

商汤科技

—  
**SenseLink Enterprise 商联云智能物联平台**  
**V1.9.0-p API 操作手册 (V1.1)**

---

**API Operation Manual**

[www.sensetime.com](http://www.sensetime.com)

PRODUCT  
MANUAL



## 目录

1. 前言 .....	1
2. 概述 .....	1
3. 查看密钥 .....	1
4. 调用规则 .....	2
4.1 客户端计算数字签名 sign 的值 .....	2
4.2 客户端提交参数给服务器 .....	2
5. API 错误对应码 .....	2
6. API 使用说明 .....	4
6.1 获取公司信息 (/api/v1/company) .....	4
6.2 设备相关 API .....	6
6.2.1 获取设备列表 (/api/v1/device) .....	6
6.2.2 修改设备信息 (/api/v2/device/update) .....	10
6.2.3 删除设备 (/api/v1/device/delete/{id}) .....	12
6.3 人员相关 API .....	13
6.3.1 添加员工 (/api/v1/user) .....	13
6.3.2 修改员工 (/api/v1/user/update/{id}) .....	16
6.3.3 删除员工 (/api/v1/user/delete/{id}) .....	20
6.3.4 添加访客 (/api/v1/guest) .....	21
6.3.5 修改访客 (/api/v1/guest/update/{id}) .....	23
6.3.6 删除访客 (/api/v1/guest/delete/{id}) .....	26
6.3.7 查询人员 (/api/v1/user/list) .....	27
6.4 人员组相关 API .....	30
6.4.1 添加组 (/api/v1/group) .....	30
6.4.2 获取组列表 (/api/v1/group) .....	32

6.4.3	修改组信息 (/api/v1/group/update)	35
6.4.4	批量添加人员到组 (/api/v1/user/add/group)	36
6.4.5	从组中批量移除人员 (/api/v1/user/remove/group)	37
6.5	记录相关 API	39
6.5.1	获取识别记录 (/api/v1/record)	39
6.5.2	识别记录推送 (/api/v1/record)	43
6.6	人脸图片相关 API	47
6.6.1	检测人脸个数 (/api/v1/recognition/check)	47
6.6.2	人脸图片质量检测 (/api/v1/recognition/quality)	48
6.7	部门相关 API	50
6.7.1	获取部门列表 (/api/v2/department)	50
6.7.2	按 code 查询部门 (/api/v2/department/search/code)	52
6.7.3	按 id 查询部门 (/api/v2/department/{id})	53
6.7.4	新增部门 (/api/v2/department)	55
6.7.5	更新部门 (/api/v2/department/update)	56
6.7.6	删除部门 (/api/v2/department/delete)	57
7.	帮助与支持	58

版本	修订日期	修订者	修订内容
V1.0	2019/04/29	黄王爵	<p>新建文档，相比 V1.8.0-p 新增了如下 API：</p> <ul style="list-style-type: none"><li>• 识别记录实时推送</li><li>• 人脸图片质量检测</li><li>• 查询人员信息</li><li>• 修改组信息</li><li>• 批量添加人员到组</li><li>• 从组中批量移除人员</li></ul>
V1.1	2019/05/09	黄王爵	<ul style="list-style-type: none"><li>• 新增部门相关接口</li></ul>

# 1. 前言

此文档用于为 SenseLink 商联云智能物联平台 Enterprise V1.9.0-p 版本提供 OpenAPI 调用指引，适用于开发者。

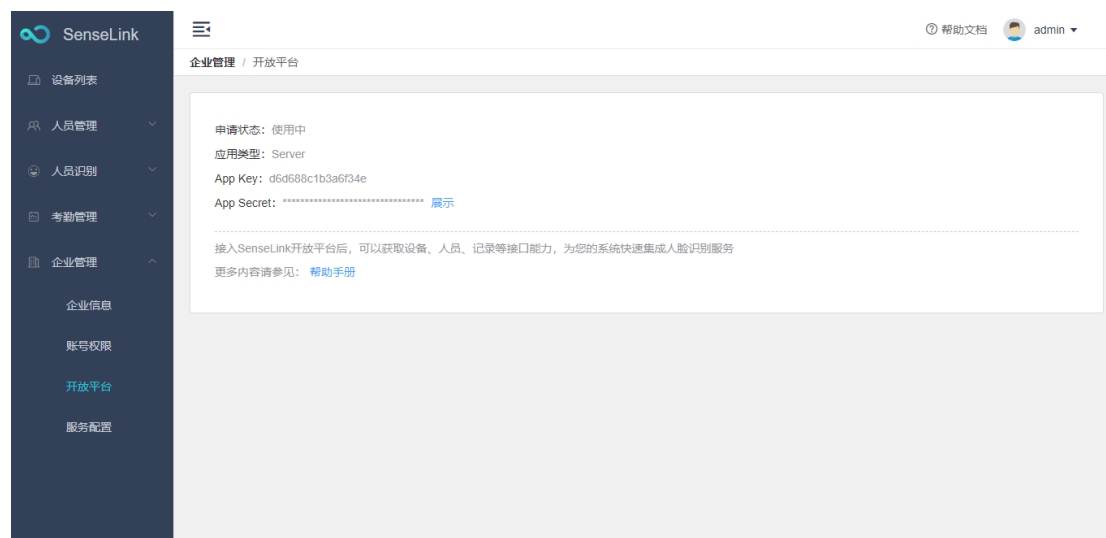
# 2. 概述

SenseLink 开放平台是基于 SenseLink 商联云智能物联平台的开放服务平台。第三方系统接入 SenseLink 后，可以获取设备、人员、记录等接口能力，快速集成 SenseLink 人脸识别能力以及多场景服务。

SenseLink Enterprise 版本中 Openapi 不限制域名或 IP 访问，所有能访问到本机的 IP 的请求均视为合法。

# 3. 查看密钥

登录 SenseLink Enterprise 云平台，点击列表的“企业管理”-“开放平台”，查看应用密钥 (AppKey 和 AppSecret)。



## 4. 调用规则

### 4.1 客户端计算数字签名 sign 的值

计算方法:  $\text{sign}=\text{md5}(\text{timestamp}\#\text{app\_secret})$ , 例如:

$\text{app\_secret}=2323\text{dsfadjewrasa}3434$

$\text{timestamp}=1477881215000$

$\text{sign}=\text{md5}(1477881215000\#2323\text{dsfadjewrasa}3434)=\text{e5ef72ef839bdf5397b4906b199b9fbf}$

