

接口标准规范 web 版

(V1.5)

文档修改记录

版本号	版本描述	责任人	日期	备注
V1.0	初稿	Emily	2021-02-06	
V1.1	将高拍仪和扫描仪接口分开	Emily	2021-02-06	
V1.2	1. 重新定义 Json 字符串格式和接口名称 2. 高拍仪增加旋转接口	Emily	2021-04-25	
V1.3	1. Camera 接口加入 ID, 可同时控制多个摄像头 2. 发送和回传加入 reqId, 以对应消息编号 3. 修改了命名方式 4. 返回结果用 result 5. 打开摄像头后, 需要调用获取视频才会回传视频	Mike	2021-04-29	
V1.4	1. 将 Json 里的 result 和 devNum 改成 int 型	Emily	2021-04-29	
V1.5	1. 增加错误信息 errorMsg 2. 增加条码识别接口 3. 增加身份证读卡器接口 4. 增加指纹接口 5. 增加人脸接口 6. 增加上传和合并 pdf 接口	Emily	2021-05-10	

目录

1.高拍仪接口说明.....	5
1.1 获取设备信息.....	5
1.2 打开设备.....	7
1.3 获取摄像头视频流.....	7
1.4 关闭摄像头.....	8
1.6 设置图像算法.....	9
1.9 拍摄图片到指定路径.....	10
1.10 拍摄成 BASE64 字符串.....	11
2.扫描仪接口说明.....	12
2.1 获取设备名称.....	12
2.2 获取扫描仪基本信息.....	13
2.3 设置图像算法.....	15
2.4 扫描(base64).....	16
2.5 扫描(文件).....	17
2.6 通知用户扫描最后一张.....	18
3.识别.....	18
3.1 识别条码二维码（暂无）.....	18
4.文件.....	20
4.1 上传（暂无）.....	20
4.2 合并文件（暂无）.....	21
4.外设接口.....	22
4.1 读卡信息.....	22
4.2 指纹采集（暂无）.....	24
4.2 指纹对比（暂无）.....	25
4.人脸模块（暂无）.....	27
4.1 获取人脸摄像头列表.....	27
4.2 打开人脸摄像头.....	28
4.3 获取人脸摄像头视频流.....	29
4.4 关闭摄像头.....	30
4.6 人脸活体检测.....	31
4.6 人证对比.....	31
5.错误码.....	32

1.高拍仪接口说明

1.1 获取设备信息

函数名称	GetCameraInfo
函数功能	获取设备信息
函数原型	<pre>{ "func": "GetCameraInfo", "reqId": "" }</pre>
参数说明 [in]	reqId:函数时序编号，以区分相同函数返回序号
返回值 [out]	<pre>{ "func": "GetCameraInfo", "reqId": "", "result": 0, "devInfo": [{ "devName": "Document Scanner", "mediaTypes": [{ "mediaName": "MJPG", "resolutions": ["1600x1200", "1280x960", "1024x768", "6480x480"] }, { "mediaName": "YUY2", "resolutions": ["1600x1200", "1280x960", "1024x768", "6480x480"] }] }] }</pre>

	<pre>] }] }, { "devName": "USB", "mediaTypes": [{ "mediaName": "MJPG", "resolutions": ["1600x1200", "1280x960", "1024x768", "6480x480"] }, { "mediaName": "YUY2", "resolutions": ["1600x1200", "1280x960", "1024x768", "6480x480"] }] }]</pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段 errorMsg (错误信息)</p> <p>devInfo: 表示是设备组信息, 包括多个设备名称和多个设备信息</p>

1.2 打开设备

函数名称	OpenCamera
函数功能	打开摄像头
函数原型	<pre>{ "func": "OpenCamera", "reqId": "", "devNum": 0 "mediaNum": 0, "resolutionNum": 0, "fps": 15 }</pre>
参数说明 [in]	<p>reqId: 函数时序编号, 以区分相同函数返回序号</p> <p>devNum: 设备编号;</p> <p>mediaNum: 视频格式编号</p> <p>resolutionNum: 分辨率编号</p> <p>fps: 帧率</p>
返回值 [out]	<pre>{ "func": "OpenCamera", "reqId": "", "result": 0 }</pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码, 增加字段</p> <p>errorMsg (错误信息)</p>

