**版本号 V0.3**

# 一、接口定义

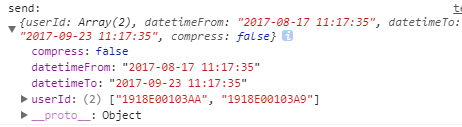
## 查询历史轨迹

Send: POST

url: /lbs/get\_history\_location

"userId": ["1918E00103AA", "1918E00103A9"], // "userId": ["all"] for all users  
"datetimeFrom" : "2017-08-17 11:17:35", // 按照时间段查询  
"datetimeTo" : "2017-09-23 11:17:35",  
"compress" : **false,** // 是否压缩，压缩的含义是，服务器会根据特定策略返回部分路径坐标。保留

"page": 1, // 查询的页码。 当记录很多时，需要分页查询。可选参数，默认为第一页  
"rows": 50 // 当前页记录条数。可选参数。默认100条



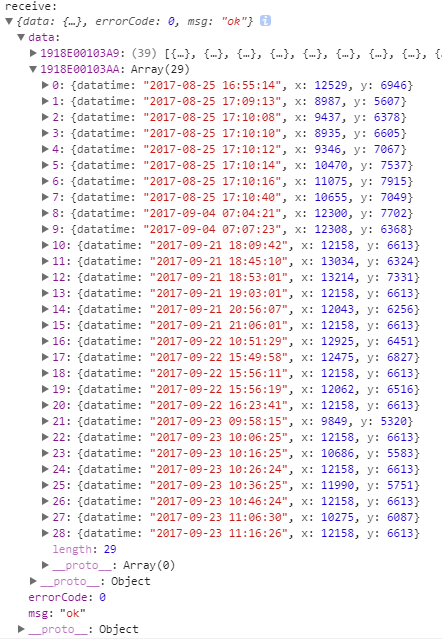
Ajax调用示例：

/\*\*  
 \* 向服务器发送 Ajax 请求，查询相关数据, POST方式  
 \* **@param** url 请求 url  
 \* **@param** txData 请求 参数，具体参数含义，见对应接口说明  
 \*/**function** ajaxRequest(url, txData) {  
 $.ajax({  
 url: url,  
 async: **true**, // 异步请求  
 type: 'POST',  
 data: {data: JSON.stringify(txData)}

}).done(**function**(data) {

console.log('接口调用成功');  
 }).fail(**function**(jqXHR, textStatus, errorThrown) {  
 console.log('接口调用失败');  
 });  
}

Receive:



返回参数增加 total。返回参数data->datatime改名为datetime。(2017-12-12)

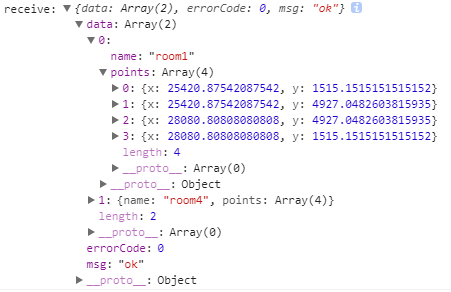
{  
 errorCode 错误码  
 0 成功  
 msg 错误信息  
 'ok' 成功  
 data[{ 历史轨迹记录。排序方式 userId， datetime。如果查询多个用户，无法控制每个用户  
 的记录条数，他们的总条数等于 输入参数 rows  
 x 横坐标，实际物理坐标，单位 mm  
 y 纵坐标  
 datetime 时间格式（北京时间）： yyyy-mm-dd HH-MM-SS  
 }]  
 total 符合条件的记录条数  
}

## 查询电子围栏配置

查询电子围栏配置信息  
输入参数(JSON格式)：  
{  
 data:{ 输入参数  
 floorNo: 查询的楼层  
 }  
}  
:return:  
{  
 errorCode 错误码  
 ==0 成功  
 msg 错误信息  
 =='ok' 成功  
 data:[{ 电子围栏配置  
 name: 围栏名称（所在房间）  
 points:[{ 围栏顶点坐标。（坐标应符合连续逆时针或者顺时针的顺序。当前要求：按逆时针顺序）  
 x: 横坐标，物理坐标，单位 毫米 mm  
 y: 纵坐标，物理坐标，单位 mm  
 }]  
 railNo: 电子围栏编号。后台生成的唯一编号。 added date 2017-10-01  
 id: 电子围栏 Id  
 }]  
}



url = '/lbs/get\_electronic\_rail';  
txData = {  
 "floorNo": "Floor3" // 查询的楼层号  
};



