

# 基于 BS 架构的 HTTP 对接协议

版本	日期	文档名称	文档编号	页码
V1.16	2021-01	基于 BS 架构的 HTTP 对接协议	AX-ACS-BS-HTTP	共 37 页
编制		审核	密级	
			III 级（选择性公开）	

## 修订历史

版本	日期	原因
V1.00	2018/10/30	创建文档
V1.01	2018/11/12	
V1.02	2019/01/10	
V1.0.3	2019/02/27	
V1.0.4	2019/05/07	
V1.0.5	2019/06/06	添加通过 ID 获取刷卡记录
V1.0.6	2019/07/18	获取刷卡记录增加工号 ( userno ) 字段过滤
V1.0.7	2020/02/14	添加停车场管理接口
V1.0.8	2020/02/25	添加获取所有门禁设备接口 添加获取门禁实时刷卡记录接口
V1.0.9	2020/04/12	增加获取所有时间组数据接口
V1.10	2020/04/14	
V1.11	2020/04/24	
V1.12	2020/04/28	添加增量获取机构, 用户, 卡片相关数据接口 增加获取人脸实时记录等相关人脸接口
V1.13	2020/05/12	增加设置门工作模式接口 用户数据加入照片 URL 字段
V1.14	2020/06/04	门禁刷卡记录增加区域 ID、设备 ID、门 ID
V1.15	2020/06/09	增加获取用户通行记录接口
V1.16	2021/01/21	增加获取门禁记录 ( 所有类型 ) 接口 增加获取某个门的某天可通行名单接口

## 目 录

1. 概述.....	1
2. 接口说明.....	2
2.1.1 人员类型 .....	2
2.1.2 机构 .....	3
2.1.3 用户、卡片 .....	4
2.1.4 事件记录 .....	14
2.1.5 门禁设备操作 .....	28
2.1.6 其他接口 .....	35

# 1. 概述

## 1. 接口方式

本接口采用 RESTful API 形式。访问 URL 一般是：`http://ip:6500/api/v{n}/func`，其中 ip 是门禁安防系统的服务器地址，func 是具体的 api 功能。详细见下面。

- HTTP 接口请求数据格式采用 utf-8 格式
- 所有 POST 请求 content-Type 为 application/json
- POST 请求接口返回数据格式为 application/json;charset=UTF-8
- POST 请求参数类型采用 json 形式

## 2. 返回数据格式

接口统一采用 json 格式返回数据，返回数据的通用格式如下：

```
{
  "status": 0,
  "msg": "success",
  "data": {} || []
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息。
data	Json	返回内容。如果是返回单个数据，则是一个 json 格式的对象，如果是返回多条记录的，则是一个 json 格式的数组。

## 3. 通用返回码对照表

返回码	说明
0	成功
1	一般错误
101	非法参数
102	数据不存在
103	数据已经存在，不能重复添加
104	登录帐号不存在
105	帐号验证错误
106	帐号已被锁定
500	服务器发生异常

## 2. 接口说明

### 2.1.1 人员类型

#### 1. 获取人员类型数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/usertypes>
- 请求参数：无
- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "userId" : 1897,
      "userName" : "职工"
    },
    {
      "userId" : 1898,
      "userName" : "车辆"
    },
    {
      "userId" : 1899,
      "userName" : "临时施工人员"
    },
    {
      "userId" : 1900,
      "userName" : "其他人员"
    },
    {
      "userId" : 1990,
      "userName" : "临时施工车辆"
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	人员类型数组。

- 人员类型数据说明：

字段名	类型	说明
userId	Int	人员类型 ID
typeName	String	人员类型名称

## 2.1.2 机构

### 1. 获取所有机构（部门）数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/orgs>
- 请求参数：无
- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "orgId" : 1,
      "orgName" : "公司",
      "orgCode" : "",
      "parentId" : 0,
      "parentName" : "",
      "gmtCreate": "2018-03-23 11:33:24.0",
      "gmtModify": "2020-03-18 16:33:29"
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	机构数据数组。

- 机构数据说明：

字段名	类型	说明
orgId	Int	机构 ID
orgName	String	机构名称

orgCode	String	机构编码
parentId	Int	上级机构 ID
parentName	String	上级机构名称
gmtCreate	String	创建时间
gmtModify	String	最后修改时间

## 2. 通过机构名称获取单个机构数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/org/name/{name}>
- 说明：根据名称获取机构信息
- 请求参数：

类型	名称	类型	说明
Path	name	String	要获取的机构名称，必填。

- 返回数据格式：

```
{
  "status": 0,
  "msg": "success",
  "data": {
    "orgId": 1,
    "orgName": "测试部门",
    "orgCode": "TD",
    "parentId": "0",
    "parentName": ""
  }
}
```

- 机构数据说明

字段名	类型	说明
orgId	Int	机构 Id
orgName	String	机构名称
orgCode	String	机构编码
parentId	Int	上级机构 Id
parentName	String	上级机构名称

### 2.1.3 用户、卡片

#### 1. 获取用户数量

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/users/count?usertype=0&orgname=>
- 请求参数：

参数名	类型	说明
-----	----	----

usertype	Int	用户类型 ID, 获取指定用户类型的用户数。默认为 0, 取所有用户类型的用户数。 ID 值参考: 获取的人员类型数据
orgname	String	机构名称, 获取指定机构下面的用户数。默认为空, 获取所有机构的用户数

- 返回数据格式:

```
{
  "status": 0,
  "msg": "success",
  "data": 22
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码, 详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	用户数

### 2. 获取用户数据

- HTTP 请求方式: GET
- HTTP 请求 URL: <http://ip:6500/api/v1/users?usertype=0&orgname=&offset=0&rows=100>
- 请求参数:

参数名	类型	说明
usertype	Int	用户类型 ID, 获取指定用户类型的用户。默认为 0, 取所有用户类型的用户。ID 值参考: 获取的人员类型数据
orgname	String	机构名称, 获取指定机构下面的用户。默认为空, 获取所有机构的用户
offset	Int	分页获取的起始位置
rows	Int	分页获取的该页记录数。rows 的最大值是 1000, 如果超过 1000, 将会返回非法参数