1.3 获取摄像头视频流

函数名称	GetCameraVideoBuff
函数功能	获取视频流数据

函数原型	<pre> { "func": "GetCameraVideoBuff", "reqId": "", "devNum": 0, "enable": "true" } </pre>
参数说明 [in]	<p>reqId:函数时序编号，以区分相同函数返回序号</p> <p>devNum: 设备编号;</p> <p>enable:是否继续回传视频</p>
返回值 [out]	<pre> { "func": "GetCameraVideoBuff", "reqId": "", "devNum": 0, "result": 0, "imageBase64Str": "", "mime": "image/jpeg", "width": 1280, "height": 720 } </pre> <p>imageBase64Str:每帧视频 base64</p> <p>mime: 文件格式</p> <p>width: 视频宽</p> <p>height: 视频高</p> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段 errorMsg (错误信息)</p>
详细说明	<p>如果不调用不会返回视频，打开视频后，调用一次之后会一直返回视频，直到发送 enable false 或者关闭视频</p>

1.4 关闭摄像头

函数名称	CloseCamera
------	-------------

函数功能	关闭摄像头
函数原型	<pre> { "func": "CloseCamera", "reqId": "", "devNum": 0 } </pre>
参数说明	reqId:函数时序编号，以区分相同函数返回序号
[in]	devNum: 设备编号;
返回值	<pre> { "func": "CloseCamera", "reqId": "", "result": 0 } </pre>
[out]	result: 返回结果, 0-成功（成功后没有字段 errorMsg），其他-错误码,增加字段 errorMsg（错误信息）
详细说明	<p>如果所有摄像头都处于关闭状态，不做处理</p> <p>关闭当前打开的摄像头</p>

1.6 设置图像算法

函数名称	SetCameraImageInfo
函数功能	设置图像算法参数
函数原型	<pre> { "func": "SetCameraImageInfo", "reqId": "", "devNum": 0, "cropType": 4, "cropRect": ["left": 4, "top": 4, "right": 4, "bottom": 4], } </pre>

	<pre> "imageType":0, "fillBorderType":0, "removalForeign":0, "rotate":0 } </pre>
	<p>reqId:函数时序编号，以区分相同函数返回序号</p> <p>devNum: 设备编号;</p> <p>cropType: 0-不裁切;1-单图裁切;2-多图裁切;3-曲面矫正; 4-手动裁切</p> <p>如果 cropType 设置为 4，增加字段 cropRect，<4 则不显示 cropType</p> <p>(单图裁切&多图裁切视频会有绿框显示，曲面矫正会显示中线)</p> <p>imageType:0-彩色原色;1-灰度图片;2-黑白文档;3-彩色文档;4-红印文档; 蓝印文档;6-彩色照片;7-票据增强</p> <p>fillBorderType:0-不填充;1-填充白色;2-填充周边色</p> <p>removalForeign:0-无;1-去除手指;2-去除装订孔</p> <p>rotate: 0-不旋转;90;180;270</p> <p>(视频会根据旋转角度进行旋转)</p>
返回值 [out]	<pre> { "func": "SetCameraImageInfo", "reqId": "", "result": 0 } </pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码,增加字段</p> <p>errorMsg (错误信息)</p>

