

目录

目录.....	2
接口列表.....	7
一、 账号.....	7
1. 登录.....	7
(1) Uri.....	7
(2) 方法.....	7
(3) 数据类型.....	7
(4) 参数说明.....	7
(5) 返回值.....	7
(6) 示例.....	8
2. 注册.....	8
(1) Uri.....	8
(2) 方法.....	8
(3) 数据类型.....	8
(4) 参数说明.....	8
(5) 返回值.....	9
(6) 示例.....	9
3. 修改.....	9
(1) Uri.....	9
(2) 方法.....	9
(3) 数据类型.....	9
(4) 参数说明.....	9
(5) 返回值.....	10
(6) 示例.....	10
4. 删除.....	10
(1) Uri.....	10
(2) 方法.....	11
(3) 数据类型.....	11
(4) 参数说明.....	11
(6) 示例.....	11
二、 人脸照片/场景照.....	12
1. 上传人员底库照片.....	12
(1) Uri.....	12
(2) 方法.....	12
(3) 数据类型.....	12
(4) 返回值.....	12
(5) 示例.....	12
2. 下载人员底库照片.....	13
(1) Uri.....	13
(2) 方法.....	13

(3) 数据类型.....	13
(4) 参数说明.....	13
(5) 返回值.....	13
(6) 示例.....	13
3. 删除人员底库照片.....	14
(1) Uri.....	14
(2) 方法.....	14
(3) 数据类型.....	14
(4) 参数说明.....	14
(5) 返回值.....	14
(6) 示例.....	14
三、 人员底库.....	15
1. 添加.....	15
(1) Uri.....	15
(2) 方法.....	15
(3) 数据类型.....	15
(4) 参数说明.....	15
(5) 返回值.....	16
(6) 示例.....	16
2. 删除.....	16
(1) Uri.....	16
(2) 方法.....	16
(3) 数据类型.....	16
(4) 参数说明.....	16
(5) 返回值.....	17
(6) 示例.....	17
3. 查询.....	17
(1) Uri.....	17
(2) 方法.....	17
(3) 数据类型.....	17
(4) 参数说明.....	18
(5) 返回值.....	18
(6) 示例.....	18
四、 人员.....	19
1. 添加.....	19
(1) Uri.....	19
(2) 方法.....	19
(3) 数据类型.....	19
(4) 参数说明.....	20
(5) 返回值.....	20
(6) 示例.....	20
2. 删除.....	21
(1) Uri.....	21
(2) 方法.....	21

(3) 数据类型.....	21
(4) 参数说明.....	21
(5) 返回值.....	21
(6) 示例.....	21
3. 修改.....	22
(1) Uri.....	22
(2) 方法.....	22
(3) 数据类型.....	22
(4) 参数说明.....	22
(5) 返回值.....	23
(6) 示例.....	23
4. 查询.....	23
(1) Uri.....	23
(2) 方法.....	23
(3) 数据类型.....	24
(4) 参数说明.....	24
(5) 返回值.....	24
(6) 示例.....	25
五、 摄像头.....	26
1. 添加.....	26
(1) Uri.....	26
(2) 方法.....	26
(3) 数据类型.....	26
(4) 参数说明.....	26
(5) 返回值.....	26
(6) 示例.....	27
2. 删除.....	27
(1) Uri.....	27
(2) 方法.....	27
(3) 数据类型.....	27
(4) 参数说明.....	27
(5) 返回值.....	28
(6) 示例.....	28
3. 修改.....	28
(1) Uri.....	28
(2) 方法.....	28
(3) 数据类型.....	28
(4) 参数说明.....	28
(5) 返回值.....	29
(6) 示例.....	29
4. 查询.....	29
(1) Uri.....	29
(2) 方法.....	30
(3) 数据类型.....	30

(4) 参数说明.....	30
(5) 返回值.....	30
(6) 示例.....	31
5. 搜索摄像头.....	31
(1) Uri.....	31
(2) 方法.....	32
(3) 数据类型.....	32
(4) 参数说明.....	32
(5) 返回值.....	32
(6) 示例.....	32
6. 查询摄像头 rtsp.....	33
(1) Uri.....	33
(2) 方法.....	33
(3) 数据类型.....	33
(4) 参数说明.....	33
(5) 返回值.....	33
(6) 示例.....	34
六、 任务.....	34
1. 添加.....	34
(1) Uri.....	34
(2) 方法.....	34
(3) 数据类型.....	35
(4) 参数说明.....	35
(5) 返回值.....	35
(6) 示例.....	35
2. 删除.....	36
(1) Uri.....	36
(2) 方法.....	36
(3) 数据类型.....	36
(4) 参数说明.....	36
(5) 返回值.....	36
(6) 示例.....	37
3. 修改.....	37
(1) Uri.....	37
(2) 方法.....	37
(3) 数据类型.....	37
(4) 参数说明.....	37
(5) 返回值.....	38
(6) 示例.....	38
4. 查询.....	39
(1) Uri.....	39
(2) 方法.....	39
(3) 数据类型.....	39
(4) 参数说明.....	39