### 4.2 客户端提交参数给服务器

将  $\text{sign}, \text{timestamp}, \text{app\_key}$  三个参数连同业务参数一起提交给服务器, 例如:

$\text{http://host:port/service/request.do?company\_id}=1\&\text{parent\_id}=3\&\text{app\_key}=35\text{uifanj8i30kdng}\&\text{timestamp}=1477881215000\&\text{Sign}=\text{e5ef72ef839bdf5397b4906b199b9fbf}$

其中  $\text{company\_id}, \text{parent\_id}$  为业务参数, 服务器对比当前的时间戳和客户端请求中的  $\text{timestamp}$ , 相差若大于 30 分钟则返回失败。

## 5. API 错误对应码

Code	Message	原因
10001	App Key Not Exist	Key 不存在
10002	App Expired	Key 已过期
10003	Request Expired	请求过期
10004	IP Invalid	IP 地址不在白名单
10005	Signature Failed	签名无效
20001	Company Out Of Date	该公司已过使用期限

20002	Permission Denied	无权限
30001	Param Invalid	参数不合法
30002	Similar User Exist	添加与已存在用户相似度高的用户
30003	No Face Detected In The Image	图片中未检测到人脸
30004	Missing Param	缺失参数
30005	Avatar File Illegal	图片格式不合格
50001	RPC Failed	服务器同步服务出错
50002	Internal Server Error	服务器内部错误
50003	Multi Request Error	批处理请求出错
50004	Request Method Not Allowed	请求方法无效
99999	Unknown Error	未知错误
620	Record non-existent Error	无效参数，找不到对应的记录
610	Similar record Exist	存在相同记录，如同级同名部门
650	Cannot delete unless include nothing	部门下有子部门或员工，无法删除

## 6. API 使用说明

### 6.1 获取公司信息 (/api/v1/company)

#### 1. 功能描述

获取公司信息。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/company

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

#### 5. 正常响应返回

```
{  
  "code": 200,  
  "message": "OK",  
  "data": {  
    "id": 58,  
    "name": "商汤科技（深圳）有限公司",  
    "logo": "5b238d7ec6681b0001466d5e",  
    "background": "5b238d7ec6681b0001466d5f",  
    "welcome": "欢迎来到商汤科技",  
  }  
}
```



```

"introduction": "坚持原创，让 AI 引领人类进步。",
"contact": "张三",
"createAt": "2018-05-28 17:31:32.0",
"updateAt": "2018-06-28 15:57:55.0",
"validFrom": "2018-06-15 00:00:00.0",
"validTo": "2018-07-31 00:00:00.0",
"createBy": 17,
"threshold": 0.85,
"language": "zh",
"serviceConfig": ""
}
}

```

## 6. 返回字段说明

参数名称	类型	说明
code	int	返回码
message	string	返回信息

## 7. 数据字段说明

参数名称	类型	说明
id	int	公司 ID
name	string	公司名
logo	string	logo url
background	string	欢迎背景图
welcome	string	欢迎语
introduction	string	简介

contact	string	联系人
create_at	date	公司创建日期
update_at	date	更新日期

## 6.2 设备相关 API

### 6.2.1 获取设备列表 (/api/v1/device)

#### 1. 功能描述

查询设备列表，默认顺序为 id 倒序。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/device

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
page	int	否	页号，默认为 1
size	int	否	每页数据条数，默认为 20
app_key	string	是	分配的 app_key
sign	string	是	参数签名，计算规则参考文档
timestamp	string	是	当前时间戳

#### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
}
```

```
"data": {
  "total": 2,
  "pageNum": 1,
  "size": 2,
  "pageSize": 20,
  "totalPage": 1,
  "data": [
    {
      "device_type": {
        "id": 47,
        "name": "SenseKeeper 人脸识别闸机",
        "identifier": "SKP",
        "description": "",
        "create_at": "2018-06-22 04:39:06.0",
        "update_at": "2018-06-22 04:39:06.0"
      },
      "device": {
        "id": 28,
        "sn": "SKP-e6f99b5b31792d024172498b45eaab5f",
        "name": "keeper713",
        "direction": 0,
        "location": "713",
        "ip": "",
        "status": 2,
        "groups": [
          {
            "id": 214,
            "name": "默认组",
            "type": 1
          },
          {
            "id": 215,
            "name": "默认组",
            "type": 2
          }
        ],
        "active": 1,
        "type_id": 47,
        "description": null,
        "software_version": null,
        "info": null,
        "company_id": 58,
        "group_count": 2,
        "create_at": "2018-06-22 12:45:51.0",
        "update_at": "2018-07-02 15:15:03.0"
      }
    }
  ]
}
```

