人脸相关 API 接口

目　　录

1.1 单张人脸注册

接口描述：

接口 url	http://${ip}:${port}/api/faces
请求方式	POST
请求参数格式	JSON
接口描述	单张人脸注册入库

请求参数：

参数	类型	描述
image	string	人脸图片，base64 编码，需带有`data:image/jpg;base64`头部信息。
description	string	人脸文件的描述（不超过 255 字节， UTF-8 编码）。
imageMd5	string	基于图片的 md5 值的唯一表示，参数为空时，入库会自动生成，不为空时，可根据规则生成，确保唯一，最大 32 字节。
name	string	人员姓名，可为空（最大 128 字节）。
gender	string	人员性别，可为空（最大 16 字节）。

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
imageId	string	入库的图片原图标识（24 字节），用于调取原图，仅在白名单有效。
faceToken	string	入库成功后，该人脸的唯一标识（24 字节），用于后续分组操作。
rect	object	人脸在入库图片上的位置坐标
|—left	int	人脸框左边界坐标
|—top	int	人脸框上边界坐标
|—right	int	人脸框右边界坐标
|—bottom	int	人脸库下边界坐标

错误码：

状 态码	错误码	消息	说明
400	104200	FACE_NOT_FOUND	未检测到人脸
400	104201	BAD_QUALITY:<reason>	人脸未通过质量判断，reason 为未通过项目
400	104204	FACE_ALREADY_EXIST	人脸已经存在

人脸入库图片说明：

- 图片类型：Jpg、Jpeg、PNG、BMP；
- 图片大小：<=4MB；

- 图片尺寸：最大<=4096*2160，最小>=100*100；
- 图片质量：入库时会做人脸质量检测，如质量低，则无法入库，检测包括人脸角度、清晰度、对比度等。

入库图片的 imageMd5 生成参考：

1. 先将文件生成 32 位的 md5 码，按每两位生成长度为 16 位的 16 进制数；
2. 将每个 16 进制数按 ASCII 码，进行 Base64 转换；
3. 最后将 base64 编码里的字符`+`替换位`-`、`/`替换为`_`，结果即为 imageId。

参考示例：

请求示例：

```
{
    "image":"...",
    "description": "8.png",
    "imageMd5":"",
    "name": "",
    "gender":""
}
```

响应示例：

```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "id": 6,
        "imageId": "_jreV7nZpLFJmPrm",
        "faceToken": "Y_h5kisqfVwqEYup",
        "rect": {
            "left": 104,
            "top": 52,
            "right": 174,
            "bottom": 162
        }
    }
}
```

1.2 查询人脸库所有人脸信息

接口描述：

接口 url	http://${ip}:${port}/api/faces
请求方式	GET
请求参数格式	url
接口描述	获取所有人脸库所有人脸信息

请求参数：

参数	类型	描述
pageOffset	int	列表结果偏移量，从 0 开始
pageSize	int	返回结果一页的最大数量

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
totalFaces	int	总人脸数。
face	array	人脸列表。
|—imageId	string	入库时图片的唯一标识，用于调取原图。
|—faceToken	string	入库时人脸的唯一标识，用于分组操作，如没有入库成功的图片该参数为空。
description	string	入库时照片文件的描述
name	string	人员姓名
gender	string	人员性别

响应示例：


```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "totalFaces": 2,
        "face":[{
            "id": 1,
            "imageId": "bMmYp2ntudEF8qgk",
            "faceToken": "XaAim2HQpSqI_sYt",
            "description": "3.png",
            "name": "",
            "gender": ""
        },{
            "id": 2,
            "imageId": "EJXGsaSZTvzahFhv",
            "faceToken": "MKkQPdlqyJ5XHXe7",
            "description": "5.png",
            "name": "",
            "gender": ""
        }]
    }
}
```


1.3 单张人脸删除

接口描述:

接口 url	http://${ip}:${port}/api/faces/{faceToken}
请求方式	DELETE
请求参数格式	无
接口描述	删除某个人脸，将会删除该 faceToken 对应的

	人脸图和分组相关绑定关系，但对应的 ImageId、原图与描述仍然存在。

响应示例：


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


1.4 单张人脸原图删除

接口描述:

接口 url	http://${ip}:${port}/api/faces/image/{imageId}
请求方式	DELETE
请求参数格式	无
接口描述	删除某个入库图片，将会删除该 imageId 对应的原始图片、人脸图、与分组绑定关系。此接口将删除所有关于该图片的信息，包括对应的 faceToken。

调用后后台执行流程

1. 如有绑定的 faceToken，则删除该 faceToken 对应的人脸图，分组绑定信息及该 faceToken；
2. 删除 ImageId 对应的原图与描述；
3. 删除 ImageId