(5) 返回值.....	39
(6) 示例.....	40
5. 启动任务.....	41
(1) Uri.....	41
(2) 方法.....	42
(3) 数据类型.....	42
(4) 参数说明.....	42
(5) 返回值.....	42
(6) 示例.....	42
6. 停止任务.....	43
(1) Uri.....	43
(2) 方法.....	43
(3) 数据类型.....	43
(4) 参数说明.....	43
(5) 返回值.....	43
(6) 示例.....	43
七、 系统配置.....	45
1. 更新配置.....	45
(1) Uri.....	45
(2) 方法.....	45
(3) 数据类型.....	45
(4) 参数说明.....	45
(5) 返回值.....	46
(6) 示例.....	46
2. 获取配置信息.....	46
(1) Uri.....	46
(2) 方法.....	46
(3) 数据类型.....	46
(4) 参数说明.....	47
(5) 返回值.....	47
(6) 示例.....	47

接口列表

url 格式: <http://ip:port/uri>

一、账号

1. 登录

(1) Uri

/Login

(2) 方法

Post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
username	账号名	Y	字符串	
password	账号密码	Y	字符串	MD5 加密

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
user_id	用户 id	Y	整型	
session	会话 session	N	字符串	成 功 返 回 session
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "username": "rose",
  "password": "md5-xxxx"
}
```

返回值：

```
{
  "code": 0,
  "user_id": 1,
  "session": "",
  "msg": ""
}
```

2. 注册

（1）Uri

/RegisterUser

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
name	账号名	Y	字符串	
password	账户密码	Y	字符串	
remark	备注信息	N	字符串	

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "name": "rose",
  "password": "md5-xxxx",
  "remark": ""
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

3. 修改

（1）Uri

/UpdateUser

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
----	----	----	----	----

session	会话	Y	字符串	登 录 返 回 session
user_id	账号 id	Y	整型	
name	账号名	N	字符串	
old_password	账户原密码	N	字符串	md5 加密
password	账号新密码	N	字符串	md5 加密
remark	备注信息	N	字符串	

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "user_id": 1,
  "name": "rose",
  "old_password": "md5-xxxx",
  "password": "md5-xxxx",
  "remark": "rose"
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

4. 删除

（1）Uri

/DeleteUser

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
user_id	账号 id	Y	整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "user_id": 1
}
```

返回值:

```
{
  "code": 0,
  "msg": ""
}
```

二、人脸照片/场景照

1. 上传人员底库照片

(1) Uri

/UploadPersonPicture

(2) 方法

multi-post

(3) 数据类型

file

(4) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
pic_url	照片地址	N	字符串	成功返回对应地址
msg	错误描述	Y	字符串	

(5) 示例

返回值:

```
{
  "code": 0,
  "pic_url": "",
  "msg": ""
}
```

2. 下载人员底库照片

(1) Uri

/DownloadPersonPicture

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
pic_url	照片地址	Y	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
data	数据	N	字符串	成功返回
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{  
  "session": "session",  
  "pic_url": ""  
}
```

返回值:

```
{
```

```
"code": 0,
"data": "",
"msg": ""
}
```

3. 删除人员底库照片

(1) Uri

/DeletePicture

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
pic_url	照片地址	Y	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
```

```
        "pic_url": ""
    }
    返回值:
    {
        "code": 0,
        "msg": ""
    }
```

三、人员底库

1. 添加

(1) Uri

/AddLibrary

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
name	底库名	Y	字符串	
type	底库类型	Y	整型	迭 代 定 义 类 型, 默认值为 0
remark	备注信息	N	字符串	

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "name": "lib1",
  "type": 0,
  "remark": ""
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

2. 删除

（1）Uri

/DeleteLibrary

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
----	----	----	----	----

session	会话	Y	字符串	登 录 返 回 session
lib_id	账号名	Y	整型	

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "lib_id": 1
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