1.9 拍摄图片到指定路径

函数名称	CameraCapture
函数功能	拍摄图片到指定位置，根据后缀名确定文件格式，文件格式超出范围即报错

函数原型	<pre> { "func": "CameraCapture", "reqId": "", "devNum": 0, "imagePath": "/tmp/capture.jpg" } </pre>
参数说明 [in]	<p>reqId: 函数时序编号, 以区分相同函数返回序号</p> <p>devNum: 设备编号;</p> <p>imagePath: 图片全路径, 包含后缀名, 如果为空, 生成默认格式, 默认路径文件</p>
返回值 [out]	<pre> { "func": "CameraCapture", "reqId": "", "result": 0, "imagePath": ["/tmp/capture.jpg", "/tmp/capture1.jpg"] } </pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码, 增加字段 errorMsg (错误信息)</p> <p>imagePath: 返回图片数组, 如果多图裁切的话, 返回多个图片路径</p>

1.10 拍摄成 BASE64 字符串

函数名称	CameraCaptureBase64
函数功能	根据格式返回图片 base64 字符串 (如果生成多张图片, 会回传多次)
函数原型	<pre> { "func": "CameraCaptureBase64", "reqId": "", "devNum": 0 } </pre>
参数说明	<p>reqId: 函数时序编号, 以区分相同函数返回序号</p> <p>devNum: 设备编号</p>

[in]	
返回值	<pre> { "func": "CameraCaptureBase64", "reqId": "", "imageCount": 4, "imageIndex": 1, "imageBase64Str": "", "result": 0 } </pre>
[out]	<p>imageBase64Str: 图片 base64 字符串</p> <p>imageCount: 有几张图</p> <p>imageIndex: 第几张图</p> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码, 增加字段 errorMsg (错误信息)</p> <p>注: 如果是多图, 循环给出即可, imgCount 最后一张即为结束</p>

2. 扫描仪接口说明

2.1 获取设备名称

函数名称	GetScannerName	是否必有	是
函数功能	获取设备信息		
函数原型	<pre> { "func": "GetScannerName", "reqId": "" } </pre>		
参数说明	reqId: 函数时序编号, 以区分相同函数返回序号		

[in]	
返回值 [out]	<pre> { "func": "GetScannerName", "reqId": "", "result": 0, "scannerName": ["Canon DRC230", "Uniscan F40D"] } </pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码, 增加字段 errorMsg (错误信息)</p> <p>scannerName: 扫描仪名称列表</p>

2.2 获取扫描仪基本信息

函数名称	GetScannerInfo	是否必有	是
函数功能	获取设备信息		
函数原型	<pre> { "func": "GetScannerInfo", "reqId": "" "devIndex": 0 } </pre>		
参数说明	reqId: 函数时序编号, 以区分相同函数返回序号		
[in]	devIndex: 设备编号 (与获取设备名称列表顺序一致)		
返回值 [out]	<pre> { "func": "GetScannerInfo", "reqId": "", "result": 0, "scanSources": ["Simplex", "Duplex"] } </pre>		

```

    ],
    "scanModes": [
        "Gray",
        "Color",
        "Black & White",
        "Error Diffusion",
        "ATEII"
    ],
    "scanResolutions": [
        "100",
        "150",
        "200",
        "300",
        "400",
        "600"
    ],
    "scanPapers": [
        "A4",
        "A5",
        "A5R",
        "A6",
        "A6R",
        "B5 (JIS)",
        "B6 (JIS)",
        "B6R (JIS)",
        "Legal",
        "Letter",
        "User Defined",
        "Maximum",
        "Auto Size"
    ]
}

```

result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段

errorMsg (错误信息)