```

    },
    {
      "device_type": {
        "id": 47,
        "name": "SenseKeeper 人脸识别闸机",
        "identifier": "SKP",
        "description": "",
        "create_at": "2018-06-22 04:39:06.0",
        "update_at": "2018-06-22 04:39:06.0"
      },
      "device": {
        "id": 31,
        "sn": "SKP-d00676f238a65e5e7d29b576dee360fd",
        "name": "liuyi",
        "direction": 0,
        "location": "",
        "ip": "",
        "status": 2,
        "groups": [
          {
            "id": 215,
            "name": "默认组",
            "type": 2
          },
          {
            "id": 517,
            "name": "liuyiTest",
            "type": 1
          }
        ],
        "active": 1,
        "type_id": 47,
        "description": null,
        "software_version": null,
        "info": null,
        "company_id": 58,
        "group_count": 2,
        "create_at": "2018-07-02 16:24:11.0",
        "update_at": "2018-07-02 21:15:03.0"
      }
    }
  ]
}

```

## 6. 返回字段说明

参数名称	类型	说明
<code>device_type</code>	object	设备类型
<code>device</code>	object	设备信息

## 7. device\_type 数据字段说明

参数名称	类型	说明
<code>id</code>	long	设备类型 id
<code>name</code>	string	设备类型名称
<code>identifier</code>	string	设备类型标识（简称）
<code>description</code>	string	设备类型描述
<code>create_at</code>	string	创建时间
<code>update_at</code>	string	更新时间

## 8. device 数据字段说明

参数名称	类型	说明
<code>id</code>	long	设备 id
<code>sn</code>	string	设备 sn 号
<code>name</code>	string	设备名称
<code>direction</code>	string	设备方向
<code>location</code>	string	设备位置
<code>ip</code>	string	设备 ip
<code>status</code>	int	设备在线状态
<code>groups</code>	list	设备绑定了的人员组
<code>active</code>	int	假删除标志

<code>type_id</code>	long	设备类型 id
<code>description</code>	string	设备描述
<code>software_version</code>	string	软件版本号
<code>info</code>	string	设备信息
<code>company_id</code>	long	公司 id
<code>group_count</code>	int	设备绑定的人员组数量
<code>create_at</code>	string	创建时间
<code>update_at</code>	string	更新时间

## 6.2.2 修改设备信息 (/api/v2/device/update)

### 1. 功能描述

修改设备信息。

### 2. 请求地址示例

http://HOST:PORT/api/v2/device/update

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	是否必须	说明
<code>id</code>	long	是	设备 id
<code>name</code>	string	是	设备名称, 长度限制 45
<code>location</code>	string	是	设备位置, 长度限制 255
<code>description</code>	string	否	设备描述, 长度限制 255

direction	int	是	开门方向
groupStaff	list	否	员工组，不传则清空当前设备绑定的员工组
groupVisitor	list	否	访客组，不传则清空当前设备绑定的访客组

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 37,
    "name": "设备",
    "location": "123",
    "state": 0,
    "description": "",
    "direction": 1,
    "ldid": "SPS-4c885be9d6e25a8b510cff11915c123a",
    "type_id": 4,
    "type_name": "SensePass",
    "user_group": [
      {
        "id": 1,
        "name": "默认组",
        "type": 1,
        "person_count": 8657
      }
    ],
    "guest_group": []
  }
}
```

## 6. 返回字段说明

参数名称	类型	说明
id	Long	设备 id
name	string	名称
location	string	位置
state	int	设备在线状态，0 离线，1 在线
description	string	介绍

<b>direction</b>	int	开门方向
<b>ldid</b>	string	设备 ldid 号
<b>type_id</b>	int	设备类型 id
<b>type_name</b>	string	设备类型名称
<b>user_group</b>	list	设备绑定的员工组集合
<b>guest_group</b>	list	设备绑定的访客组集合

## 6.2.3 删除设备 (/api/v1/device/delete/{id})

### 1. 功能描述

删除访客。

### 2. 请求地址示例

https://HOST:PORT/api/v1/device/delete/{id}

### 3. 请求方式

GET

### 4. 请求参数

参数名称	类型	是否必须	说明
<b>id</b>	long	是	设备 id, 类型为 path, 写在 url 中
<b>app_key</b>	string	是	分配的 app_key
<b>sign</b>	string	是	参数签名, 计算规则参考文档
<b>timestamp</b>	string	是	当前时间戳

### 5. 正常响应返回

```
{
  "code": 200,
```



```
"message": "OK"  
}
```

## 6.3 人员相关 API

### 6.3.1 添加员工 (/api/v1/user)

#### 1. 功能描述

添加员工。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/user

#### 3. 请求方式

POST: form-data

#### 4. 请求参数

参数名称	类型	是否必须	说明
avatarFile	file	否	人员头像, 要求: Content-Type: image/jpeg
force	int	否	是否强制添加, 默认 0 不强制, 1 强制
groups	array	是	人员组列表, 默认值根据 type 决定
icNumber	string	是	IC 卡号, 可以为空串, 长度限制 45
jobNumber	string	是	人员工号, 可以为空串, 长度限制 45
mobile	string	是	手机号, 可以为空串, 长度限制 20
name	string	是	人员名称
remark	string	是	备注, 可以为空串, 长度限制 255

<b>departmentId</b>	Long	否	部门 id
<b>areaCode</b>	String	否	区号
<b>birthday</b>	String	否	生日
<b>entryTime</b>	String	否	入职时间
<b>mail</b>	String	否	邮箱, 长度限制 45
<b>position</b>	String	否	职位, 长度限制 45
<b>location</b>	String	否	工位, 长度限制 45
<b>idNumber</b>	string	否	身份证号, 长度限制 45
<b>gender</b>	int	否	性别, 1: 女, 2: 男
<b>app_key</b>	string	是	分配的 app_key
<b>sign</b>	string	是	参数签名, 计算规则参考文档
<b>timestamp</b>	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 65838,
    "name": "1312313123123122",
    "avatar": "5cbe78b135060d0001750ef5",
    "mobile": "",
    "remark": "",
    "type": 1,
    "birthday": "",
    "mail": null,
    "position": null,
    "location": null,
    "groupList": [
      {
        "id": 1,
        "name": "默认组",
        "type": 1
      }
    ]
  }
}
```

```

    }
  ],
  "gender": 1,
  "ic_number": "",
  "id_number": "te9+d6aXoDe5zTCUywn9+zmX4HE1B4W",
  "job_number": "",
  "company_id": 1,
  "create_at": "2019-04-23 10:30:08",
  "update_at": "2019-04-23 10:30:08",
  "department_id": 0,
  "area_code": null,
  "entry_time": "2018-9-10",
  "country_code": null,
  "place_code": null,
  "staff_type": 0
}
}

```

## 6. 存在相似人员返回