响应示例：


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


1.5 单张人脸查询

接口描述：

接口 url	http://${ip}:${port}/api/faces/faceToken/{faceToken}
请求方式	GET
请求参数格式	无
接口描述	获取某人脸信息与抠图

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
imageId	string	入库的图片标识，24 字节。
description	string	入库的图片描述，255 字节。

name	string	人员姓名
gender	string	人员性别
rect	object	人脸在图片上面的框坐标
|—left	int	左边界坐标
|—top	int	上边界坐标
|—right	int	右边界坐标
|—bottom	int	下边界坐标
faceGroupList	array	绑定的分组列表
|—groupId	string	分组 Id
image	string	base64 编码的人脸图片

响应示例：


```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "imageId": "5VUg64l2G4kzm8t-",
        "description": "3_best.jpg",
        "name": "",
        "gender": "female",
        "rect": {
            "left": 355,
            "top": 389,
            "right": 455,
            "bottom": 289
        },
        "faceGroupList": [{
            "groupId": "default"
        }],
        "image": "..."
    }
}
```


1.6 获取某个原始图片信息

接口描述：

接口 url	http://${ip}:${port}/api/faces/image/{imageId}
请求方式	GET
请求参数格式	无
接口描述	获取某个原始图片信息

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
image	string	Base64 编码后的原始图片，24 字节。
faceToken	array	一张原始图可能对应多个人脸，目前只支持一个人脸
|—	string	faceToken

响应示例：


```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "image": "...",
        "faceToken": ["XaAim2HQpSqI_sYt"]
    }
}
```


1.6 初始化人脸库

接口描述：

接口 url	http://${ip}:${port}/api/faces/facesLib
请求方式	DELETE
请求参数格式	无
接口描述	初始化人脸库，将删除所有入库的原始图片和人脸图片，并删除所有关联关系

响应示例：


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


2.1 查询所有人脸分组信息

接口描述：

接口 url	http://${ip}:${port}/api/faceGroups
请求方式	GET
请求参数格式	无
接口描述	获取人脸分组列表

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
faceGroupList	array	人脸分组列表
|—groupId	string	人脸分组 ID
|—groupType	string	人脸分组类型

|—groupName	string	人脸分组名称
|—groupFaces	int	人脸分组关联的人脸总数量

响应示例：


```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "faceGroupList": [{
            "groupId": "default",
            "groupType": "",
            "groupScore": 80,
            "groupName": "",
            "groupFaces": 5
        }, {
            "groupId": "1",
            "groupType": "1",
            "groupScore": 80,
            "groupName": "",
            "groupFaces": 5
        }
    }
}
```


2.2 创建人脸分组

接口描述:

接口 url	http://${ip}:${port}/api/faceGroups
请求方式	POST
请求参数格式	JSON
接口描述	创建人脸分组，最多支持 64 个分组

请求参数：

参数	类型	描述	是否必填
groupId	string	分组 ID，分组的唯一标识，分组的删除、修改、查询都通过 groupId 操作，最大 32 字节	是
groupType	string	分组类型，最大 128 字节	否
groupName	string	分组名称，最大 128 字节	是
groupScore	int	分组阈值，与该分组比对人脸时，大于这个值认为比对成功，默认 80	否

错误码：

状态码	错误码	消息	描述
400	104310	GROUP_ID_CONFLICT	分组 ID 已经存在
400	104300	GROUP_NAME_CONFICT	分组名称已经存在
400	104302	GROUP_OVER_LIMIT	分组数超上限

请求示例：


```
{
```

```
    "groupId": "1",
    "groupName": "员工",
    "groupScore": 80,
    "groupType": "白名单"
}
```

响应示例：

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


2.3 删除指定人脸分组

接口描述:

接口 url	http://${ip}:${port}/api/faceGroups/{groupId}
请求方式	DELETE
请求参数格式	JSON
接口描述	删除指定人脸分组

请求参数：

参数	类型	描述	是否必填
deleteAllFace	int	1. 不删除组中人脸 2. 删除组中人脸，同时解除组中人脸与其他分组的关联关系 3. 清除组中人脸，并保留当前分组 4. 删除只绑定到该分组的人脸	是

参数各个值详细流程：

value	流程
1	解绑分组的人脸；删除分组
2	解绑分组中人脸与其他分组的绑定；删除分组中的人脸；删除分组
3	解绑分组中人脸；如果这个人脸只绑定当前分组，仅删除对应的人脸图；不删除原始图；保留分组
4	解绑分组中的人脸；如果这个人脸只绑定当前分组，删除人脸图和原始图；删除分组