scanSources: 扫描类型, 如果驱动没有此选项, 则返回-1

scanModes: 扫描模式, 如果驱动没有此选项, 则返回-1

scanResolutions: 扫描分辨率, 如果驱动没有此选项, 则返回-1

scanPapers: 扫描纸张大小, 如果驱动没有此选项, 则返回-1

2.3 设置图像算法

函数名称	SetScannerImageInfo	是否必有	是
函数功能	设置图像算法参数		
函数原型	<pre>{ "func": "SetCameraImageInfo", "reqId": "", "cropType": 4, "cropRect": ["left": 4, "top": 4, "width": 4, "height": 4], "imageType": 0, "fillBorderType": 0, "removalForeign": 0, "rotate": 0 }</pre>		
参数说明 [in]	<p>result: 返回结果, 0-成功(成功后没有字段 errorMsg), 其他-错误码, 增加字段 errorMsg (错误信息)</p> <p>cropType: 0-不裁切; 1-单图裁切; 2-多图裁切; 3-曲面矫正</p> <p>imageType: 0-彩色原色; 1-灰度图片; 2-黑白文档; 3-彩色文档; 4-红印文档; 5-蓝印文档; 6-彩色照片; 7-票据增强</p> <p>fillBorderType: 0-不填充; 1-填充白色; 2-填充周边色</p> <p>removalForeign: 0-无; 1-去除手指; 2-去除装订孔</p> <p>rotate: 0-不旋转; 90; 180; 270</p>		
返回值 [out]	<pre>{ "func": "SetCameraImageInfo", "reqId": "", "result": 0 }</pre>		

	<pre>} result: 返回结果, 0-成功, 其他-错误码</pre>
--	---

2.4 扫描(base64)

函数名称	Scan	是否必有	是
函数功能	扫描		
函数原型	<pre>{ "func": "Scan", "reqId": "", "devIndex": 0, "scanSource": 2, "scanMode": 0, "scanResolution": 0, "scanPaperSize": 0 }</pre>		
参数说明 [in]	<p>reqId: 函数时序编号, 以区分相同函数返回序号</p> <p>devIndex: 设备编号 (与获取设备名称列表顺序一致)</p> <p>scanSource: 扫描类型编号 (与获取参赛的编号一致)</p> <p>scanMode: 扫描模式编号 (与获取参赛的编号一致)</p> <p>scanResolution: 扫描分辨率编号 (与获取参赛的编号一致)</p> <p>scanPaperSize: 扫描纸张大小编号 (与获取参赛的编号一致)</p>		
返回值 [out]	<pre>{ "func": " Scan ", "reqId": "", "imageBase64Str": " ", "result": 0 }</pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码, 增加字段 errorMsg (错误信息)</p>		

	imageBase64Str:图片 base64
--	--------------------------

2.5 扫描(文件)

函数名称	Scan	是否必有	是
函数功能	扫描		
函数原型	<pre> { "func": "Scan", "reqId": "", "devIndex": 0, "scanSource": 2, "scanMode": 0, "scanResolution": 0, "scanPaperSize": 0 "imagePath": "/tmp/scannerimage.jpg" } </pre>		
参数说明 [in]	<p>reqId:函数时序编号，以区分相同函数返回序号</p> <p>devIndex: 设备编号（与获取设备名称列表顺序一致）</p> <p>scanSource: 扫描类型编号（与获取参赛的编号一致）</p> <p>scanMode: 扫描模式编号（与获取参赛的编号一致）</p> <p>scanResolution: 扫描分辨率编号（与获取参赛的编号一致）</p> <p>scanPaperSize: 扫描纸张大小编号（与获取参赛的编号一致）</p> <p>imagePath:图片保存路径，全路径，如果为空，生成默认格式，默认路径文件</p>		
返回值 [out]	<pre> { "func": " Scan ", "reqId": "", "imagePath": ["/tmp/scanner.jpg", "/tmp/scanner1.jpg"] } </pre>		

	<pre>"result":0 }</pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码,增加字段</p> <p>errorMsg (错误信息)</p> <p>imagePath:返回多个路径</p>
--	--

2.6 通知用户扫描最后一张

函数名称	GetScanEndStatue	是否必有	是
函数功能	通知用户扫描最后一张		
函数原型	无		
参数说明 [in]	reqId:函数时序编号, 以区分相同函数返回序号		
返回值 [out]	<pre>{ "func": "GetScanEndStatue", "reqId": "", "result": 0, "scanEndStatue": 0 }</pre> <p>scanEndStatue: 0, 最后一张扫描结束</p> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码,增加字段</p> <p>errorMsg (错误信息)</p>		