## 电子围栏告警通知

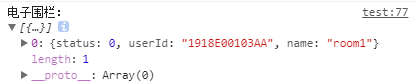
用户进入或者退出电子围栏，服务器推送信息给Web页面

推送给页面的信息是一个数组，数组中每个元素的含义：

userId 用户id，产生电子围栏告警的用户id

name 电子围栏对应的 房间 编号

status 状态, 1 进入围栏，0 退出围栏



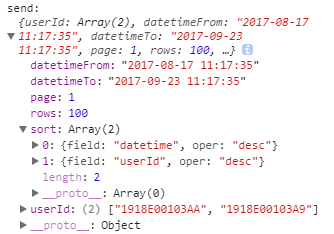
WebSocket接收消息的接口如下：

**var** namespace = '/HeZhong';  
**var** socket = io.connect(location.protocol + '//' + document.domain + ':' + location.port + namespace);  
  
socket.on('hz\_electronic\_tail', **function**(msg) {  
 console.log('电子围栏：', msg);  
});

在多个页面调用WebSocket，注意考虑使用全局WebSocket变量。

## 查询电子围栏告警信息

用户进入或者退出电子围栏，服务器推送信息给Web页面



url = '/lbs/get\_electronic\_rail\_info';  
txData = {  
 "userId": ["1918E00103AA", "1918E00103A9"], // "userId": ["all"] for all users  
 "datetimeFrom": "2017-08-17 11:17:35", // 按照时间段查询, 可选，若不填写，时间段不作为过滤条件  
 "datetimeTo": "2017-09-23 11:17:35", // 结束日期，可选，不填写时，结束日期为 Now  
 "page": 1, // 查询的页码。 当记录很多时，需要分页查询。可选参数，默认为第一页  
 "rows": 100, // 当前页记录条数。可选参数。默认100条  
 "sort": [{"field": "datetime", "oper": "desc"}, // 记录排序规则， 可选参数。  
 {"field": "userId", "oper": "desc"}] // 默认按照时间降序排序  
 // 当前条件只支持 and 。  
 // oper 取值： desc 降序（默认）， asc 升序。 可以不填oper，默认降序  
 // 字段顺序 代表 查询时的排序顺序。  
 // 目前支持 datetime, userId, room排序  
};

可选字段用法：1. 不写该字段，2. 将该字段内容赋值为空字符串。

返回信息：

errorCode 为 0 接口调用成功， 非0失败。具体错误码含义，需要统一定义。

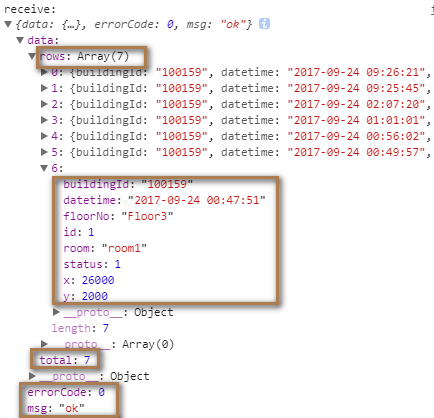
msg 成功为 ‘ok’， 失败时，返回错误原因

data 内为具体的返回记录

其中 total 为 满足查询条件的记录总数。

rows 为记录列表。

记录中的每个字段 见下图



可以参照接口调用示例：

send to server: >>>>>>>>>>  
url: /lbs/get\_electronic\_rail\_info  
{"userId":["1918E00103AA","1918E00103A9"],"datetimeFrom":"2017-08-17 11:17:35","page":1,"rows":100,"sort":[{"field":"datetime","oper":"desc"},{"field":"userId","oper":"desc"}]}