- 返回数据格式:

```
{
  "status": 0,
  "msg": "success",
  "data": [
    {
      "userId": 20173,
      "userNo": "12700106",
      "userName": "张三",
      "sex": 0,
      "orgId": 3658,
      "orgName": "qt 洗衣服务部",
      "email": "",
      "mobile": "",
      "status": 1,
      "idCard": ""
    }
  ]
}
```

```
"userId": 1899,
"userTypeId": 1899,
"userTypeName": "临时施工人员",
"gmtBirthday": "",
"jobPosition": "员工",
"plateNumber": "",
"address": "",
"company": "",
"registeredResidence": "",
"gmtExpireDate": "",
"educatingPerson": "",
"gmtEducate": "",
"gmtSafetyEducateStart": "",
"gmtSafetyEducateEnd": "",
"constructionProjectName": "",
"constructionLocation": "",
"gmtConstructionStart": "",
"gmtConstructionEnd": "",
"constructionPersonInCharge": "",
"safetyPersonInCharge": "",
"mobilePersonInCharge": "",
"phonePersonInCharge": "",
"carDriverName": "",
"carDriverPassport": "",
"carTypeName": "",
"carName": "",
"carRatedLoad": "",
"carServiceTypeName": "",
"remark": "",
"gmtCreate": "2018-03-09 15:06:19.0",
"gmtModify": "2018-03-09 15:06:19.0",
"cardState": 1,
"cardCount": 1,
"cardNos": "17435103",
"gmtCardExpireDate": "2099-12-31 23:59:59",
"cardTypeId": 5,
"cardGroupId": 3,
"cardPassword": "",
"cardData": "upload/faceimages/f7494baec995407d84ed1c836544c006.jpg"
"photoUrl": "upload/image/aa.jpg"
},
]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码,详情参照通用返回码对照表。如果没有数据,则 status=102
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	用户的数据数组

● 用户数据说明：

字段名	类型	说明
userId	Int	用户 ID
userNo	String	用户工号
userName	String	用户姓名
sex	Int	性别, 0 : 男, 1 : 女
orgId	Int	所属机构 ID
orgName	String	所属机构名称
email	String	电子邮件
mobile	String	手机号码
status	Int	登录状态。0 : 禁用, 1 : 正常
idCard	String	证件号码
userId	Int	用户类型 ID
userIdName	String	用户类型名称
gmtBirthday	String	出生日期
jobPosition	String	职位
plateNumber	String	车牌号码
address	String	地址
company	String	所属单位
registeredResidence	String	户口所在地
gmtExpireDate	String	到期日期
educatingPerson	String	教育人
gmtEducate	String	教育日期
gmtSafetyEducateStart	String	安全教育期开始日期
gmtSafetyEducateEnd	String	安全教育期结束日期
constructionProjectName	String	项目名称
constructionLocation	String	施工地点
gmtConstructionStart	String	工程期限开始日期
gmtConstructionEnd	String	工程期限结束日期
constructionPersonInCharge	String	施工负责人
safetyPersonInCharge	String	安全负责人
mobilePersonInCharge	String	施工安全固定电话
phonePersonInCharge	String	施工安全手机电话
carDriverName	String	司机
carDriverPassport	String	驾驶人通行证编码
carTypeName	String	车型
carName	String	车辆名称
carRatedLoad	String	额定载质量

carServiceTypeName	String	服务性质
remark	String	人员备注
gmtCreate	String	创建时间
gmtModify	String	最后修改时间
cardState	Int	卡片状态, 只返回最高级别一个。挂失>正常
cardCount	Int	卡片数量
cardNos	String	卡号集合, 逗号隔开
gmtCardExpireDate	String	卡片失效时间, 只返回一个, 最先失效的一个
cardTypeId	Int	卡片类型, 只返回最高级别一个。人脸>IC 卡
cardGroupId	Int	卡组, 只返回最高级别一个。人脸>IC 卡
cardPassword	String	卡片密码, 只返回最高级别一个。人脸>IC 卡
cardData	String	动态人脸照片 URL
photoUrl	String	人员照片 URL

### 3. 获取单个用户数据

- HTTP 请求方式：GET
- HTTP 请求 URL：http://ip:6500/api/v1/user?usertype=0&userno=123
- 请求参数：

参数名	类型	说明
usertype	Int	用户类型 ID, 获取指定用户类型的用户。默认为 0, 忽略用户类型。ID 值参考：获取的人员类型数据
userno	String	用户工号或者证件号码关键字。

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : {
    "userId" : 289,
    "userNo" : "300005",
    "userName" : "买买提",
    "sex" : 0,
    "orgId" : 2483,
    "orgName" : "机关党群工作部",
    "email" : "",
    "mobile" : "",
    "status" : 1,
    "idCard" : "",
    "userId" : 1897,
    "typeName" : "职工",
    "gmtBirthday" : "",
    "jobPosition" : "员工",
```

```

"plateNumber" : "",
"address" : "",
"company" : "",
"registeredResidence" : "",
"gmtExpireDate" : "",
"educatingPerson" : "",
"gmtEducate" : "",
"gmtSafetyEducateStart" : "",
"gmtSafetyEducateEnd" : "",
"constructionProjectName" : "",
"constructionLocation" : "",
"gmtConstructionStart" : "",
"gmtConstructionEnd" : "",
"constructionPersonInCharge" : "",
"safetyPersonInCharge" : "",
"mobilePersonInCharge" : "",
"phonePersonInCharge" : "",
"carDriverName" : "",
"carDriverPassport" : "",
"carTypeName" : "",
"carName" : "",
"carRatedLoad" : "",
"carServiceTypeName" : "",
"remark" : "",
"gmtCreate": "2018-03-23 11:33:24.0",
"gmtModify": "2020-03-18 16:33:29",
"cardState": 1,
"cardCount": 3,
"cardNos": "22,111,10000033",
"gmtCardExpireDate": "2099-12-31 23:59:59",
"cardTypeId": 5,
"cardGroupId": 3,
"cardPassword": "",
"cardData":
"upload/faceimages/f7494baec995407d84ed1c836544c006.jpg" }
}
    
```