错误码：

状态码	错误码	消息	描述
400	104304	GROUP_BEUSED	分组正被使用，不能删除
400	104311	GROUP_ID_NOT_EXIST	分组 ID 不存在

请求示例：

```
{
    "deleteAllFace": 4
}
```

响应示例：


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


2.4 查询某一人脸分组的所有人脸

接口描述：

接口 url	http://${ip}:${port}/api/faceGroups/{groupId}
请求方式	GET
请求参数格式	无
接口描述	获取某分组绑定列表

请求参数：

参数	类型	描述	是否必填
groupId	string	人脸分组 ID	是
pageOffset	int	列表结果偏移量，从 0 开始	是
pageSize	int	返回结果一页的最大数量	是

请求示例：

http://192.168.1.79/api/faceGroups/default?pageOffset=0&pageSize=10

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
totalFaces	int	分组内的人脸数
face	array	分组内的人脸列表
|—imageId	string	原图唯一标识，24 字节
|—faceToken	string	人脸唯一标识，24 字节
|—description	string	原图描述，最大 255 字节
|—name	string	人员姓名
|—gender	string	人员性别

错误码：

状态码	错误码	消息	描述
400	104311	GROUP_ID_NOT_EXIST	分组 ID 不存在

响应示例：


```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "totalFaces": 2,
        "face": [{
```

```
            "imageId": "5VUg64l2G4kzm8t-",
            "faceToken": "RPq4hO5VOg1KFqv3",
            "description": "3_best.jpg",
            "name": "",
            "gender": "female"
        }, {
            "imageId": "HQ3QimOMfvxRmXb8",
            "faceToken": "ep5JmObasYrSMlib",
            "description": "4.png",
            "name": "",
            "gender": "female"
        }]
    }
}
```


2.5 绑定人脸到分组

接口描述:

接口 url	http://${ip}:${port}/api/faceGroups/binding/{faceToken}
请求方式	POST
请求参数格式	JSON
接口描述	绑定人脸到分组

请求参数：

参数	类型	描述	是否必填
faceGroupList	array	需要绑定的人脸分组列表，允许绑定多个分组	是

请求示例：


```
{
    "faceGroupList":["default"]
}
```


响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
totalCnt	int	绑定的分组数
successCnt	int	绑定成功的分组数
failCnt	int	绑定失败的分组数
failInfo	array	绑定失败的具体信息
|—groupId	string	分组 ID
|—errorMsg	string	错误信息

返回示例：


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

```
    "msg": "SUCCESS",
    "data": {
        "totalCnt": 3,
        "successCnt": 1,
        "failCnt": 2,
        "failInfo": [{
            "groupId": "123",
            "errorMsg": "GROUP_ID_NOT_EXIST"
        }, {
            "groupId": "456",
            "errorMsg": "GROUP_ID_NOT_EXIST"
        }]
    }
}
```


2.6 解绑人脸与分组关系

接口描述:

接口 url	http://${ip}:${port}/api/faceGroups/unbind/{faceToken}
请求方式	DELETE
请求参数格式	JSON
接口描述	解绑人脸与分组关系

请求参数：

参数	类型	描述	是否必填
faceGroupList	array	需要解绑的人脸分组列表，允许多个分组	是

请求示例：


```
{
    "faceGroupList":["default"]
}
```


响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
totalCnt	int	待解绑的分组数
successCnt	int	解绑成功的分组数
failCnt	int	解绑失败的分组数
failInfo	array	解绑失败的具体信息
|—groupId	string	分组 ID
|—errorMsg	string	错误信息

返回示例：


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

```
    "msg": "SUCCESS",
    "data": {
        "totalCnt": 3,
        "successCnt": 1,
        "failCnt": 2,
        "failInfo": [{
            "groupId": "123",
            "errorMsg": "GROUP_ID_NOT_EXIST"
        }, {
            "groupId": "456",
            "errorMsg": "GROUP_ID_NOT_EXIST"
        }]
    }
}
```


2.7 修改人脸分组信息

接口描述:

接口 url	http://${ip}:${port}/api/faceGroups
请求方式	PUT
请求参数格式	JSON
接口描述	修改人脸分组

请求参数：

参数	类型	描述	是否必填
groupId	string	分组 ID，分组的唯一标识(删除、修改、查询)，最大 32 字节	是
groupType	string	分组类型，最大 128 字节	否
groupName	string	分组名称，最大 128 字节	否
groupScore	int	分组比对阈值，0-100	否

错误码：

状态码	错误码	消息	描述
400	104311	GROUP_ID_NOT_EXIST	分组 ID 不存在

请求示例：


```
{
    "groupId": "1",
    "groupName": "1222",
    "groupScore": 80,
    "groupType": "1"
}
```