receive from server: <<<<<<<<<<  
{"data":{"rows":[{"buildingId":"100159","datetime":"2017-09-24 09:26:21","floorNo":"Floor3","id":7,"room":"room1","status":1,"x":26000,"y":2000},{"buildingId":"100159","datetime":"2017-09-24 09:25:45","floorNo":"Floor3","id":6,"room":"room1","status":0,"x":26000,"y":5000},{"buildingId":"100159","datetime":"2017-09-24 02:07:20","floorNo":"Floor3","id":5,"room":"room1","status":1,"x":26000,"y":2000},{"buildingId":"100159","datetime":"2017-09-24 01:01:01","floorNo":"Floor3","id":4,"room":"room1","status":0,"x":26000,"y":5000},{"buildingId":"100159","datetime":"2017-09-24 00:56:02","floorNo":"Floor3","id":3,"room":"room1","status":1,"x":26000,"y":2010},{"buildingId":"100159","datetime":"2017-09-24 00:49:57","floorNo":"Floor3","id":2,"room":"room1","status":0,"x":24000,"y":2000},{"buildingId":"100159","datetime":"2017-09-24 00:47:51","floorNo":"Floor3","id":1,"room":"room1","status":1,"x":26000,"y":2000}],"total":7},"errorCode":0,"msg":"ok"}

## 修改/新增电子围栏配置

url: /lbs/electronic\_rail\_cfg\_modify

修改/新增电子围栏信息  
输入参数：  
{  
 data: [{ 信息内容  
 id: 记录ID，新增时 ID可不填或者填 <=0 的数值。 修改时，必须返回记录ID。  
 若查询返回的围栏配置没有记录ID，则不允许修改。  
 name: 电子围栏名称，不允许重复  
 buildingId: 建筑id，保留  
 floorNo: 楼层号  
 points:[{ 围栏顶点坐标。（坐标应符合连续逆时针或者顺时针的顺序。

当前要求：按逆时针顺序）  
 可选参数。 当修改记录时，不涉及顶点修改，可以不传递。

其他情况（新增，修改顶点）都需要  
 传递最终状态的完整顶点信息。  
 x: 横坐标，物理坐标，单位 毫米 mm  
 y: 纵坐标，物理坐标，单位 mm  
 }]  
 }]  
}  
**:return**:  
{  
 errorCode 错误码  
 == 0 成功  
 == 100 [data] field required.  
 == 101 输入参数错误  
 == 102 围栏[%s]已经存在！ (重复添加)  
 == 103 ID为[%d]的电子围栏不存在！ (修改id不存在的电子围栏)  
 == 104 修改后的围栏名称[%s]和现有的重名！  
 == 105 围栏顶点数须大于等于3！  
 msg 错误信息  
 == 'ok' 成功提示  
}

## 删除数据接口

url: /lbs/hz\_data\_del

通用删除入口  
输入参数：  
{  
 who: 要删除的模块。字符串  
 == 'elect\_rail\_cfg' 电子围栏配置 模块  
 ids: [id1, id2, ...] 记录 id list  
}  
**:return**:  
{  
 errorCode 错误码  
 == 0 成功  
 == 101 输入参数错误！  
 == 202 未知模块[ %s ]  
 msg 错误信息  
 == [%d]个围栏，[%d]个顶点信息被删除。 (errorCode = 0 时)  
}

## 新增盘点区域

HTTP POST请求方式，url:

/lbs/people\_stat\_cfg\_add

POST数据格式：JSON。示例：

{

data: [{

'name': '北斗羲和',

'points': [{'x': 25407, 'y': 1362},

{'x': 25407, 'y': 4981}, {'x': 28058, 'y': 4981},

{'x': 28058, 'y': 1362}],

'peopleNum': 1

},{

'name': '会议室',

'points': [{'x': 1343, 'y': 1362}, {'x': 1343, 'y': 6785},

{'x': 5733, 'y': 6785}, {'x': 5733, 'y': 1362}]

}]

}

输入参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **必填** | **说明** |
| name | str | √ | 盘点区域名称 |
| points | 数组 | √ | 盘点区域顶点坐标 |
| x | int | √ | 顶点横坐标，单位mm |
| y | int | √ | 顶点纵坐标，单位mm |
| peopleNum | int | × | 盘点区域期望人数，默认为0 |