```

{
  "code": 30002,
  "message": "Similar User Exist",
  "data": {
    "similar_user_id": 65833,
    "similar_user_name": "11111111",
    "similar_user_avatar": "5cbe7dec35060d0001750f15",
    "can_force": 0
  },
  "desc": "ex.similar-user"
}

```

## 7. 返回字段说明

参数名称	类型	说明
id	long	人员 id
name	string	人员名称
avatar	string	人员头像
mobile	string	手机号
remark	string	备注

type	int	类型, 1 员工, 2 访客
birthday	string	生日
mail	string	邮箱
position	string	职位
location	string	座位
groupList	list	人员组信息
gender	string	性别
ic_number	string	ic 卡号
id_number	string	身份证号, 对称加密之后的值
job_number	string	工号
company_id	long	公司 id
create_at	string	创建时间
update_at	string	更新时间
department_id	long	部门 id
area_code	string	手机号区号
entry_time	string	入职时间
country_code	string	国家码
place_code	string	地区码
staff_type	int	员工类型

### 6.3.2 修改员工 (/api/v1/user/update/{id})

#### 1. 功能描述

修改员工信息。

## 2. 请求地址示例

https://HOST:PORT/api/v1/user/update/{id}

## 3. 请求方式

POST: form-data

## 4. 请求参数

参数名称	类型	是否必须	说明
id	Long	是	待修改信息的员工 id, 类型为 path, 写在 url 中
avatarFile	file	否	人员头像, 要求: Content-Type: image/jpeg
force	int	否	是否强制添加, 默认 0 不强制, 1 强制
groups	array	是	人员组列表, 传空列表将不把人员添加到任何组中.
icNumber	string	是	IC 卡号, 可以为空串, 长度限制 45
idNumber	string	是	身份证号, 长度限制 45
jobNumber	string	是	人员工号, 可以为空串, 长度限制 45
mobile	string	是	手机号, 可以为空串, 长度限制 20
name	string	是	人员名称, 长度限制 45
remark	string	是	备注, 可以为空串, 长度限制 255
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳
department_id	long	否	部门 id

area_code	string	否	手机号区号
entry_time	string	否	入职时间
birthday	string	否	生日
mail	string	否	邮箱，长度限制 45
position	String	否	职位，长度限制 45
location	String	否	工位，长度限制 45

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 65838,
    "name": "11111",
    "avatar": "5cbe801835060d0001750f1b",
    "mobile": "",
    "remark": "",
    "type": 1,
    "birthday": "2018-8-1",
    "mail": null,
    "position": null,
    "location": null,
    "groupList": [
      {
        "id": 1,
        "name": "默认组",
        "type": 1
      }
    ],
    "gender": null,
    "ic_number": "",
    "id_number": "f0bZE1BF8oZ2PTC/wXDBfQ==",
    "job_number": "",
    "company_id": 1,
    "create_at": null,
    "update_at": "2019-04-23 11:01:43",
    "department_id": null,
    "area_code": null,
    "entry_time": "2018-1-1",
  }
}
```

```

        "country_code": null,
        "place_code": null,
        "staff_type": 0
    }
}

```

## 6. 返回字段说明

参数名称	类型	说明
id	long	人员 id
name	string	人员名称
avatar	string	人员头像
mobile	string	手机号
remark	string	备注
type	int	类型, 1 员工, 2 访客
birthday	string	生日
mail	string	邮箱
position	string	职位
location	string	座位
groupList	list	人员组信息
gender	string	性别
ic_number	string	ic 卡号
id_number	string	身份证号, 对称加密之后的值
job_number	string	工号
company_id	long	公司 id
create_at	string	创建时间
update_at	string	更新时间
department_id	long	部门 id

area_code	string	手机号区号
entry_time	string	入职时间
country_code	string	国家码
place_code	string	地区码
staff_type	int	员工类型

### 6.3.3 删除员工 (/api/v1/user/delete/{id})

#### 1. 功能描述

删除人员。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/user/delete/{id}

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
id	Long	是	用户 id, 类型为 path, 写在 url 中
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

#### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK"
}
```



```
}
```

## 6. 返回字段说明

参数名称	类型	说明
code	int	返回码
message	string	返回信息

### 6.3.4 添加访客 (/api/v1/guest)

#### 1. 功能描述

添加访客。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/guest

#### 3. 请求方式

POST: form-data

#### 4. 请求参数

参数名称	类型	是否必须	说明
name	string	是	人员名称, 长度限制 45
avatarFile	file	是	人员头像
mobile	string	是	手机号, 长度限制 20
remark	string	是	备注, 可以为空串,, 长度限制 255
groups	array	是	人员组列表, 默认值根据 type 决定
force	int	否	是否强制添加, 默认 0 不强制, 1 强制
idNumber	string	是	身份证号, 长度限制 45

receptionUserId	Long	是	接待人员 Id
dateTimeFrom	datetime	是	到访时间, 精确到秒, 如: 2018-07-20 12:30:45
dateTimeTo	datetime	是	到访时间, 精确到秒, 如: 2019-07-20 12:30:45
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 65833,
    "name": "11111111",
    "avatar": "5cbe739e35060d0001750edb",
    "mobile": "35123123",
    "remark": "",
    "groups": [
      {
        "id": 2,
        "name": "默认组",
        "type": 2
      }
    ],
    "ic_number": "",
    "id_number": "jwS7LXdgbXQ=",
    "company_id": 1,
    "reception_user_id": 0,
    "reception_user_name": null,
    "group_count": 1,
    "date_time": "2019-01-01 12:00:01.0",
    "date_time_from": "2019-01-01 12:00:01.0",
    "date_time_to": "2019-01-21 06:00:01.0",
    "create_at": "2019-04-23 10:08:30.0",
    "update_at": "2019-04-23 10:08:30.0"
  }
}
```

## 6. 返回字段说明

参数名称	类型	说明
id	long	人员 id
name	string	人员名称
avatar	string	人员头像
mobile	string	手机号
remark	string	备注
groups	list	人员组信息
ic_number	string	ic 卡号
id_number	string	身份证号, 对称加密之后的值
company_id	long	公司 id
create_at	string	创建时间
update_at	string	更新时间
reception_user_id	long	接待员工 id
reception_user_name	string	接待员工姓名
date_time_from	string	来访开始时间, 精确到秒级别
date_time_to	string	来访结束时间, 精确到秒级别
date_time	string	来访时间,, 精确到天级别, 已废弃, 不建议使用

### 6.3.5 修改访客 (/api/v1/guest/update/{id})

#### 1. 功能描述

修改一个访客信息。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/guest/update/{id}

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	是否必须	说明
name	string	是	人员名称, 长度限制 45
avatarFile	file	是	人员头像
mobile	string	是	手机号, 长度限制 20
remark	string	是	备注, 可以为空串, 长度限制 255
groups	array	是	人员组列表, 默认值根据 type 决定
force	int	否	是否强制添加, 默认 0 不强制, 1 强制
idNumber	string	是	身份证号, 长度限制 45
receptionUserId	Long	是	接待人员 Id
dateTimeFrom	datetime	是	到访时间, 精确到秒, 如: 2018-07-20 12:30:45
dateTimeTo	datetime	是	到访时间, 精确到秒, 如: 2019-07-20 12:30:45
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳
id	Long	是	访客 id

### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
}
```

```

"data": {
  "id": 65833,
  "name": "11111111",
  "avatar": "5cbe7dec35060d0001750f15",
  "mobile": "35123123",
  "remark": "",
  "groups": [
    {
      "id": 2,
      "name": "默认组",
      "type": 2
    }
  ],
  "ic_number": "",
  "id_number": "c6y20tRtvM4=",
  "company_id": 1,
  "reception_user_id": 0,
  "reception_user_name": null,
  "group_count": 1,
  "date_time": "2019-01-01 12:00:01.0",
  "date_time_from": "2019-01-01 12:00:01.0",
  "date_time_to": "2019-01-21 06:00:01.0",
  "create_at": "2019-04-23 10:08:30.0",
  "update_at": "2019-04-23 10:52:28.0"
}
}

```

## 6. 返回字段说明

参数名称	类型	说明
code	int	返回码
message	string	返回信息

## 7. 数据字段说明

参数名称	类型	说明
id	int	访客 ID
name	string	访客名
avatar	string	logo url

mobile	string	访客手机
remark	string	备注
groups	string	人员组
lc_number	string	lc 卡号
id_number	string	身份证号, 对称加密之后的值
company_id	string	公司 id
reception_user_id	string	接待人员 id
reception_user_name	string	接待人员姓名
create_at	date	创建时间
update_at	date	更新时间
date_time_from	string	来访开始时间, 精确到秒级别
date_time_to	string	来访结束时间, 精确到秒级别
date_time	string	来访时间, 精确到天级别, 已废弃, 不建议使用

### 6.3.6 删除访客 (/api/v1/guest/delete/{id})

#### 1. 功能描述

删除一个访客。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/guest/delete/{id}

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
id	Long	是	访客 id, 类型为 path, 写在 url 中
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

#### 5. 正常响应返回

```
{  
  "code": 200,  
  "message": "OK"  
}
```

#### 6. 返回字段说明

参数名称	类型	说明
code	int	返回码
message	string	返回信息

### 6.3.7 查询人员 (/api/v1/user/list)

#### 1. 功能描述

根据特定字段查询员工或访客, 或获取所有人员列表。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/user/list

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
name	string	否	人员名称
jobNumber	string	否	人员工号
mobile	string	否	手机号
mail	String	否	邮箱
icNumber	string	否	lc 卡号
id	Long	否	id
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": [
    {
      "id": 65838,
      "name": "11111",
      "avatar": "5cbe801835060d0001750f1b",
      "mobile": "",
      "remark": "",
      "type": 1,
      "birthday": "2018-08-01",
      "mail": null,
      "position": null,
      "location": null,
      "groupList": [
        {
          "id": 1,
          "name": "默认组",
          "type": 1
        }
      ],
      "gender": 1,
    }
  ]
}
```



```

        "ic_number": "",
        "id_number": "f0bZE1BF8oZ2PTC/wXDBfQ==",
        "job_number": "",
        "company_id": 1,
        "create_at": "2019-04-23 10:30:08.0",
        "update_at": "2019-04-23 03:01:44.0",
        "department_id": 0,
        "area_code": null,
        "entry_time": "2018-01-01",
        "country_code": null,
        "place_code": null,
        "staff_type": 0
    }
}
]
}

```

## 6. 返回字段说明

参数名称	类型	说明
id	long	人员 id
name	string	人员名称
avatar	string	人员头像
mobile	string	手机号
remark	string	备注
type	int	类型, 1 员工, 2 访客
birthday	string	生日
mail	string	邮箱
position	string	职位
location	string	座位
groupList	list	人员组, 此接口如果返回结果只有一个人则包含人员组信息, 大于 1 个人不返回人员组信息
gender	string	性别
ic_number	string	ic 卡号

id_number	string	身份证号，对称加密之后的值
job_number	string	工号
company_id	long	公司 id
create_at	string	创建时间
update_at	string	更新时间
department_id	long	部门 id
area_code	string	手机号区号
entry_time	string	入职时间
country_code	string	国家码
place_code	string	地区码
staff_type	int	员工类型

## 6.4 人员组相关 API

### 6.4.1 添加组 (/api/v1/group)

#### 1. 功能描述

添加一个组。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/group

#### 3. 请求方式

POST: form-data

#### 4. 请求参数

参数名称	类型	是否必须	说明
------	----	------	----

<b>name</b>	string	是	组名称, 长度限制 45
<b>type</b>	int	是	组类型。(员工组是 1, 访客组是 2)
<b>app_key</b>	string	是	分配的 app_key
<b>sign</b>	string	是	参数签名, 计算规则参考文档
<b>timestamp</b>	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 616,
    "name": "小霸王",
    "type": 1,
    "company_id": 58,
    "create_at": "2018-08-07 18:37:32",
    "update_at": "2018-08-07 18:37:32"
  }
}
```

## 6. 返回字段说明

参数名称	类型	说明
<b>code</b>	int	返回码
<b>message</b>	string	返回信息

## 7. 数据字段说明

参数名称	类型	说明
<b>id</b>	Long	组 id
<b>name</b>	string	组名称

type	int	组类型
company_id	Long	公司 id
create_at	string	创建时间
update_at	string	最近一次修改时间

## 6.4.2 获取组列表 (/api/v1/group)

### 1. 功能描述

查询人员组列表，默认顺序为 id 倒序。

### 2. 请求地址示例

https://HOST:PORT/api/v1/group

### 3. 请求方式

GET

### 4. 请求参数

参数名称	类型	是否必须	说明
page	int	否	页号，默认为 1
size	int	否	每页数据条数（默认为 20）
type	int	否	人员组类型，员工为 1，访客为 2，默认为 0 返回所有
app_key	string	是	分配的 app_key
sign	string	是	参数签名，计算规则参考文档
timestamp	string	是	当前时间戳

### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "total": 5,
    "pageNum": 1,
    "size": 5,
    "pageSize": 20,
    "totalPage": 1,
    "data": [
      {
        "id": 214,
        "name": "默认组",
        "type": 1,
        "user_number": 1327,
        "company_id": 58,
        "create_at": "2018-05-28 17:31:32.0",
        "update_at": "2018-05-28 17:31:32.0"
      },
      {
        "id": 215,
        "name": "默认组",
        "type": 2,
        "user_number": 0,
        "company_id": 58,
        "create_at": "2018-05-28 17:31:32.0",
        "update_at": "2018-05-28 17:31:32.0"
      },
      {
        "id": 226,
        "name": "123",
        "type": 1,
```

```

        "user_number": 1543,
        "company_id": 58,
        "create_at": "2018-05-30 14:50:17.0",
        "update_at": "2018-05-30 14:50:17.0"
    },
    {
        "id": 231,
        "name": "ccc",
        "type": 1,
        "user_number": 4,
        "company_id": 58,
        "create_at": "2018-06-04 21:52:45.0",
        "update_at": "2018-06-04 21:52:53.0"
    },
    {
        "id": 517,
        "name": "liuyiTest",
        "type": 1,
        "user_number": 2,
        "company_id": 58,
        "create_at": "2018-07-02 21:14:41.0",
        "update_at": "2018-07-02 21:14:41.0"
    }
]
}
}