字段名	类型	说明
status	Int	接口执行的返回状态码,详情参照通用返回码对照表。如果没有数据,则 status=102
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	单个用户的数据对象,参考:用户数据说明

## 4. 根据工号获取单个用户数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/user/userno/{userno}>
- 说明：根据用户工号或者证件号码获取用户数据
- 请求参数：

类型	名称	类型	说明
Path	userno	String	用户工号或者证件号码关键字

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : {
    "userId" : 289,
    "userNo" : "300005",
    "userName" : "买买提",
    "sex" : 0,
    "orgId" : 2483,
    "orgName" : "机关党群工作部",
    "email" : "",
    "mobile" : "",
    "status" : 1,
    "idCard" : "",
    "userId" : 1897,
    "typeName" : "职工",
    "gmtBirthday" : "",
    "jobPosition" : "员工",
    "plateNumber" : "",
    "address" : "",
    "company" : "",
    "registeredResidence" : "",
    "gmtExpireDate" : "",
    "educatingPerson" : "",
    "gmtEducate" : "",
    "gmtSafetyEducateStart" : "",
    "gmtSafetyEducateEnd" : "",
    "constructionProjectName" : "",
    "constructionLocation" : "",
    "gmtConstructionStart" : "",
    "gmtConstructionEnd" : "",
    "constructionPersonInCharge" : "",
    "safetyPersonInCharge" : ""
  }
}
```

```

"mobilePersonInCharge" : "",
"phonePersonInCharge" : "",
"carDriverName" : "",
"carDriverPassport" : "",
"carTypeName" : "",
"carName" : "",
"carRatedLoad" : "",
"carServiceTypeName" : "",
"remark" : "",
"gmtCreate": "2018-03-23 11:33:24.0",
"gmtModify": "2020-03-18 16:33:29",
"cardState": 1,
"cardCount": 3,
"cardNos": "22,111,10000033",
"gmtCardExpireDate": "2099-12-31 23:59:59",
"cardTypeId": 5,
"cardGroupId": 3,
"cardPassword": "",
"cardData":
"upload/faceimages/f7494baec995407d84ed1c836544c006.jpg" }
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码,详情参照通用返回码对照表。如果没有数据,则 status=102
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	单个用户的数据对象,参考:用户数据说明

## 5. 根据 ID 获取单个用户数据

- HTTP 请求方式: GET
- HTTP 请求 URL: <http://ip:6500/api/v1/user/userid/{userid}>
- 说明: 根据用户 ID 获取用户数据
- 请求参数:

类型	名称	类型	说明
Path	userid	int	用户 id

- 返回数据格式:

```

{
  "status" : 0,
  "msg" : "success",
  "data" : {
    "userId" : 289,
    "userNo" : "300005",
    "userName" : "买买提",
    "sex" : 0,

```

```
"orgId" : 2483,
"orgName" : "机关党群工作部",
"email" : "",
"mobile" : "",
"status" : 1,
"idCard" : "",
"userTypeId" : 1897,
"userTypeName" : "职工",
"gmtBirthday" : "",
"jobPosition" : "员工",
"plateNumber" : "",
"address" : "",
"company" : "",
"registeredResidence" : "",
"gmtExpireDate" : "",
"educatingPerson" : "",
"gmtEducate" : "",
"gmtSafetyEducateStart" : "",
"gmtSafetyEducateEnd" : "",
"constructionProjectName" : "",
"constructionLocation" : "",
"gmtConstructionStart" : "",
"gmtConstructionEnd" : "",
"constructionPersonInCharge" : "",
"safetyPersonInCharge" : "",
"mobilePersonInCharge" : "",
"phonePersonInCharge" : "",
"carDriverName" : "",
"carDriverPassport" : "",
"carTypeName" : "",
"carName" : "",
"carRatedLoad" : "",
"carServiceTypeName" : "",
"remark" : "",
"gmtCreate": "2018-03-23 11:33:24.0",
"gmtModify": "2020-03-18 16:33:29",
"cardState": 1,
"cardCount": 3,
"cardNos": "22,111,10000033",
"gmtCardExpireDate": "2099-12-31 23:59:59",
"cardTypeId": 5,
"cardGroupId": 3,
```

```

    "cardPassword": "",
    "cardData":
"upload/faceimages/f7494baec995407d84ed1c836544c006.jpg"  }
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码,详情参照通用返回码对照表。如果没有数据,则 status=102
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	单个用户的数据对象,参考:用户数据说明

## 6. 获取所有卡组数据

- HTTP 请求方式: GET
- HTTP 请求 URL: <http://ip:6500/api/v1/cardgroups>
- 请求参数: 无
- 返回数据格式:

```

{
  "status": 0,
  "msg": "success",
  "data": [
    {
      "id": "1",
      "name": "特权卡"
    },
    {
      "id": "2",
      "name": "胁迫卡"
    },
    {
      "id": "3",
      "name": "普通组"
    }
  ]
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码,详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	数据数组。

- 卡组数据说明:

字段名	类型	说明
id	String	ID

name	String	名称
------	--------	----

## 7. 获取所有卡类型数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/cardtypes>
- 请求参数：无
- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "id" : "1",
      "name" : "韦根 26"
    },
    {
      "id" : "2",
      "name" : "韦根 34"
    },
    {
      "id" : "3",
      "name" : "人脸识别"
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	数据数组。

- 卡类型数据说明：

字段名	类型	说明
id	String	ID
name	String	名称

## 2.1.4 事件记录

### 1. 获取刷卡记录数量

- HTTP 请求方式：GET
- HTTP 请求 URL：

<http://ip:6500/api/v1/>

[swipecardrecords/count?usertype=0&isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59&userno=11](http://ip:6500/api/v1/swipecardrecords/count?usertype=0&isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59&userno=11)