返回值示例：

{

“errorCode”: 0,

“msg”: “新增成功！”

}

返回参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| errorCode | int | 错误码 |
| msg | str | 错误信息 |

## 删除盘点区域

HTTP POST请求方式，url:

/lbs/ hz\_data\_del

POST数据格式：JSON。示例：

{

“who”: “people\_stat\_cfg”,

“ids”: [1, 2, 3]

}

输入参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **类型** | **必填** | **说明** |
| who | str | √ | 删除数据所在模块,删除盘点区域需填people\_stat\_cfg |
| ids | int数组 | √ | 盘点区域记录id列表 |

返回值示例：

{

“errorCode”: 0,

“msg”: “[3]条盘点记录被删除。”

}

返回参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| errorCode | int | 错误码 |
| msg | str | 错误信息 |

## 修改盘点区域

/lbs/people\_stat\_cfg\_chg

修改盘点区域  
输入参数：  
{  
 data: [{ 信息内容  
 id: 记录ID  
 name: 盘点区域名称，可选参数，不提供该参数，表示名称不做修改  
 points:[{ 盘点区域顶点坐标。（坐标应符合连续逆时针或者顺时针的顺序。当前要求：按逆时针顺序）  
 可选参数。 不涉及顶点修改，可以不传递。  
 若该参数有改动，需要传递最终状态的完整顶点信息。  
 x: 横坐标，物理坐标，单位 毫米 mm  
 y: 纵坐标，物理坐标，单位 mm  
 }]  
 expectNum: 应到人数，可选参数  
 buildingId: 建筑id，保留  
 floorNo: 楼层号，保留  
 }]  
}  
**:return**:  
{  
 errorCode msg  
 ------------ --------------------------------  
 0 更新成功  
 100 [data] field required.  
 200 [id] field required.  
 101 输入参数错误  
 102 盘点区域[%s]已经存在！ (重复添加)  
 103 ID为[%d]的盘点区域不存在！  
 104 修改后的盘点区域名称[%s]和现有的重名！  
 105 盘点区域顶点数须大于等于3！  
}

## 查询盘点区域

/lbs/people\_stat\_cfg\_get

查询盘点区域配置输入参数(JSON格式)：  
{  
 floorNo: 查询的楼层, 可选参数，不填则不作为过滤条件。目前该参数无实际意义  
 page: 查询的页码。 当记录很多时，需要分页查询。可选参数，默认为第1页  
 rows: 当前页记录条数。可选参数。默认50条  
  
}  
**:return**:  
{  
 errorCode msg  
 --------- -----------------------  
 0 'ok'  
  
 data:{ 盘点区域配置  
 total: 符合条件的记录总条数  
 rows:[{ 盘点信息详情  
 name: 盘点区域名称（所在房间）  
 points:[{ 盘点区域顶点坐标。（坐标应符合连续逆时针或者顺时针的顺序。当前要求：按逆时针顺序）  
 x: 横坐标，物理坐标，单位 毫米 mm  
 y: 纵坐标，物理坐标，单位 mm  
 }]  
 zoneNo: 盘点区域编号。后台生成的唯一编号。  
 id: 盘点区域 Id  
 expectNum: 期望人数  
 createAt: 区域创建时间  
 buildId: 建筑ID  
 floorNo: 楼层号  
 }]  
 }  
}

## 设置默认盘点区域

/lbs/people\_stat\_cfg\_add\_default

新增默认盘点区域  
  
输入参数：  
无  
**:return**:  
{  
 errorCode msg  
 ------------ -----------------------------  
 0 新增[%d]个盘点区域！  
 105 盘点区域顶点数须大于等于3！  
}

默认增加北斗羲和、智慧消防、智慧安防、健身房、智慧园区、智慧医疗、大会议室、会议室、副总办公室、总裁室、仓库1、仓库2、机房、前台共14个区域（保存在rooms.py文件中）。