```

## 6. 返回字段说明

参数名称	类型	说明
code	int	返回码
message	string	返回信息

## 7. 数据字段说明

参数名称	类型	说明
id	int	人员组 ID
name	string	人员组名称
company_id	int	公司 ID
create_at	int	添加时间
update_at	int	更新时间
type	int	人员组类型

### 6.4.3 修改组信息 (/api/v1/group/update)

#### 1. 功能描述

修改组信息。

#### 2. 请求地址示例

http://HOST:PORT/api/v1/group/update

#### 3. 请求方式

POST: form-data

#### 4. 请求参数

参数名称	类型	是否必须	说明
id	Long	是	人员组 id, 长度限制 45
name	String	是	员工组名。备注：不能为空字符串，例如：“ ”
type	Integer	是	组类型。1-员工组，2-访客组。

app_key	string	是	分配的 app_key
sign	string	是	参数签名，计算规则参考文档
timestamp	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK"
}
```

## 6. 返回 code 说明

code	说明	备注
200	请求成功	
30001	参数不合法	例如：groupId 传 null
320	参数无效	例如：groupId 不存在

## 6.4.4 批量添加人员到组 (/api/v1/user/add/group)

### 1. 功能描述

批量添加人员到组。

### 2. 请求地址示例

http://HOST:PORT/api/v1/user/add/group

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	是否必须	说明
------	----	------	----



groupId	long	是	人员组 id
userIds	list	是	批量添加的人员 id。每次操作的最大值为 2000 人
app_key	string	是	分配的 app_key
sign	string	是	参数签名，计算规则参考文档
timestamp	string	是	当前时间戳

#### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK"
}
```

#### 6. 返回 code 说明

code	说明	备注
200	请求成功	
620	参数不存在	例如：groupId 不存在等
30001	参数不合法	例如：groupId 为空等
50001	同步 go 层失败	例如：调 go 层绑定人员到组失败

### 6.4.5 从组中批量移除人员 (/api/v1/user/remove/group)

#### 1. 功能描述

从组中批量移除人员。

#### 2. 请求地址示例

http://HOST:PORT/api/v1/user/remove/group

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	是否必须	说明
groupId	long	是	员工组 id
userIds	list	是	批量移除的人员 id。每次操作的最大值为 2000 人
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK"
}
```

### 6. 返回 code 说明

code	说明	备注
200	请求成功	
620	参数不存在	例如: groupId 不存在等
30001	参数不合法	例如: groupId 为空等
50001	同步 go 层失败	例如: 调 go 层绑定人员到组失败

## 6.5 记录相关 API

### 6.5.1 获取识别记录 (/api/v1/record)

#### 1. 功能描述

查询识别记录，默认时间顺序为从近到远。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/record

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
startRecordId	long	否	起始记录 id
endRecordId	long	否	结束记录 id
deviceSN	string	否	设备号，默认空串
type	int	否	类型:{1: 员工, 2: 访客, 3: 陌生人}, 默认 0: 全部
userId	Long	否	人员 id, 默认 0
startTime	timestamp	否	起始日期，默认为当日，时间戳，精确到秒
endTime	timestamp	否	结束日期，默认为当日，时间戳，精确到秒
username	String	否	人员姓名,默认空串
page	int	否	页号 (默认为 1)

size	int	否	每页数据条数（默认为 20）
app_key	String	是	分配的 app_key
sign	String	是	参数签名，计算规则参考文档
timestamp	String	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "total": 14,
    "pageNum": 1,
    "size": 3,
    "pageSize": 3,
    "totalPage": 5,
    "data": [
      {
        "id": 535344,
        "userId": 55591,
        "name": "张三",
        "avatar": "5c94925a35060d00016daed6",
        "icNumber": "13123",
        "idNumber": "wFrud7ZifzCIXFCY7lOiiT+qIeDqCG/U",
        "jobNumber": "183719",
        "receptionUserId": 0,
        "receptionUserName": "",
        "groupId": 0,
        "groupName": null,
        "groups": [
          {
            "id": 1,
            "name": "默认组",
            "type": 1
          }
        ],
        "sn": "SPS-9f5fda14f29e81255215d87c4fa01a94",
        "deviceName": "hb",
        "direction": 0,
        "signAvatar": "5cd2f25b2afec20001d42675",
        "signBgAvatar": "5cd2f25b2afec20001d42676",
        "signDate": "2019-05-08",
        "signTime": 1557328420,

```

```
"type": 1,
"mobile": "18682217152",
"remark": null,
"entryMode": 1
},
{
  "id": 535343,
  "userId": 55591,
  "name": "李四",
  "avatar": "5c94925a35060d00016daed6",
  "icNumber": "13123",
  "idNumber": "wFrud7ZifzCIXFCY7l0iit+qIeDqCG/U",
  "jobNumber": "183719",
  "receptionUserId": 0,
  "receptionUserName": "",
  "groupId": 0,
  "groupName": null,
  "groups": [
    {
      "id": 1,
      "name": "默认组",
      "type": 1
    }
  ],
  "sn": "SPS-9f5fda14f29e81255215d87c4fa01a94",
  "deviceName": "hb",
  "direction": 0,
  "signAvatar": "5cd2f2592afec20001d42673",
  "signBgAvatar": "5cd2f2592afec20001d42674",
  "signDate": "2019-05-08",
  "signTime": 1557328419,
  "type": 1,
  "mobile": "18682217152",
  "remark": null,
  "entryMode": 1
},
{
  "id": 535342,
  "userId": 0,
  "name": "",
  "avatar": null,
  "icNumber": null,
  "idNumber": null,
  "jobNumber": null,
  "receptionUserId": null,
  "receptionUserName": null,
```

```

        "groupId": 0,
        "groupName": "",
        "groups": null,
        "sn": "SPS-9f5fda14f29e81255215d87c4fa01a94",
        "deviceName": null,
        "direction": null,
        "signAvatar": "5cd2f2562afec20001d42671",
        "signBgAvatar": "5cd2f2562afec20001d42672",
        "signDate": "2019-05-08",
        "signTime": 1557328415,
        "type": 3,
        "mobile": null,
        "remark": null,
        "entryMode": 1
    }
}
]
}
}