- 请求参数：

参数名	类型	说明
usertype	Int	用户类型 ID, 获取指定用户类型的刷卡数量。默认为 0, 取所有用户类型的数量。ID 值参考：获取的人员类型数据
isalarm	Int	正常刷卡和非法刷卡过滤参数。默认为 0, 只获取正常刷卡记录数, 1：只获取非法刷卡记录数, -1：获取正常和非法刷卡记录数。
begintime	String	刷卡记录的开始时间, 不能为空。格式为：yyyy-MM-DD+HH:MM:SS
endtime	String	刷卡记录的结束时间, 不能为空。格式为：yyyy-MM-DD+HH:MM:SS
userno	String	人员工号, 这里使用精确匹配, 如果为空, 不判断工号

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : 22
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码, 详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	刷卡记录数

## 2. 获取刷卡记录数据

- HTTP 请求方式：GET
- HTTP 请求 URL：

<http://ip:6500/api/v1/>

[swipecardrecords?usertype=0&isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59&userno=11&offset=0&rows=100](http://ip:6500/api/v1/swipecardrecords?usertype=0&isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59&userno=11&offset=0&rows=100)

- 请求参数：

参数名	类型	说明
usertype	Int	用户类型 ID, 获取指定用户类型的刷卡记录。默认为 0, 取所有用户类型的刷卡记录。ID 值参考：获取的人员类型数据
isalarm	Int	正常刷卡和非法刷卡过滤参数。默认为 0, 只获取正常刷卡记录, 1：只获取非法刷卡记录, -1：获取正常和非法刷卡记录。
begintime	String	刷卡记录的开始时间, 不能为空。格式为：yyyy-MM-DD+HH:MM:SS
endtime	String	刷卡记录的结束时间, 不能为空。格式为：yyyy-MM-DD+HH:MM:SS
userno	String	人员工号, 这里使用精确匹配, 如果为空, 不判断工号
offset	Int	分页获取的起始位置
rows	Int	分页获取的该页记录数。rows 的最大值是 1000, 如果超过 1000, 将会返回非法参

		数
--	--	---

● 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "recordId": 29,
      "gmtSwipeCardTime": "2019-04-04 18:52:46",
      "areaId": 160,
      "areaName": "test",
      "deviceId": "3aec7c4e58ab4a3dbd56deb98cf4bbd6",
      "deviceName": "192.168.1.64",
      "doorId": "132ccea02e5f4498be494730fba3d9a6",
      "doorName": "东门",
      "acsEventValue": 3,
      "acsEventName": "外部刷卡开门",
      "isSwipeCard": 1,
      "isOpenDoor": 1,
      "inOrOut": 1,
      "isAlarm": 0,
      "accessBlockedReason": 0,
      "cardNo": "888888",
      "cardState": 1,
      "userId": 3,
      "userNo": "888888",
      "userName": "888888",
      "userOrgId": 0,
      "userOrgName": "",
      "plateNumber": "",
      "userType": 2    },
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	刷卡记录数组

● 刷卡记录说明：

字段名	类型	说明
recordId	Int	记录 ID
gmtSwipeCardTime	String	刷卡时间

areaid	Int	所属区域 ID
areaName	String	刷卡所在区域名称
deviceId	String	刷卡所在的门禁控制器 ID
deviceName	String	刷卡所在门禁控制器名称
doorId	String	刷卡所在的门 ID
doorName	String	刷卡所在的门名称
acsEventValue	Int	门禁事件值
acsEventName	String	门禁事件名称
isSwipeCard	Int	是否是刷卡事件。1：是，0：不是
isOpenDoor	Int	是否是开门时间。1：是，0：不是
inOrOut	Int	进出标记。1：进，2：出
isAlarm	Int	是否是非法报警事件
accessBlockedReason	Int	拒绝原因（保留）
cardNo	String	刷卡卡号
cardState	Int	卡状态。1：正常，2：挂失。
userId	Int	刷卡用户 ID
userNo	String	刷卡用户工号
userName	String	刷卡用户姓名
userOrgId	Int	刷卡用户所属机构 ID
userOrgName	String	刷卡用户所属机构名称
plateNumber	String	刷卡用户的车牌号码
userType	Int	刷卡用户的用户类型 ID

### 3. 根据记录 ID 获取刷卡记录数量

- HTTP 请求方式：GET
- HTTP 请求 URL：  
<http://ip:6500/api/v1/swipecardrecords/byid/count?id=1&usertype=0&isalarm=0>
- 请求参数：

参数名	类型	说明
id	Int	记录 ID。
usertype	Int	用户类型 ID，获取指定用户类型的刷卡数量。默认为 0，取所有用户类型的数量。 ID 值参考：获取的人员类型数据
isalarm	Int	正常刷卡和非法刷卡过滤参数。默认为 0，只获取正常刷卡记录数，1：只获取非法刷卡记录数，-1：获取正常和非法刷卡记录数。