**如果之前添加过重名区域，则本次不会对重名区域做修改，会继续添加后续区域。**

## 立即盘点

对配置过的盘点区域，执行人员盘点。会返回盘点结果。

/lbs/people\_stat\_do

立即盘点  
  
输入参数：  
无  
**:return**:  
{  
 errorCode msg  
 ------------ -----------------------------  
 0 'ok'  
  
 statInfo:{ 盘点信息  
 id 记录ID  
 statNo 盘点编号  
 roomName 盘点区域名称  
 roomId 盘点区域ID  
 roomNo 盘点区域编号  
 roomCreateAt 盘点区域创建时间  
 curPeopleNum 盘点时人数  
 expectNum 期望人数  
 datetime 盘点时间  
 }  
}

## 新增定时盘点任务

定时盘点任务时间到后，自动执行盘点。省去人工执行盘点的人力开销。

/lbs/people\_stat\_task\_add

增加定时盘点任务  
**:param** param: 输入参数，JSON 格式  
{  
 data: [{ 日期相关参数，限制任务的开始时间  
 name: str 任务名称，唯一  
 dayOfWeek: str 周几，可选，不填或为空字符，表示每天循环。 0 表示周一，6 表示周六。  
 多个日期用逗号分割，例如：0,1,2,3,4 表示周一至周五  
 hour: str 小时，取值范围 0~23。多个小时用逗号分割  
 minute: str 分钟，取值范围 0~59。多个分钟用逗号分割。可选，默认为0  
 second: str 秒，取值范围 0~59。多个秒值用逗号分割。 可选，默认为0  
 startDate: str 任务开始日期（包含本日期），格式： 年-月-日 时:分:秒 （2017-12-24 13:30:00）  
 可选参数，不填或为空字符，该字段无意义  
 endDate: str 任务结束日期（包含本日期），格式： 年-月-日 时:分:秒 （2018-12-24 13:30:00）  
 可选参数，不填或为空字符，该字段无意义  
 }]  
}  
**:return**:  
{  
 errorCode msg  
 --------- --------------------------------------------------------  
 0 ok  
 100 缺少输入参数！  
 101 输入参数错误！缺少[data]字段.  
 102 输入参数错误！缺少[hour]字段  
 103 输入参数错误！缺少[name]字段  
 104 [name]字段[=%s]已经存在！  
 200 任务信息写数据库失败。  
}

## 删除定时盘点任务

/lbs/hz\_data\_del

删除定时盘点任务  
:输入参数：

{

who: 要删除的模块。字符串  
 == ' people\_stat\_job' 定时盘点任务  
 ids: [id1, id2, ...] 记录 id list，id为int类型  
}  
**:return**:  
{  
 errorCode msg  
 --------- --------------------------------------------------------  
 0 [%d]个job记录被删除，[%d]个运行job被删除。  
}

## 修改定时盘点任务

/lbs/people\_stat\_task\_chg

修改定时盘点任务  
  
**:param** param: 输入参数，JSON 格式  
{  
 data: [{ 要修改的任务字段，可选字段，表示忽略对应字段  
 id: int 任务id  
 name: str 任务名称，可选参数  
 dayOfWeek: str 周几，可选参数  
 0 表示周一，6 表示周六。  
 多个日期用逗号分割，例如：0,1,2,3,4 表示周一至周五  
 hour: str 小时，取值范围 0~23。多个小时用逗号分割， 可选参数  
 minute: str 分钟，取值范围 0~59。多个分钟用逗号分割。可选参数  
 second: str 秒，取值范围 0~59。多个秒值用逗号分割。 可选参数  
 startDate: str 任务开始日期（包含本日期），格式： 年-月-日 时:分:秒 （2017-12-24 13:30:00）  
 可选参数  
 endDate: str 任务结束日期（包含本日期），格式： 年-月-日 时:分:秒 （2018-12-24 13:30:00）  
 可选参数  
 }]  
}  
**:return**:  
{  
 errorCode msg  
 --------- --------------------------------------------------------  
 0 ok  
 100 缺少输入参数！  
 101 输入参数错误，缺少[data]字段！  
 102 输入参数错误，缺少[id]字段！  
}