3.识别

3.1 识别条码二维码（暂无）

函数名称	RecogBarCode	是否必有	是
------	--------------	------	---

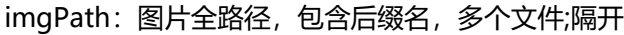
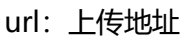
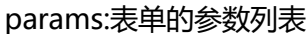
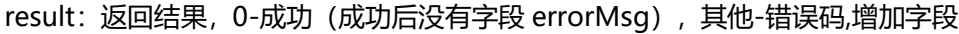
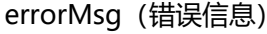
函数功能	识别图片上的条码
函数原型	<pre> { "functionName": "RecogBarCode", "reqId": "", "imageBase64": "", "imagePath": "" } </pre>
参数说明	reqId: 函数时序编号，以区分相同函数返回序号
[in]	imageBase64: base64; 如果为空，读取 imagePath 路径去识别
	imagePath: 图片全路径，包含后缀名，如果多个文件需要去做条码识别，用；隔 开
返回值	<pre> { "func": " RecogBarCode", "reqId": "", "result": 0, "codeInfos": [{ "codeNum": "XXXXXXXX", "codeType": "code128", "left": "10", "top": "10", "right": "100", "bottom": "100" }, { "codeNum": "XXXXX", "codeType": "QR", "left": "100", "top": "100", "right": "100", "bottom": "100" }] } </pre>
[out]	codeNum: 条码

	codeType;编码方式, 需支持 (Code128、Code39、EAN13、EAN8、MSI Pharma、UPC-A、UPC-E、Codabar、Interleaved 2 of 5、Code 93、UCC_128、QR code) result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段 errorMsg (错误信息)
详细说明	识别路径指定的文件条码, 如果有多个条码, 读取所有条码的信息

4.文件

4.1 上传（暂无）

函数名称	UploadHttpFile	是否必有	是
函数功能	上传文件		
函数原型	<pre>{ "functionName": "UploadHttpFile", "reqId": "", "url": "", "imageBase64": "", "imagePath": "", "params": { "name": "name1Value", "name": 1111, "name": "name2Value" }, }</pre>		
参数说明	reqId:函数时序编号, 以区分相同函数返回序号		

[in]	  
返回值 [out]	<pre>{ "func": " UploadHttpFile ", "reqId": "", "result": 0 }</pre>  
详细说明	

4.2 合并文件（暂无）

函数名称	MergeFile	是否必有	是
函数功能	合并文件		
函数原型	<pre>{ "functionName": "MergeFile", "reqId": "", "imagePath": ["C:/tmp/sacnnerimage1.jpg", "C:/tmp/sacnnerimage2.jpg"] "fileType": 0 "filePath": "" }</pre>		
参数说明 [in]	  		

	filePath: 生成 pdf 文件路径, 全路径, 可以传空, 由 SDK 返回路径
返回值 [out]	<pre> { "func": "MergeFile", "reqId": "", "result": 0, "filePath": "" } </pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码,增加字段 errorMsg (错误信息)</p>

4.外设接口

4.1 读卡信息

函数名称	ReadIDCardInfo	是否必有	是
函数功能	读取身份证读卡器信息		
函数原型	<pre> { "func": "ReadIDCardInfo", "reqId": "", "idType": 0 } </pre>		
参数说明 [in]	<p>reqId:函数时序编号, 以区分相同函数返回序号</p> <p>idType:0—读身份证信息; 1-读社保卡信息; 2-读银行卡信息;</p>		

返回值

[out]

身份证返回信息:

```
{
    "functionName": "ReadIDCardInfo",
    "reqId": "",
    "result": 0,
    "calrdInfo": {
        "name": "123",
        "gender": "女",
        "folk": "汉",
        "birth": "1999-05-01",
        "address": "",
        "id": "",
        "issue": "",
        "startDate": "",
        "endDate": "",
        "imageBase64": "",
        "fingerFeature": "123"
    }
}
```