- 返回数据格式：

```
{
  "status": 0,
  "msg": "success",
  "data": 22
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表

msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	返回记录 ID 比参数 id 大的刷卡记录数

#### 4. 根据记录 ID 获取刷卡记录数据

- HTTP 请求方式：GET
- HTTP 请求 URL：  
<http://ip:6500/api/v1/swipecardrecords/byid/?id=1&usertype=0&isalarm=0&offset=0&rows=100>
- 请求参数：

参数名	类型	说明
id	Int	记录 ID。
usertype	Int	用户类型 ID，获取指定用户类型的刷卡记录。默认为 0，取所有用户类型的刷卡记录。ID 值参考：获取的人员类型数据
isalarm	Int	正常刷卡和非法刷卡过滤参数。默认为 0，只获取正常刷卡记录，1：只获取非法刷卡记录，-1：获取正常和非法刷卡记录。
offset	Int	分页获取的起始位置
rows	Int	分页获取的该页记录数。rows 的最大值是 1000，如果超过 1000，将会返回非法参数

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "recordId": 29,
      "gmtSwipeCardTime": "2019-04-04 18:52:46",
      "areaId": 160,
      "areaName": "test",
      "deviceId": "3aec7c4e58ab4a3dbd56deb98cf4bbd6",
      "deviceName": "192.168.1.64",
      "doorId": "132ccea02e5f4498be494730fba3d9a6",
      "doorName": "东门",
      "acsEventValue": 3,
      "acsEventName": "外部刷卡开门",
      "isSwipeCard": 1,
      "isOpenDoor": 1,
      "inOrOut": 1,
      "isAlarm": 0,
      "accessBlockedReason": 0,
      "cardNo": "888888",
      "cardState": 1,
      "userId": 3,
    }
  ]
}
```

```

    "userNo": "888888",
    "userName": "888888",
    "userOrgId": 0,
    "userOrgName": "",
    "plateNumber": "",
    "userType": 2
  }
]
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码, 详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	刷卡记录数组

- 刷卡记录说明：

见“获取刷卡记录数据”说明

## 5. 获取门禁记录数量

- HTTP 请求方式：GET
- HTTP 请求 URL：

<http://ip:6500/api/v1/>

<acsrecords/count?&isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59>

- 请求参数：

参数名	类型	说明
isalarm	Int	报警和非报警过滤参数。默认为 0，只获取正常记录数，1：报警记录数，-1：获取正常和报警记录数。注意：此记录包含刷卡记录
begintime	String	记录的开始时间，不能为空。格式为：yyyy-MM-DD+HH:MM:SS
endtime	String	记录的结束时间，不能为空。格式为：yyyy-MM-DD+HH:MM:SS

- 返回数据格式：

```

{
  "status" : 0,
  "msg" : "success",
  "data" : 22
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码, 详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	记录数

## 6. 获取门禁记录数据

- HTTP 请求方式：GET

- HTTP 请求 URL :

<http://ip:6500/api/v1/>

[acsrecords?isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59&offset=0&rows=100](http://ip:6500/api/v1/acsrecords?isalarm=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59&offset=0&rows=100)

- 请求参数 :

参数名	类型	说明
isalarm	Int	报警和非报警过滤参数。默认为 0，只获取正常记录数，1：报警记录数，-1：获取正常和报警记录数。注意：此记录包含刷卡记录
begintime	String	记录的开始时间，不能为空。格式为：yyyy-MM-DD+HH:MM:SS
endtime	String	记录的结束时间，不能为空。格式为：yyyy-MM-DD+HH:MM:SS
offset	Int	分页获取的起始位置
rows	Int	分页获取的该页记录数。rows 的最大值是 1000，如果超过 1000，将会返回非法参数

- 返回数据格式 :

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "recordId": 29,
      "gmtSwipeCardTime": "2019-04-04 18:52:46",
      "areaId": 160,
      "areaName": "test",
      "deviceId": "3aec7c4e58ab4a3dbd56deb98cf4bbd6",
      "deviceName": "192.168.1.64",
      "doorId": "132ccea02e5f4498be494730fba3d9a6",
      "doorName": "东门",
      "acsEventValue": 3,
      "acsEventName": "外部刷卡开门",
      "isSwipeCard": 1,
      "isOpenDoor": 1,
      "inOrOut": 1,
      "isAlarm": 0,
      "accessBlockedReason": 0,
      "cardNo": "888888",
      "cardState": 1,
      "userId": 3,
      "userNo": "888888",
      "userName": "888888",
      "userOrgId": 0,
      "userOrgName": "",
      "plateNumber": "",
      "userType": 2    },
  ]
}
```

```

    ]
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	刷卡记录数组

● 门禁记录说明：

字段名	类型	说明
recordId	Int	记录 ID
gmtSwipeCardTime	String	刷卡时间
areaId	Int	所属区域 ID
areaName	String	刷卡所在区域名称
deviceId	String	刷卡所在的门禁控制器 ID
deviceName	String	刷卡所在门禁控制器名称
doorId	String	刷卡所在的门 ID
doorName	String	刷卡所在的门名称
acsEventValue	Int	门禁事件值
acsEventName	String	门禁事件名称
isSwipeCard	Int	是否是刷卡事件。1：是，0：不是
isOpenDoor	Int	是否是开门时间。1：是，0：不是
inOrOut	Int	进出标记。1：进，2：出
isAlarm	Int	是否是非法报警事件
accessBlockedReason	Int	拒绝原因（保留）
cardNo	String	刷卡卡号
cardState	Int	卡状态。1：正常，2：挂失。
userId	Int	刷卡用户 ID
userNo	String	刷卡用户工号
userName	String	刷卡用户姓名
userOrgId	Int	刷卡用户所属机构 ID
userOrgName	String	刷卡用户所属机构名称
plateNumber	String	刷卡用户的车牌号码
userType	Int	刷卡用户的用户类型 ID

### 7. 根据记录 ID 获取门禁记录数量

- HTTP 请求方式：GET
- HTTP 请求 URL：  
<http://ip:6500/api/v1/acsrecords/byid/count?id=1&isalarm=0>
- 请求参数：

参数名	类型	说明
id	Int	记录 ID。
isalarm	Int	报警和非报警过滤参数。默认为 0，只获取正常记录数，1：报警记录数，-1：获

		取正常和报警记录数。注意：此记录包含刷卡记录
--	--	------------------------

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : 22
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	返回记录 ID 比参数 id 大的门禁记录数

## 8. 根据记录 ID 获取门禁记录数据

- HTTP 请求方式：GET
- HTTP 请求 URL：

<http://ip:6500/api/v1/acsrecords/byid/?id=1&isalarm=0&offset=0&rows=100>

- 请求参数：

参数名	类型	说明
id	Int	记录 ID。
isalarm	Int	报警和非报警过滤参数。默认为 0，只获取正常记录数，1：报警记录数，-1：获取正常和报警记录数。注意：此记录包含刷卡记录
offset	Int	分页获取的起始位置
rows	Int	分页获取的该页记录数。rows 的最大值是 1000，如果超过 1000，将会返回非法参数