3. 查询

（1）Uri

/QueryLibraryList

（2）方法

post

（3）数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
type	底库类型	N	整型	
name	底库名	N	字符串	模糊匹配

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	数据信息	N	Json 对象	成功时才有 此对象
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
total_count	总数	Y	整型	获取匹配数 据总数
items	数据列表	Y	Json 数组	
lib_id	底库 id	Y	整型	
name	底库名	Y	字符串	
type	底库类型	Y	整型	
person_count	人员总数	Y	整型	
remark	备注信息	Y	字符串	
create_time	创建时间	Y	字符串	格 式 : yyyy-mm-dd hh:mm:ss
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{  
  "session": "session",  
  "page_no": 1,  
  "page_size": 1,  
  "type": 1,
```

```
        "name": "lib1"
    }
    返回值:
    {
        "code": 0,
        "info":
        {
            "page_no": 1,
            "page_size": 1,
            "total_count":10,
            "items":[{"lib_id":1,
                "name": "",
                "type":1,
                "person_count": 10,
                "remark": " ",
                "create_time": "2001-01-01 00: 00:00 "
            }
            ...]
        }
        "msg": ""
    }
}
```

四、人员

1. 添加

(1) Uri

/AddPerson

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
lib_id	底库 id	Y	整型	
name	姓名	Y	字符串	
sex	性别	N	整型	0: 未知 1: 男 2: 女
pic_url	照片地址	Y	字符串	
card_no	证件号	N	字符串	
phone	手机号	N	字符串	
category	人员分类	N	整型	0: 白名单 1: 黑名单
remark	备注信息	N	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "lib_id": 1,
  "name": "",
  "lib_id": 1,
  "pic_url": "",
  "card_no": "",
  "phone": "",
  "category": 0,
  "remark": ""
}
```

返回值:

```
{
  "code": 0,
```

```
    "msg": ""
}
```

2. 删除

(1) Uri

/DeletePerson

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
lib_id	底库 id	Y	整型	
person_id	人员 id	Y	长整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数：

```
{
  "session": "session",
  "lib_id": 1,
```

```
        "person_id": 1
    }
    返回值:
    {
        "code": 0,
        "msg": ""
    }
```

3. 修改

(1) Uri

/UpdatePerson

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
lib_id	底库 id	Y	整型	
person_id	人员 id	Y	长整型	
name	姓名	N	字符串	
sex	性别	N	整型	0: 未知 1: 男 2: 女
pic_url	照片地址	N	字符串	
card_no	证件号	N	字符串	
phone	手机号	N	字符串	
category	人员分类	N	整型	0: 白名单 1: 黑名单
remark	备注信息	N	字符串	

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "lib_id": 1,
  "person_id": 1,
  "name": "",
  "lib_id": 1,
  "pic_url": "",
  "card_no": "",
  "phone": "",
  "category": 0,
  "remark": ""
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

4. 查询

（1）Uri

/QueryPersonList

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
name	底库名	N	字符串	模糊匹配
card_no	证件号	N	字符串	精确匹配
category	人员分类	N	整型	
phone	手机号	N	字符串	模糊匹配

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	数据信息	N	Json 对象	成 功 时 才 有 此对象
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
total_count	总数	Y	整型	获 取 匹 配 数 据总数
items	数据列表	Y	Json 数组	
lib_id	底库 id	Y	整型	
lib_name	底库名	Y	字符串	
person_id	人员 id	Y	长整型	
name	姓名	Y	字符串	
pinyin	姓名拼音	Y	字符串	
sex	性别	Y	整型	
card_no	证件号	Y	字符串	
phone	手机号	Y	字符串	
category	人员分类	Y	整型	
pic_url	照片地址	Y	字符串	
remark	备注信息	Y	字符串	
create_time	创建时间	Y	字符串	格 式： yyyy-mm-dd

				hh:mm:ss
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "page_no": 1,
  "page_size": 1,
  "name": "",
  "card_no": "",
  "category": 1,
  "phone": ""
}
```

返回值:

```
{
  "code": 0,
  "info": {
    "page_no": 1,
    "page_size": 1,
    "total_count": 10,
    "items": [{
      "lib_id": 1,
      "lib_name": "",
      "person_id": 1,
      "name": "",
      "pinyin": "",
      "sex": 1,
      "card_no": "",
      "phone": "",
      "category": 1,
      "pic_url": " ",
      "remark": " ",
      "create_time": "2001-01-01 00: 00:00 "
    }
    ...]
  }
  "msg": ""
}
```

五、摄像头

1. 添加

(1) Uri

/AddCamera

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
name	摄像头标识	Y	字符串	
rtsp	rtsp取流地址	Y	字符串	
type	分类	N	整型	0: 默认 后续可能对 接第三方盒子
remark	备注信息	N	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "name": "",
  "rtsp": "",
  "type": 0,
  "remark": ""
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

2. 删除

（1）Uri

/DeleteCamera

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
camera_id	摄像头 id	Y	整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

```
输入参数：
{
    "session": "session",
    "camera_id": 1
}
返回值：
{
    "code": 0,
    "msg": ""
}
```

3. 修改

(1) Uri

```
/UpdateCamera
```

(2) 方法

```
post
```

(3) 数据类型

```
Application/json
```

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session

camera_id	摄像头 id	Y	整型	
name	摄像头标识	N	字符串	
rtsp	rtsp取流地址	N	字符串	
type	分类	N	整型	0: 默认 后续可能对接第三方盒子
remark	备注信息	N	字符串	

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "camera_id": 1,
  "name": "",
  "rtsp": "",
  "type": 0,
  "remark": ""
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

4. 查询

（1）Uri

/QueryCameraList

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
name	摄像头标识	N	字符串	模糊匹配
type	摄像头类型	N	整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	数据信息	N	Json 对象	成功时才有 此对象
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
total_count	总数	Y	整型	获取匹配数 据总数
items	数据列表	Y	Json 数组	
camera_id	摄像头 id	Y	整型	
name	摄像头标识	Y	字符串	
rtsp	取流地址	Y	字符串	
type	类型	Y	整型	
online	在线状态	Y	整型	0: 离线 1: 在线
remark	备注信息	Y	字符串	
create_time	创建时间	Y	字符串	格 式 : yyyy-mm-dd hh:mm:ss
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "page_no": 1,
  "page_size": 1,
  "name": "",
  "type": 1
}
```

返回值:

```
{
  "code": 0,
  "info": {
    "page_no": 1,
    "page_size": 1,
    "total_count": 10,
    "items": [{
      "camera_id": 1,
      "name": "",
      "rtsp": "",
      "type": 1,
      "state": 1,
      "remark": " ",
      "create_time": "2001-01-01 00: 00:00 "
    }
    ...]
  }
  "msg": ""
}
```

5. 搜索摄像头

(1) Uri

/SearchCamera

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	数据信息	N	Json 对象	成 功 时 采 用 此对象
total_count	总数	Y	整型	获取匹配数 据总数
items	数据列表	Y	Json 数组	
ip	摄像头 ip	Y	字符串	
manufacturer	厂商	Y	字符串	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session"
}
```

返回值：

```
{
  "code": 0,
  "info":
  {
```



```
        "total_count":10,
        "items":[{"ip":"","manufacturer": " "
        }
        ...]
    }
    "msg": ""
}
```

6. 查询摄像头 rtsp

(1) Uri

/QueryCameraRtsp

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
ip	摄像头 ip	Y	字符串	
user	摄像头登录 用户名	N	字符串	
password	摄像头登录 密码	N	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	

info	数据信息	N	Json 对象	成功时采用此对象
rtsp	取流地址	Y	字符串	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "ip": "",
  "user": "",
  "password": "",
}
```

返回值：

```
{
  "code": 0,
  "info": {
    "rtsp": ""
  },
  "msg": ""
}
```

六、任务

1. 添加

（1）Uri

/AddTask

（2）方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
name	任务标识	Y	字符串	
type	分类	N	整型	0: 普通 1: 点名 3: 签到
camera_list1	入口摄像头	Y	字符串	摄像头 id 列表, 以逗号间隔
camera_list2	出口摄像头	N	字符串	普通任务只需一组摄像头
lib_list	底库列表	Y	字符串	人员分组列表, 逗号间隔
interval	检测间隔	Y	整型	
threshold	阈值	Y	浮点型	>= 此值为一个人
plan	计划模板	N	字符串	
remark	备注信息	N	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{  
  "session": "session",  
  "name": "",
```