```

## 6. 返回字段说明

参数名称	类型	说明
id	long	识别记录 id
userId	long	被识别人 id
name	string	被识别人姓名
avatar	string	被识别人头像
mobile	string	被识别人手机号
icNumber	string	被识别人 icNumber
idNumber	string	被识别人 idNumber (需解密)
jobNumber	string	被识别人工号
receptionUserId	long	被识别人的接待人员 id (访客独有)
receptionUserName	string	被识别人的接待人员姓名 (访客独有)
groupId		无意义, 已废弃
groupName		无意义, 已废弃

groups	list	被识别人员的人员组信息
sn	string	上传这条识别记录的设备 LDID 号
deviceName	string	设备名称
direction	int	通行方向
signAvatar	string	识别小图
signBgAvatar	string	识别大图
signDate	string	识别时间，字符串格式
signTime	long	识别时间，时间戳格式
type	int	识别类型
entryMode	int	核验模式 {1: 刷脸, 2: 二维码, 3: 刷卡, 4: 刷脸+刷卡}

## 6.5.2 识别记录推送 (/websocket/record)

### 1. 功能描述

实时接收所有设备的识别记录推送 (WebSocket)。

### 2. 请求地址示例

ws://HOST:9000/websocket/record/{app\_key}/{timestamp}/{sign}

### 3. 请求方式

POST

### 4. 请求参数

参数名称	类型	是否必须	说明
app_key	String	是	分配的 app_key
sign	String	是	参数签名，计算规则参考文档

<b>timestamp</b>	String	是	当前时间戳
------------------	--------	---	-------

## 5. 响应参数

参数名称	类型	说明
id	long	识别记录 id
userId	long	被识别人 id
name	string	被识别人姓名
avatar	string	被识别人头像
mobile	string	被识别人手机号
icNumber	string	被识别人 icNumber
idNumber	string	被识别人 idNumber (需解密)
jobNumber	string	被识别人工号
receptionUserId	long	被识别人的接待人员 id (访客独有)
receptionUserName	string	被识别人的接待人员姓名 (访客独有)
groupId		无意义, 已废弃
groupName		无意义, 已废弃
groups	list	被识别人的人员组信息
sn	string	上传这条识别记录的设备 ldid 号
deviceName	string	设备名称
direction	int	通行方向
signAvatar	string	识别小图
signBgAvatar	string	识别大图
signDate	string	识别时间, 字符串格式
signTime	long	识别时间, 时间戳格式
type	int	识别类型



entryMode	int	核验模式 {1: 刷脸, 2: 二维码, 3: 刷卡, 4: 刷脸+刷卡}
-----------	-----	---------------------------------------

## 6.正常响应返回

连接成功:

```
{
  "code": 20000,
  "message": " Connected!",
  "desc": "Connected!"
}
```

连接失败:

```
{
  "code": 20002,
  "message": "Permission Denied!",
  "desc": "app key not exist"
}
```

心跳检测:

```
{
  "code": 10000,
  "desc": "heart beat",
  "message": "heart beat"
}
```

消息推送体:

```
{
  "code": 30000,
  "data": {
    "companyId": 1,
```

```
"deviceName": "liuyi2",
"deviceSN": "SPS-23da60ccda8f91a865a7472c1c5d9df5",
"direction": 1,
"groupId": 0,
"groupName": "",
"id": 2,
"originAvatar": "",
"receptionUserId": 0,
"receptionUserName": "",
"signAvatar": "5c80cb55bac9be0001d4e93a",
"signBgAvatar": "5c80cb55bac9be0001d4e93b",
"signTime": 1551944533,
"userId": 0,
"userName": "",
"userType": 3,
"verifyScore": 0.0
},
"desc": "",
"message": "push record"
}
```

## 7. 返回字段说明

Code	说明
200	成功
498	失败

## 6.6 人脸图片相关 API

### 6.6.1 检测人脸个数 (/api/v1/recognition/check)

#### 1. 功能描述

识别照片中人脸的个数。

#### 2. 请求地址示例

https://HOST:PORT/api/v1/recognition/check

#### 3. 请求方式

POST: form-data

#### 4. 请求参数

参数名称	类型	是否必须	说明
face_avatar	file	是	人脸图片, 要求: Content-Type: image/jpeg

#### 5. 正常响应返回

```
{  
  "code": 200,  
  "message": "OK",  
  "face_number": 4  
}
```

#### 6. 失败返回

```
{  
  "code": 30003,  
  "message": "No Face Detected In The Image",  
  "desc": "rpc.no-face-detect",  
}
```

#### 7. 返回字段说明

code	类型	说明
code	int	返回码
message	string	返回信息
desc	string	错误的返回描述
data	int	人脸数量

## 6.6.2 人脸图片质量检测 (/api/v1/recognition/quality)

### 1. 功能描述

检测人脸图片的质量。

### 2. 请求地址示例

https://HOST:PORT/api/v1/recognition/quality

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	是否必须	说明
face_avatar	file	是	人脸图片

### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "qa_code": 1030301,
    "qa_desc": "the face pose pitch score is [22.19]-float",
    "qa_msg": "BAN:the face pose pitch is substandard"
  }
}
```

}

## 6. 返回字段说明

参数名称	类型	说明
qa_code	int	返回码
qa_msg	string	返回信息
qa_desc	string	返回描述

## 7. 返回 code 说明

返回码	返回信息	返回描述
0	PASS	通过
101	BAN:face count is over the max	人脸数量超过最大数量限制
10201	BAN:image size wrong,two small	图片太小
10301	BAN:the alignment score is substandard	检测评分过低
1030201	BAN:the face rect's pixels is too low	人脸框像素过低
1030301	BAN:the face pose pitch is substandard	俯仰角过大
1030302	BAN:the face pose yaw is substandard	偏转角过大
1030303	BAN:the face pose roll is substandard	侧倾角过大
10401	BAN:the face quality{distance to center} is substandard	人脸到中心距离不标准
10402	BAN:the face quality{face size} is substandard	人脸尺寸占比过低 (人脸矩形面积/图像面积)
10403	BAN:the face quality{face cover} is substandard	面部遮挡过多

10404	BAN:the face quality{brightness} is substandard	明亮度不达标
10405	BAN:the face quality{sharpness} is substandard	模糊度不达标
10406	BAN:the face quality{mouth opened} is substandard	嘴部过度张开
10407	BAN:the face quality{face missing} is substandard	面部缺失过多

## 6.7 部门相关 API

### 6.7.1 获取部门列表 (/api/v2/department)

#### 1. 功能描述

获取部门列表信息

#### 2. 请求地址示例

https://HOST:PORT/api/v1/department

#### 3. 请求方式

GET

#### 4. 请求参数

参数名称	类型	是否必须	说明
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

#### 5. 正常响应返回