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "recordId": 29,
      "gmtSwipeCardTime": "2019-04-04 18:52:46",
      "areaId": 160,
      "areaName": "test",
      "deviceId": "3aec7c4e58ab4a3dbd56deb98cf4bbd6",
      "deviceName": "192.168.1.64",
      "doorId": "132ccea02e5f4498be494730fba3d9a6",
      "doorName": "东门",
      "acsEventValue": 3,
      "acsEventName": "外部刷卡开门",
      "isSwipeCard": 1,
    }
  ]
}
```

```
"isOpenDoor": 1,
"inOrOut": 1,
"isAlarm": 0,
"accessBlockedReason": 0,
"cardNo": "888888",
"cardState": 1,
"userId": 3,
"userNo": "888888",
"userName": "888888",
"userOrgId": 0,
"userOrgName": "",
"plateNumber": "",
"userType": 2    },
]
}
```

说明：参考门禁记录说明

### 9. 获取门禁报警未接警和未清除的记录

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/alarmevent/notconfirmandclear>
- 说明：根据门禁设备 ID 获取实时刷卡记录
- 请求参数：无
- 返回数据格式：

```
{
  "status": 0,
  "msg": "success",
  "data": [
    {
      "eventId": 990,
      "deviceId": "12fb0b8ff5954f569fbc4fbeb62fd458",
      "subDeviceId": "214ddb7d639940099341d1023f9aabda",
      "areaName": "工厂",
      "deviceName": "192.168.2.64",
      "subDeviceName": "2 门[2]",
      "eventTime": "2022-05-24 18:28:07",
      "eventName": "非法开门",
      "orgName": "",
      "userNo": "",
      "userName": "",
      "userType": null,
      "cardNo": "0",
      "cardName": null,
    }
  ]
}
```

```

        "plateNumber": null,
        "cardState": 0,
        "cardGroup": null,
        "cardType": null,
        "isAlarm": 1,
        "alarmClass": "1",
        "accessBlockedReason": 0,
        "alarmEventTroubleId": 2873,
        "bkColor": "#ea5353",
        "tColor": "#4f4a4a",
        "photoUrl": null,
        "userTypeName": null,
        "confirmOpenDoorId": null,
        "isAlarmTop": 0,
        "userId": 0,
        "areaId": 162,
        "field1": null,
        "field2": null,
        "field3": null,
        "field4": null,
        "field5": null,
        "confirmState": 0,
        "alarmClassPriority": 254,
        "alarmClearStatus": 0,
        "alarmStatus": 0,
        "alarmRecoverStatus": 1,
        "eventType": 25
    }
]
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	实时刷卡数据数组。

● 刷卡数据说明：

字段名	类型	说明
eventId	Int	事件 ID
deviceId	String	门禁设备 ID
subDeviceId	String	门 ID
areaName	String	区域名称
deviceName	String	门禁设备名称
subDeviceName	String	门名称

eventTime	String	事件时间
eventName	String	事件名称
orgName	String	人员机构名称
userNo	String	人员工号
userName	String	人员姓名
userType	Int	人员类型，参考 2.1.1
cardNo	String	卡号
cardName	String	卡片名称
plateNumber	String	车牌号码
cardState	Int	卡片状态。1：正常，2：挂失
cardGroup	String	卡组名称
cardType	String	卡片类型
isAlarm	Int	是否报警记录。1：是，0：否
alarmClass	String	报警等级
accessBlockedReason	Int	拒绝类型。 0：成功 100：卡片不存在 101：卡片过期 102：权限无效 103：时段无效 104：节假日无效，在节假日期间不允许通行 105：密码错误，超级密码或者用户密码错误 106：没有启用超级密码 107：违反了门点互锁规则 108：违反了多卡组合规则 109：违反了 APB 规则 110：违反了刷卡次数，该用户的刷卡次数已为 0 111：违反了工作模式，控制器处于休眠模式
alarmEventTroubleId	Int	报警记录 ID
bkColor	String	记录背景颜色
tColor	String	记录字体颜色
photoUrl	String	用户头像 URL
userTypeName	String	用户类型
isAlarmTop	Int	是否是置顶
userId	Int	用户 ID
areaId	Int	所属区域 ID
alarmClassPriority	Int	报警等级优先级，数值越大，越高
alarmClearStatus	Int	清除状态，0：未清除，1：已经清除
alarmStatus	Int	接警状态，0：未接警，1：已经接警
alarmRecoverStatus	Int	报警状态是否已经恢复。0：未恢复，1：已经恢复
eventType	Int	事件值

备注：最多返回 1000 条

## 10. 获取区域内人员

- HTTP 请求方式：GET
- HTTP 请求 URL：
- <http://ip:6500/api/v1/acsrecords/areausers?areaid=0&begintime=2018-10-01+00:00:00&endtime=2018-10-30+23:59:59>
- 请求参数：

参数名	类型	说明
areaid	Int	区域 ID，获取指定某个区域的人员。默认为 0，获取所有区域人员。注意：包括下级区域
begintime	String	开始时间，可以为空。格式为：yyyy-MM-DD+HH:MM:SS。为空则不用时间筛选
endtime	String	结束时间，可以为空。格式为：yyyy-MM-DD+HH:MM:SS。为空则不用时间筛选