如果是外国人身份证, 如果没有 folk 字段, 那就置为空

name: 姓名

gender: 性别

folk: 民族

birth: 生日

address: 地址

startDate: 时间

endDate: 到期时间

imageBase: 身份证头像 base64

社保卡返回信息:

```
{
    "functionName": "ReadIDCardInfo",
    "reqId": "",
    "result": 0,
```

	<pre>"IDCardInfo": { "name": "123", "id": "", "gender": "女", "folk": "汉", "birth": "1999-05-01", "address": "" } }</pre> <p>银行卡返回信息:</p> <pre>{ "functionName": "ReadIDCardInfo", "reqId": "", "result": 0, "idNum": 1234456789 }</pre> <p>result: 返回结果, 0-成功(成功后没有字段 errorMsg), 其他-错误码,增加字段 errorMsg (错误信息)</p>
--	--

4.2 指纹采集（暂无）

函数名称	ReadFingerInfo	是否必有	是
函数功能	读取指纹信息		
函数原型	<pre>{ "func": "ReadFingerInfo", "reqId": "" }</pre>		
参数说明 [in]	reqId:函数时序编号, 以区分相同函数返回序号		

返回值	<pre> { "functionName": "ReadFingerInfo", "reqId": "", "result": 0, "fingerImageBase64": "", "fingerFeature": "" } </pre>
[out]	<p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段</p> <p>errorMsg (错误信息)</p> <p>fingerBase64: 指纹信息</p> <p>fingerFeature: 指纹特征</p>

4.2 指纹对比（暂无）

函数名称	FingerMatch	是否必有	是
函数功能	指纹对比		
函数原型	<pre> { "func": "FingerMatch", "reqId": "", "fingerImageBase64": "", "fingerFeature": "", "timeout": 5000, "threshold": 0.9 } </pre>		

<p>参数说明</p> <p>[in]</p>	<p>reqId:函数时序编号，以区分相同函数返回序号</p> <p>fingerImageBase64: 指纹 base64</p> <p>fingerFeature: 指纹 特征 如果 Base64 为空，用 fingerFeature 对比</p> <p>timeOut:对比超时时间 单位 ms</p> <p>threshold: 相似度阈值，可以不传，默认 0.9</p>
<p>返回值</p> <p>[out]</p>	<pre> { "functionName": "FingerMatch", "reqId": "", "result": 0, "fingerImageBase64": "" } </pre> <p>result: 返回结果，0-成功（成功后没有字段 errorMsg），其他-错误码,增加字段</p> <p>errorMsg (错误信息)</p> <p>fingerImageBase64:返回匹配成功图片 Base64</p>

4.人脸模块 （暂无）

4.1 获取人脸摄像头列表

函数名称	GetFaceCameraInfo
函数功能	获取人脸设备（双目）信息（用分辨率<500 万的设备均为人脸设备，红白双目摄像头会枚举为单一设备）
函数原型	<pre>{ "func": " GetFaceCameraInfo", "reqId": "" }</pre>
参数说明 [in]	reqId:函数时序编号，以区分相同函数返回序号
返回值 [out]	<pre>{ "func": " GetFaceCameraInfo", "reqId": "", "result": 0, "faceCameraInfo": [{ "faceCameraName ": "Document Scanner", "faceCameraMedias ": ["MJPEG", "YUY2"], "faceCameraType ": 1, "faceCameraResolutions ": ["1600x1200", "1280x960", "1024x768", "6480x480"] }, { "faceCameraName": "USB Camera", "faceCameraMedias": ["MJPEG",</pre>

	<pre> "YUY2"], "faceCameraResolutions": ["1600x1200", "1280x960", "1024x768", "640x480"] }] } </pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段 errorMsg (错误信息)</p> <p>faceInfo: 表示是设备组信息, 包括多个设备名称和多个设备信息</p> <p>faceCameraType: 1--普通摄像头; 2---红白双目</p>