响应示例：


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

}

3.1 人脸 1:1

接口描述:

接口 url	http://${ip}:${port}/api/face/compare
请求方式	POST
请求参数格式	JSON
接口描述	人脸 1:1

请求参数：

参数	类型	描述	是否必填
image1	string	人脸图片 1，base64 编码的字符串	是
image2	string	人脸图片 2，base64 编码的字符串	是
livenessEnabled	bool	活体使能开关	是

请求示例：

```
{
    "image1":"data:image/png;base64...",
    "image2":"data:image/png;base64...",
    "livenessEnabled":false
}
```

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
compareScore	float	1:1 相似度
detectionResult1	object	人脸图片 1 检测结果
|—left	int	脸部框左边界坐标
|—top	int	脸部框上边界坐标
|—right	int	脸部框右边界坐标
|—bottom	int	脸部框下边界坐标
|—roll	int	人脸旋转角度
|—yaw	int	人脸水平角度
|—pitch	int	人脸垂直角度
|—blurness	float	图像模糊度(0~100)，越小越模糊
|—livenessScore	float	活体检测结果(0~100)
detectionResult2	object	人脸图片 2 检测结果
|—left	int	脸部框左边界坐标
|—top	int	脸部框上边界坐标
|—right	int	脸部框右边界坐标
|—bottom	int	脸部框下边界坐标
|—roll	int	人脸旋转角度

|—yaw	int	人脸水平角度
|—pitch	int	人脸垂直角度
|—blurness	float	图像模糊度(0~100)，越小越模糊
|—livenessScore	float	活体检测结果(0~100)

响应示例：


```
{
    "code": 0,
    "msg": "SUCCESS",
    "data": {
        "compareScore": 58.0530891418457,
        "detectionResult1": {
            "left": 723,
            "top": 257,
            "right": 985,
            "bottom": 525,
            "roll": 0,
            "yaw": 0,
            "pitch": 0,
            "blurness": 22.477077484130859
        },
        "detectionResult2": {
            "left": 833,
            "top": 217,
            "right": 1034,
            "bottom": 408,
            "roll": 0,
            "yaw": 0,
            "pitch": 0,
            "blurness": 10.853910446166992
        }
    }
}
```


3.2 人脸 1:N

接口描述:

接口 url	http://${ip}:${port}/api/face/search
请求方式	POST
请求参数格式	JSON
接口描述	人脸 1:N

请求参数：

参数	类型	描述	是否必填
image	string	人脸图片，base64 编码的字符串	是
group	array	指定搜索的分组列表	是
topN	int	得分最高的人脸个数，取值[1,5]	否

livenessEnabled	bool	活体开关，默认关	否
featureUpload	bool	特征值开关，默认关	否

请求示例：

```
{
    "image"："data:image/jpeg;base64...",
    "group":["group1", "group2"],
    "topN":5,
    "livenessEnabled":false,
    "featureUpload":false
}
```

响应参数：

参数	类型	说明
code	int	结果码(0 即为 OK)
msg	string	结果描述
data	json	返回数据

字段信息(data):

参数	类型	说明
detectionResult	obj	人脸图片的检测结果
|—left	int	脸部框左边界坐标
|—top	int	脸部框上边界坐标
|—right	int	脸部框右边界坐标
|—bottom	int	脸部框下边界坐标
|—roll	int	人脸旋转角度
|—yaw	int	人脸水平角度
|—pitch	int	人脸垂直角度
|—blurness	float	图像模糊度(0~100)，越大越模糊
|—livenessScore	float	活体检测结果(0~100)
|—quality	float	图像质量(0~1)，越大质量越好
top1	object	比对结果 top1
|—group	array	比对结果人脸绑定的分组列表
|—|—groupId	string	人脸分组 ID
|—searchScore	float	相似度(0~100)
|—faceToken	string	人脸标识
|—description	string	人脸描述
|—imageUrl	string	base64 编码的人脸图片
top2	object	同 top1
top3	object	同 top1
top4	object	同 top1
top5	object	同 top1

响应示例：

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

```
    "data": {
        "detectionResult": {
            "left": 723,
            "top": 257,
            "right": 985,
            "bottom": 525,
            "roll": 0,
            "yaw": 0,
            "pitch": 0,
            "blurness": 22.477077484130859,
            "quality": 6.408821711007979e-39
        },
        "top1": {
            "group": [{
                "groupId": "default"
            }, {
                "groupId": "1"
            }],
            "searchScore": 100,
            "faceToken": "XaAim2HQpSqI_sYt",
            "description": "3.png",
            "imageUrl": "..."
        },
        "top2": {},
        "top3": {},
        "top4": {},
        "top5": {}
    }
}
```