- 返回数据格式：

```
{
  "status": 0,
  "msg": "success",
  "data": [
    {
      "uaid": 2,
      "gmtAcsevttime": "2022-09-02 12:26:04",
      "deviceId": "12fb0b8ff5954f569fbc4fbeb62fd458",
      "deviceName": "192.168.2.64",
      "subdeviceId": "7dd55168b4d4439b9923797e87da1cd4",
      "subdeviceName": "1A002SE",
      "areaId": 161,
      "areaName": "食堂",
      "acsevValue": 22,
      "acsevName": "外部无效卡",
      "cardNo": "1",
      "cardState": 1,
      "userId": 77875,
      "userNo": "F4567",
      "userName": "冯生",
      "userOrgId": 4185,
      "userOrgName": "门禁授权用户管理",
      "gmtCreatetime": "2020-12-27 15:25:23",
      "cardname": null,
      "cardtype": "门禁锁人脸",
      "cardgroup": "普通组",
      "userType": 1897,
      "flag": 0,
      "handleState": 0,
    }
  ]
}
```

```

"deviceSource": "acs",
"field1": "",
"field2": "",
"field3": "",
"option1": 0,
"option2": 0,
"option3": 0,
"userTypeName": "职工",
"phone": "111"
},
]
}
    
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	记录数组

● 门禁记录说明：

字段名	类型	说明
uaid	Int	记录 ID
gmtAcsevertime	String	刷卡时间
areaid	Int	所属区域 ID
areaName	String	刷卡所在区域名称
deviceId	String	刷卡所在的门禁控制器 ID
deviceName	String	刷卡所在门禁控制器名称
subdeviceId	String	刷卡所在的门 ID
subdeviceName	String	刷卡所在的门名称
acsevValue	Int	门禁事件值
acsevName	String	门禁事件名称
cardNo	String	刷卡卡号
cardState	Int	卡状态。1：正常，2：挂失。
userId	Int	刷卡用户 ID
userNo	String	刷卡用户工号
userName	String	刷卡用户姓名
userOrgId	Int	刷卡用户所属机构 ID
userOrgName	String	刷卡用户所属机构名称
gmtCreatetime	String	创建时间
cardname	String	卡片名称
cardtype	String	卡片类型
cardgroup	String	卡组
userType	Int	用户类型值
flag	Int	标记

handleState	Int	处理状态
deviceSource	String	设备源
userTypeName	String	用户类型名
phone	String	电话号码
field1	String	备用
field2	String	备用
field3	String	备用
option1	Int	备用
option2	Int	备用
option3	Int	备用

## 2.1.5 门禁设备操作

### 1. 获取所有区域数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/areas>
- 请求参数：无
- 返回数据格式：

```
{
  "status": 0,
  "msg": "success",
  "data": [
    {
      "areaId": 1,
      "areaName": "公司机关",
      "areaCode": "公司机关",
      "parentId": 0
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	区域数据数组。

- 区域数据说明：

字段名	类型	说明
areaId	Int	区域 ID
areaName	String	区域名称
areaCode	String	区域编码
parentId	Int	上级区域 ID

## 2. 获取所有门禁设备数据

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/devices>
- 请求参数：无
- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "deviceId" : "3aec7c4e58ab4a3dbd56deb98cf4bbd6",
      "deviceName" : "192.168.2.64",
      "areaId" : 162,
      "areaName" : "BC",
      "deviceSn" : "8791",
      "commAddr" : "192.168.2.64",
      "isOnline" : 1,
      "commErrorMsg" : ""
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	门禁设备数据数组。

- 门禁设备数据说明：

字段名	类型	说明
deviceId	String	设备 ID
deviceName	String	设备名称
areaId	String	所属区域 ID
areaName	Int	所属区域名称
deviceSn	String	设备序列号
commAddr	String	设备 IP 地址
isOnline	int	设备在线状态。1：在线，0：离线
commErrorMsg	String	设备通信错误消息。当离线时有意义

## 3. 获取所有门数据

- HTTP 请求方式：GET

- HTTP 请求 URL : <http://ip:6500/api/v1/acs/device/doors>
- 请求参数 : 无
- 返回数据格式 :

```

{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "doorId": "6e9b3dad91b049598f142fc02c50d63d",
      "doorName": "门 1",
      "doorAddr": "1",
      "areaId": 1,
      "areaName": "东区",
      "deviceId": "0580ad81acd34ab5bbaf8946165b4a64",
      "deviceName": "办公楼大门",
    }
  ]
}
    
```

字段名	类型	说明
status	Int	接口执行的返回状态码, 详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	门数据数组。

- 门数据说明 :

字段名	类型	说明
doorId	String	门 ID
doorName	String	门名称
doorAddr	String	门地址 (保留使用)
areaId	Int	门所属区域 ID
areaName	String	门所属区域名称
deviceId	String	门所属控制器 ID
deviceName	String	门所属控制器名称

#### 4. 根据区域 ID 获取门数据

- HTTP 请求方式 : GET
- HTTP 请求 URL : <http://ip:6500/api/v1/acs/device/doors/areaid/{areaid}>
- 说明 : 根据区域 ID 获取该区域下的门
- 请求参数 :

类型	名称	类型	说明
Path	areaid	int	区域 id

- 返回数据格式 :

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "doorId": "6e9b3dad91b049598f142fc02c50d63d",
      "doorName": "门 1",
      "doorAddr": "1",
      "areaId": 1,
      "areaName": "东区",
      "deviceId": "0580ad81acd34ab5bbaf8946165b4a64",
      "deviceName": "办公楼大门",
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	门数据数组。

● 门数据说明：

字段名	类型	说明
doorId	String	门 ID
doorName	String	门名称
doorAddr	String	门地址（保留使用）
areaId	Int	门所属区域 ID
areaName	String	门所属区域名称
deviceId	String	门所属控制器 ID
deviceName	String	门所属控制器名称

## 5. 获取单个门状态

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/device/door/state/{doorid}>
- 说明：根据门 ID 获取门的状态
- 请求参数：

类型	名称	类型	说明
Path	doorid	String	要获取的门 ID，必填。

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
```

```
"isOnline" : 1,
"isOpen" : 0,
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
isOnline	Int	是否在线。1：在线，0：离线
isOpen	Int	是否打开。1：打开，0：关闭

## 6. 开门

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/device/door/open/{user}/{psw}/{doorid}>
- 说明：根据门 ID 开门
- 请求参数：

类型	名称	类型	说明
Path	user	String	登录帐号，必填。
Path	psw	String	登录密码，必填
Path	doorid	String	门 ID

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success"
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息

## 7. 关门

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/device/door/close/{user}/{psw}/{doorid}>
- 说明：根据门 ID 关门
- 请求参数：

类型	名称	类型	说明
Path	user	String	登录帐号，必填。
Path	psw	String	登录密码，必填
Path	doorid	String	门 ID

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success"
}
```

```
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息

### 8. 设置门工作模式

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/device/door/workmode/{user}/{psw}/{doorid}/{mode}>
- 说明：根据门 ID 设置门的工作模式
- 请求参数：