```
{
```

```
"code": 200,
"message": "OK",
"data": {"id": 0,
  "name": "深圳市商汤科技有限公司",
  "children": [
    {
      "id": 1,
      "name": "行政部",
      "children": [
        {
          "id": 3,
          "name": "云服务",
          "children": [
            {}
          ],
          "parent_id": 1,
          "leader_id": 370108,
          "leader_name": "luhan",
          "user_num": 15
        },
        {
          "id": 56,
          "name": "啥的",
          "children": [],
          "parent_id": 1,
          "leader_id": null,
          "leader_name": null,
          "user_num": 0
        }
      ],
      "parent_id": 0,
      "leader_id": 372345,
```

```

        "leader_name": "kkk",
        "user_num": 17
    }
]
}

```

## 6. 返回字段说明

参数名称	类型	说明
id	long	部门 id
name	string	部门名称
parent_id	long	上级部门 id,0 表示公司
leader_id	long	部门长 id
leader_name	string	部门长姓名
code	string	部门编码
user_num	int	部门人数
children	list	子部门列表

## 6.7.2 按 code 查询部门 (/api/v2/department/search/code)

### 1. 功能描述

通过部门 code 获取部门信息

### 2. 请求地址示例

https://HOST:PORT/api/v1/department/search/code

### 3. 请求方式

GET

### 4. 请求参数



参数名称	类型	能否为空	说明
code	string	N	部门 code, 长度限制 45
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 11,
    "name": "步步步步",
    "code": "003",
    "parent_id": 0,
    "leader_id": null
  }
}
```

## 6. 返回字段说明

参数名称	类型	说明
id	long	部门 id
name	string	部门名称
parent_id	long	上级部门 id, 0 表示公司
leader_id	long	部门长 id
code	string	部门编码

## 6.7.3 按 id 查询部门 (/api/v2/department/{id})

### 1. 功能描述

通过部门 id 获取部门信息

## 2. 请求地址示例

https://HOST:PORT/api/v1/department/{id}

## 3. 请求方式

GET

## 4. 请求参数

参数名称	类型	能否为空	说明
id	long	N	部门 id
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 2,
    "name": "工程院",
    "code": null,
    "parent_id": 0,
    "leader_id": 55505
  }
}
```

## 6. 返回字段说明

参数名称	类型	说明
id	long	部门 id
name	string	部门名称
parent_id	long	上级部门 id,0 表示公司
leader_id	long	部门长 id

code	string	部门编码
------	--------	------

## 6.7.4 新增部门 (/api/v2/department)

### 1. 功能描述

新增部门接口

### 2. 请求地址示例

https://HOST:PORT/api/v1/department

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	能否为空	说明
parentId	long	N	上级部门 id
name	string	N	名称, 长度限制 50
leaderId	long	Y	主管 id
code	string	Y	部门编码, 长度限制 45
app_key	string	N	分配的 app_key
sign	string	N	参数签名, 计算规则参考文档
timestamp	string	N	当前时间戳

### 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 2,
    "name": "工程院",
    "code": null,
  }
}
```

```

        "parent_id": 0,
        "leader_id": 55505
    }
}

```

## 6. 返回字段说明

参数名称	类型	说明
id	long	部门 id
name	string	部门名称
parent_id	long	上级部门 id,0 表示公司
leader_id	long	部门长 id
code	string	部门编码

## 6.7.5 更新部门 (/api/v2/department/update)

### 1. 功能描述

更新部门接口

### 2. 请求地址示例

https://HOST:PORT/api/v1/department/update

### 3. 请求方式

POST: form-data

### 4. 请求参数

参数名称	类型	能否为空	说明
id	long	N	部门 id
parentId	long	N	上级部门 id
name	string	N	名称, 长度限制 50

<b>leaderId</b>	long	Y	主管 id
<b>code</b>	string	Y	部门编码，长度限制 45
<b>app_key</b>	string	N	分配的 app_key
<b>sign</b>	string	N	参数签名，计算规则参考文档
<b>timestamp</b>	string	N	当前时间戳

## 5. 正常响应返回

```
{
  "code": 200,
  "message": "OK",
  "data": {
    "id": 2,
    "name": "工程院",
    "code": null,
    "parent_id": 0,
    "leader_id": 55505
  }
}
```

## 6. 返回字段说明

参数名称	类型	说明
<b>id</b>	long	部门 id
<b>name</b>	string	部门名称
<b>parent_id</b>	long	上级部门 id,0 表示公司
<b>leader_id</b>	long	部门长 id
<b>code</b>	string	部门编码

## 6.7.6 删除部门 (/api/v2/department/delete)

### 1. 功能描述

删除部门

## 2. 请求地址示例

https://HOST:PORT/api/v1/department/delete

## 3. 请求方式

GET

## 4. 请求参数

参数名称	类型	是否必须	说明
id	long	是	部门 id
app_key	string	是	分配的 app_key
sign	string	是	参数签名, 计算规则参考文档
timestamp	string	是	当前时间戳

## 5. 正常响应返回

```
{  
  "code": 200,  
  "message": "OK"  
}
```

# 7. 帮助与支持

如在产品使用过程中有任何问题, 请随时联系我们:

[bi-service@sensetime.com](mailto:bi-service@sensetime.com)



微信订阅号



微博

咨询电话 4009005986  
(周一至周五 9:30 - 18:00)  
商务合作 [business@sensetime.com](mailto:business@sensetime.com)  
媒体垂询 [media@sensetime.com](mailto:media@sensetime.com)  
市场合作 [mkt@sensetime.com](mailto:mkt@sensetime.com)