```
"type": 0,
"camera_list1": "1,2",
"camera_list2": "3,4",
"lib_list": "1,2",
"interval": 3000,
"threshold": 0.8,
"plan": "",
"remark": ""
}
返回值:
{
  "code": 0,
  "msg": ""
}
```

2. 删除

(1) Uri

/DeleteTask

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
task_id	任务 id	Y	整型	

(5) 返回值

字段	定义	必填	类型	备注
----	----	----	----	----

code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "task_id": 1
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

3. 修改

（1）Uri

/UpdateTask

（2）方法

post

（3）数据类型

Application/json

（4）参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
task_id	任务 id	Y	整型	
name	任务标识	N	字符串	
type	分类	N	整型	0：普通 1：点名

				3: 签到
camera_list1	入口摄像头	Y	字符串	摄像头 id 列表, 以逗号间隔
camera_list2	出口摄像头	N	字符串	普通任务只需一组摄像头
lib_list	底库列表	Y	字符串	人员分组列表, 逗号间隔
interval	检测间隔	N	整型	
threshold	阈值	N	浮点型	>= 此值为一个人
plan	计划模板	N	字符串	
remark	备注信息	N	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "task_id": 1,
  "name": "",
  "type": 0,
  "camera_list1": "",
  "camera_list2": "",
  "lib_list": "1,2",
  "interval": 3000,
  "threshold": 0.8,
  "plan": "",
  "remark": ""
}
```

返回值:

```
{
  "code": 0,
  "msg": ""
}
```

}

4. 查询

(1) Uri

/QueryTaskList

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
name	任务标识	N	字符串	模糊匹配
type	任务类型	N	整型	
state	任务状态	N	整型	0: 关闭 1: 开启

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	数据信息	N	Json 对象	成功时才有 此对象
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
total_count	总数	Y	整型	获取匹配数 据总数
items	数据列表	Y	Json 数组	

task_id	任务 id	Y	整型	
name	任务标识	Y	字符串	
type	类型	Y	整型	
camera_list1	入口摄像头	Y	Json 数组	
camera_id	摄像头 id	Y	整型	
name	摄像头标识	Y	字符串	
state	打开状态	Y	整型	0: 失败 1: 成功
camera_list2	出口摄像头	Y	Json 数组	
camera_id	摄像头 id	Y	整型	
name	摄像头标识	Y	字符串	
state	打开状态	Y	整型	0: 失败 1: 成功
lib_list	底库列表	Y	Json 数组	
lib_id	底库 id	Y	整型	
name	底库名	Y	整型	
interval	识别间隔	Y	整型	
threshold	阈值	Y	整型	
state	状态	Y	整型	
plan	计划任务	Y	字符串	
create_time	创建时间	Y	字符串	格 式 : yyyy-mm-dd hh:mm:ss
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
    "session": "session",
    "page_no": 1,
    "page_size": 1,
    "name": "",
    "type": 1,
    "state": 1
}
```

返回值:

```
{
    "code": 0,
    "info":
    {
        "page_no": 1,
```



```

    "page_size": 1,
    "total_count": 10,
    "items": [{
        "task_id": 1,
        "task_name": "",
        "type": 1,
        "camera_list1": [{
            "camera_id": 1,
            "name": "camera1",
            "state": 1
        }
        ...
    ],
    "camera_list2": [{
        "camera_id": 1,
        "name": "camera1",
        "state": 1
    }
    ...
    ],
    "lib_list": [{
        "lib_id": 1,
        "name": "test"
    }
    ...
    ],
    "interval": 1,
    "threshold": 1,
    "state": 1,
    "plan": "",
    "remark": " ",
    "create_time": "2001-01-01 00: 00:00 "
    }
    ...]
}
"msg": ""
}

```

5. 启动任务

(1) Uri

/StartTask

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
task_id	任务 id	Y	整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "task_id": 1
}
```

返回值:

```
{
  "code": 0,
  "msg": ""
}
```

6. 停止任务

(1) Uri

/StopTask

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
task_id	任务 id	Y	整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "task_id": 1
}
```

返回值:

```
{
  "code": 0,
```

```
"msg": ""  
}
```

7. 1:1 比对

(1) Uri

/Compare1V1

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
first_pic	人员照片地址	Y	字符串	
second_pic	人员照片地址	Y	字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
similarity	人员相似度	Y	浮点型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数：

```
{
  "session": "session",
  "first_pic": "",
  "second_pic": ""
}
返回值:
{
  "code": 0,
  "similarity": 0.82,
  "msg": ""
}
```

七、系统配置

1. 更新配置

(1) Uri

/UpdateSystemConfig

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
name	配置标识	Y	字符串	各个配置标识不一样 存储服务： StorageServer
config	配置属性	Y	json 字符串	

remark	备注信息	N	字符串	
--------	------	---	-----	--

（5）返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

（6）示例

输入参数：