类型	名称	类型	说明
Path	user	String	登录帐号，必填。
Path	psw	String	登录密码，必填
Path	doorid	String	门 ID
Path	mode	Int	工作模式。0：自动，1：刷卡，2：卡+密码，3：卡或密码，4：常开，5：常闭，6：卡数据

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success"
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息

### 9. 获取实时刷卡记录

- HTTP 请求方式：GET
- HTTP 请求 URL：<http://ip:6500/api/v1/acs/device/eventcache/swipecard/{deviceids}>
- 说明：根据门禁设备 ID 获取实时刷卡记录
- 请求参数：

类型	名称	类型	说明
Path	deviceids	String	设备 ID，必填。可以填多个设备 ID，用“,”号隔开，例如：123,234

- 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "eventId" : 3311,
      "deviceId" : "3aec7c4e58ab4a3dbd56deb98cf4bbd6",

```

```

    "subDeviceId" : "ffa67848670e42869e34795471ce7b4d",
    "areaName" : "BC",
    "deviceName" : "192.168.2.64",
    "subDeviceName" : "门[1]",
    "eventTime" : "2020-02-25 11:52:14",
    "eventName" : "外部无效卡",
    "orgName" : "门禁授权用户管理",
    "userNo" : "M001124",
    "userName" : "M00112",
    "userType" : 1899,
    "cardNo" : "232376",
    "cardName" : "",
    "plateNumber" : null,
    "cardState" : 1,
    "cardGroup" : "普通组",
    "cardType" : "韦根 34",
    "isAlarm" : 1,
    "alarmClass" : "",
    "accessBlockedReason" : 102
  }
]
}

```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Json	实时刷卡数据数组。

● 刷卡数据说明：

字段名	类型	说明
eventId	Int	事件 ID
deviceId	String	门禁设备 ID
subDeviceId	String	门 ID
areaName	String	区域名称
deviceName	String	门禁设备名称
subDeviceName	String	门名称
eventTime	String	事件时间
eventName	String	事件名称
orgName	String	人员机构名称
userNo	String	人员工号
userName	String	人员姓名
userType	Int	人员类型，参考 2.1.1
cardNo	String	卡号
cardName	String	卡片名称

plateNumber	String	车牌号码
cardState	Int	卡片状态。1：正常，2：挂失
cardGroup	String	卡组名称
cardType	String	卡片类型
isAlarm	Int	是否报警记录。1：是，0：否
alarmClass	String	报警等级
accessBlockedReason	Int	拒绝类型。 0：成功 100：卡片不存在 101：卡片过期 102：权限无效 103：时段无效 104：节假日无效，在节假日期间不允许通行 105：密码错误，超级密码或者用户密码错误 106：没有启用超级密码 107：违反了门点互锁规则 108：违反了多卡组合规则 109：违反了 APB 规则 110：违反了刷卡次数，该用户的刷卡次数已为 0 111：违反了工作模式，控制器处于休眠模式

注意：实时记录在系统的缓存时间是 2 秒，将会有以下几种情况

- 1、 客户端在调用该接口时的间隔时间大于 2 秒的，可能会出现漏读实时事件的可能性。
- 2、 如果间隔时间小于 2 秒，可能会出现重复读的可能性。

为了防止漏读、重读，比较好的解决思路，读取的间隔时间少于 2 秒（比如 1 秒，500 毫秒）。如果重读，可以通过事件 ID 来处理，客户端记录上一次的读到的最大记录 ID，如果读到的记录的记录 ID 比这个 ID 大，就是最新的数据。

### 2.1.6 其他接口

#### 1. 获取用户通行记录

- HTTP 请求方式：GET
- 说明：该接口或者门禁刷卡记录、人脸识别记录、车牌识别记录
- HTTP 请求 URL：

<http://ip:6500/api/v1/>

[useraccessrecords?begintime=2018-10-01+00:00:00&endtime=2019-10-02+00:00:00&offset=0&rows=100&userno=99999](http://ip:6500/api/v1/useraccessrecords?begintime=2018-10-01+00:00:00&endtime=2019-10-02+00:00:00&offset=0&rows=100&userno=99999)

- 请求参数：

参数名	类型	说明
begintime	String	刷卡记录的开始时间。格式为：yyyy-MM-DD+HH:MM:SS。如果为空就查询当天记录

endtime	String	刷卡记录的结束时间。格式为：yyyy-MM-DD+HH:MM:SS
userno	String	人工工号，这里使用精确匹配，如果为空，不判断工号
offset	Int	分页获取的起始位置
rows	Int	分页获取的该页记录数。如果 rows=0，将不会分页，返回所有记录

● 返回数据格式：

```
{
  "status" : 0,
  "msg" : "success",
  "data" : [
    {
      "deviceName" : "192.168.1.64 门[2]",
      "deviceId" : "b8982e35420c4b33ae918be4d3533751",
      "areaId" : 160,
      "areaName" : "test",
      "evValue" : 3,
      "evName" : "外部刷卡开门",
      "userId" : 4,
      "userName" : "99999",
      "userNo" : "99999",
      "accessPassport" : "99999",
      "orgName" : null,
      "evTime" : "2019-04-04 19:04:58",
      "inOut" : 1,
      "accessType" : 1
    }
  ]
}
```

字段名	类型	说明
status	Int	接口执行的返回状态码，详情参照通用返回码对照表
msg	String	返回的消息。当 status!=0 都应该有错误信息
data	Int	通行记录数组

● 通行记录说明：

字段名	类型	说明
deviceName	String	设备名
deviceId	String	设备 ID
areaId	Int	所属区域 ID
areaName	String	所在区域名称
evValue	Int	事件值
evName	String	事件名称
userId	Int	用户 ID
userNo	String	用户工号

userName	String	用户姓名
accessPassport	String	认证凭证，门禁刷卡是卡号，人脸识别是登记号，车牌识别是车牌号
orgName	String	用户所属机构名称
evTime	String	事件时间
inOut	Int	进出方向，1：进，2：出
accessType	Int	通行类型。1：门禁，2：人脸，3：车牌识别