## 查询定时盘点任务

/lbs/people\_stat\_task\_get

查询定时盘点任务  
**:param** param:  
{  
 page: 查询的页码。 当记录很多时，需要分页查询。可选参数，默认为第1页  
 rows: 当前页记录条数。可选参数。默认50条  
}  
**:return**:  
{  
 errorCode msg  
 --------- --------------------------------------------------------  
 0 ok  
 100 缺少输入参数！  
  
  
 data: [{  
 total: int 符合条件的记录条数  
 rows:[{  
 id: int 任务id， 唯一  
 no: str 任务编号，唯一  
 name: str 任务名称  
 dayOfWeek: str 周几，为空，表示每天循环。 0 表示周一，6 表示周六。  
 多个日期用逗号分割，例如：0,1,2,3,4 表示周一至周五  
 hour: str 小时，取值范围 0~23。多个小时用逗号分割  
 minute: str 分钟，取值范围 0~59。多个分钟用逗号分割  
 second: str 秒，取值范围 0~59。多个秒值用逗号分割。 默认为0  
 startDate: str 任务开始日期（包含本日期），格式： 年-月-日 时:分:秒 （2017-12-24 13:30:00）  
 为空字符，该字段无意义  
 endDate: str 任务结束日期（包含本日期），格式： 年-月-日 时:分:秒 （2018-12-24 13:30:00）  
 为空字符，该字段无意义  
 createAt: datetime 任务创建时间，格式 yyyy-mm-dd HH:mm:ss  
 }]  
 }]  
}

## 查询盘点结果

/lbs/people\_stat\_get

查询盘点结果  
  
输入参数：  
{  
 statNo: 盘点编号， 例如：'PD-20171222-165201', 可选，若不填写，该字段不作为过滤条件  
 roomName: 盘点区域名称，例如：["北斗羲和", "智慧消防"], 可选，若不填写，该字段不作为过滤条件  
 datetimeFrom: 查询起始时间，例如："2017-08-17 11:17:35", 可选，若不填写，该字段不作为过滤条件  
 datetimeTo: 查询结束时间，例如："2017-09-23 11:17:35", 可选，不填写时，该字段不作为过滤条件  
 page: 查询的页码。 当记录很多时，需要分页查询。可选参数，默认为第1页  
 rows: 当前页记录条数。可选参数。默认100条  
 sort: [{"field": "datetime", "oper": "desc"}, 记录排序规则， 可选参数，默认按照时间降序排序  
 {"field": "roomName", "oper": "desc"}] 按照房间名称排序，可选参数，默认按照升序排序  
 当前条件只支持 and 。  
 oper 取值： desc 降序（默认）， asc 升序。 可以不填oper，默认降序  
 字段顺序 代表 查询时的排序顺序。  
 目前支持 datetime, roomName 排序  
}  
  
**:return**:  
{  
 errorCode msg  
 --------- --------------------------------------------------------  
 0 ok  
  
 data:{ 数据内容  
 total: 符合条件的记录条数  
 rows:[{ 盘点信息详情  
 id: 记录ID  
 statNo: 盘点编号  
 roomName: 盘点区域名称  
 roomId: 盘点区域ID  
 roomNo: 盘点区域编号  
 roomCreateAt: 盘点区域创建时间  
 curPeopleNum: 盘点时人数  
 expectNum: 期望人数  
 datetime: 盘点时间  
 }]  
 }  
}

## 删除盘点结果

/lbs/hz\_data\_del

删除盘点结果  
输入参数：  
{  
 who: 要删除的模块。字符串  
 == 'people\_stat\_result' 盘点记录  
 ids: [id1, id2, ...] 记录 id list  
}  
**:return**:  
{  
 errorCode msg  
 ---------- ------------------------------  
 0 [%d]条盘点记录被删除。

}

# 二、修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修改日期 | 修改内容 | 修改人 |
| V0.3 | 2071-12-12 | 1.查询历史轨迹  返回参数data下*datatime*改名为datetime。 | 王秀国 |
| V0.3 | 2017-12-24 | 新增人员盘点接口 7~18  新增盘点区域  删除盘点区域  修改盘点区域  查询盘点区域  设置默认盘点区域  立即盘点  新增定时盘点任务  删除定时盘点任务  修改定时盘点任务  查询盘点结果  删除盘点结果 | 王秀国 |
|  |  |  |  |
|  |  |  |  |