```
{
  "session": "session",
  "name": "StorageServer",
  "config": "{\"ip\":\"127.0.0.1\",\"port\":10009}",
  "remark": ""
}
```

返回值：

```
{
  "code": 0,
  "msg": ""
}
```

2. 获取配置信息

（1）Uri

/GetSystemConfig

（2）方法

post

（3）数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
name	配置标识	Y	字符串	各个配置标识不一样 存储服务： StorageServer
config	配置属性	Y	json 字符串	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	配置信息	Y	json 对象	成功返回此对象
name	配置标识	Y	字符串	
config	配置详情	Y	Json 对象	
msg	错误描述	Y	字符串	

(6) 示例

输入参数：

```
{  
  "session": "session",  
  "name": "StorageServer"  
}
```

返回值：

```
{  
  "code": 0,  
  "Info":  
  {  
    "name": "StorageServer",  
    "config": "{ \"ip\": \"127.0.0.1\", \"port\": 10009 }"  
  },  
  "msg": ""  
}
```

八、历史记录

1. 任务抓拍历史记录查询

(1) Uri

/QueryTaskCaptureRecord

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
task_id	任务 id	N	整型	
camera_id	摄像头 id	N	整型	
lib_id	底库 id	N	整型	
person_name	任务姓名	N	字符串	模糊匹配
card_no	人员证件号	N	字符串	精确匹配
classify	识别结果	N	整型	0：陌生人 1：白名单
similarity	识别相似度	N	浮点型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
info	数据信息	N	Json 对象	成功时才有

				此对象
page_no	当前页数	Y	整型	
page_size	每页个数	Y	整型	
total_count	总数	Y	整型	获取匹配数据总数
items	数据列表	Y	Json 数组	
record_id	记录 id	Y	长整型	
task_id	任务 id	Y	整型	
task_name	任务标识	Y	字符串	
camera_id	摄像头 id	Y	整型	
camera_name	摄像头标识	Y	字符串	
lib_id	底库 id	Y	整型	
lib_name	底库标识	Y	字符串	
person_id	人员 id	Y	整型	
person_name	人员标识	Y	字符串	
sex	人员性别	Y	整型	
category	人员类型	Y	整型	
card_no	人员证件号	Y	字符串	
classify	识别分类	Y	整型	0: 陌生人 1: 白名单
similarity	识别相似度	Y	浮点型	
pic_url	人员底库照 url	Y	字符串	
capture_url	抓配照 url	Y	字符串	
capture_time	抓拍时间	Y	字符串	格式 : yyyy-mm-dd hh:mm:ss
create_time	创建时间	Y	字符串	格式 : yyyy-mm-dd hh:mm:ss
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{
  "session": "session",
  "page_no": 1,
  "page_size": 1,
  "task_id": 1,
  "camera_id": 1,
```

```

        "lib_id": 1,
        "person_name": "",
        "card_no": "",
        "classify": 1,
        "similarity": 0.8
    }
返回值:
{
    "code": 0,
    "info":
    {
        "page_no": 1,
        "page_size": 1,
        "total_count": 10,
        "items": [{
            "camera_id": 1,
            "task_id": 1,
            "task_name": "",
            "camera_id": 1,
            "camera_name": "",
            "lib_id": 1,
            "lib_name": "",
            "person_id": 1,
            "person_name": "",
            "sex": 1,
            "category": 1,
            "card_no": "",
            "classify": 1,
            "similarity": 0.8,
            "pic_url": "",
            "capture_url": "",
            "capture_time": "2001-01-01 00: 00:00 ",
            "create_time": "2001-01-01 00: 00:00 "
        }
        ...]
    }
    "msg": ""
}

```

2. 删除

(1) Uri

/DeleteTaskCaptureRecord

(2) 方法

post

(3) 数据类型

Application/json

(4) 参数说明

字段	定义	必填	类型	备注
session	会话	Y	字符串	登 录 返 回 session
record_id	记录 id	Y	长整型	

(5) 返回值

字段	定义	必填	类型	备注
code	错误码	Y	整型	
msg	错误描述	Y	字符串	

(6) 示例

输入参数:

```
{  
  "session": "session",  
  "record_id": 1  
}
```

返回值:

```
{  
  "code": 0,  
}
```

```
    "msg": ""  
  }
```