.14.127.10 |
| 28 | Yunshun Road | 云顺路站 | 10.14.128.10 |
| 29 | Pudong Football Stadium | 浦东足球场站 | 10.14.129.10 |
| 30 | Jinyue Road | 金粤路站 | 10.14.130.10 |
| 31 | Guiqiao Road | 桂桥路站 | 10.14.131.10 |
| 99 | - | 封浜车辆段 | 10.14.199.10 |
| 98 | - | 金桥停车场 | 10.14.198.10 |

备注：端口号为8086

## 18号线车站

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **车站名（英文）** | **车站名（中文）** | **服务端地址** |
| 0 | occ | 控制中心 | 10.18.202.10 |
| 1 | Hangtou | 航头站 | 10.18.101.10 |
| 2 | Xiasha | 下沙站 | 10.18.102.10 |
| 3 | Hetao Road | 鹤涛路站 | 10.18.103.10 |
| 4 | Shenmei Road | 沈梅路站 | 10.18.104.10 |
| 5 | Fanrong Road | 繁荣路站 | 10.18.105.10 |
| 6 | Zhoupu | 周浦站 | 10.18.106.10 |
| 7 | Kangqiao | 康桥站 | 10.18.107.10 |
| 8 | Yuqiao | 御桥站 | 10.18.108.10 |
| 9 | Lianxi Road | 莲溪路站 | 10.18.109.10 |
| 10 | Beizhong Road | 北中路站 | 10.18.110.10 |
| 11 | Fangxin Road | 芳芯路站 | 10.18.111.10 |
| 12 | Longyang Road | 龙阳路站 | 10.18.112.10 |
| 13 | Yingchun Road | 迎春路站 | 10.18.113.10 |
| 14 | Middle Yanggao Road | 杨高中路站 | 10.18.114.10 |
| 15 | Minsheng Road | 民生路站 | 10.18.115.10 |
| 16 | Changyi Road | 昌邑路站 | 10.18.116.10 |
| 17 | Danyang Road | 丹阳路站 | 10.18.117.10 |
| 18 | Pingliang Road | 平凉路站 | 10.18.118.10 |
| 19 | Jiangpu Park | 江浦公园站 | 10.18.119.10 |
| 20 | Jiangpu Road | 江浦路站 | 10.18.120.10 |
| 21 | Fushun Road | 抚顺路站 | 10.18.121.10 |
| 22 | Guoquan Road | 国权路站 | 10.18.122.10 |
| 23 | Fudan University | 复旦大学站 | 10.18.123.10 |
| 24 | Shanghai University of Finance and Economics | 上海财经大学站 | 10.18.124.10 |
| 25 | Yingao Road | 殷高路站 | 10.18.125.10 |
| 26 | South Changjiang Road | 长江南路站 | 10.18.126.10 |
| 99 | - | 航头定修段 | 10.18.199.10 |

备注：端口号为8086