4.2 打开人脸摄像头

函数名称	OpenFaceCamera
函数功能	打开人脸摄像头
函数原型	<pre> { "func": "OpenFaceCamera", "reqId": "", "faceCameraNum": 0, "faceMediaNum": 0, "faceResolutionNum": 0, "faceFps": 15 } </pre>
参数说明 [in]	<p>reqId:函数时序编号, 以区分相同函数返回序号</p> <p>faceCameraNum: 设备编号;</p> <p>faceMedaiNUm: 人脸设备视频</p>

	faceResolutionNum:分辨率编号
返回值	<pre>{ "func": "OpenFaceCamera", "reqId": "", "result": 0 }</pre>
[out]	result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码,增加字段 errorMsg (错误信息)

4.3 获取人脸摄像头视频流

函数名称	GetFaceCameraBuff
函数功能	获取视频流数据
函数原型	<pre>{ "func": "GetFaceCamraBuff", "reqId": "", "faceCameraNum": 0, "enable": "true" }</pre>
参数说明	reqId:函数时序编号, 以区分相同函数返回序号
[in]	faceCameraNum: 设备编号; faceEnable:是否继续回传视频
返回值	<pre>{ "func": "GetFaceCamraBuff", "reqId": "", "faceCameraNum": 0, "result": 0, "imgBase64Str": "", "mime": "image/jpg", "width": 1280, "height": 720 }</pre>
[out]	imgBase64Str:每帧视频 base64

	<p>width: 视频宽</p> <p>height: 视频高</p> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段 errorMsg (错误信息)</p>
详细说明	<p>如果不调用不会返回视频, 打开视频后, 调用一次之后会一直返回视频, 直到发送 enable false 或者关闭视频</p>

4.4 关闭摄像头

函数名称	CloseFaceCamera
函数功能	关闭摄像头
函数原型	<pre>{ "func": "CloseFaceCamera", "reqId": "", "faceNum": 0 }</pre>
参数说明	reqId:函数时序编号, 以区分相同函数返回序号
[in]	faceNum: 设备编号;
返回值	<pre>{ "func": "CloseFaceCamera", "reqId": "", "result": 0 }</pre>
[out]	<p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段 errorMsg (错误信息)</p>
详细说明	<p>如果所有摄像头都处于关闭状态, 不做处理</p> <p>关闭当前打开的摄像头</p>

4.6 人脸活体检测

函数名称	SetFaceLivingBodyState
函数功能	人脸活体检测
函数原型	<pre>{ "func": "SetFaceLivingBodyState", "reqId": "", "faceCameraNum": 0, "faceLivingBody": 0 }</pre>
参数说明	reqId: 函数时序编号, 以区分相同函数返回序号
[in]	faceNum: 设备编号;
	faceLivingBody: 1-打开活体检测; 0-关闭
	注: 和读取参数编号一一对应
返回值	<pre>{ "func": "SetFaceLivingBodyState", "reqId": "", "result": 0 }</pre>
[out]	result: 返回结果, 0-成功 (成功后没有字段 errorMsg), 其他-错误码, 增加字段 errorMsg (错误信息)

4.6 人证对比

函数名称	FaceMatch	是否必有	是
函数功能	支持文件和视频比对, 文件和文件对比, 视频和视频之间对比		
函数原型	<pre>"func": "FaceMatch", "reqId": "", "faceImageBase64": "", "faceImagePath": "",</pre>		

	<pre> "threshold":0.7 } </pre>
参数说明 [in]	reqId:函数时序编号，以区分相同函数返回序号 faceFilePath: 文件对比 faceBase64: 视频 base64 对比 threshold:相似度阈值 默认 0.7(0-1) 注：如果 faceFilePath 为一个文件，faceBase64 不为空，则表示文件和视频比对
返回值 [out]	<pre> { "func": " FaceMatch ", "reqId": "", "result": 0, "faceImageBase64": } </pre> <p>result: 返回结果, 0-成功 (成功后没有字段 errorMsg) , 其他-错误码,增加字段</p> <p>errorMsg (错误信息)</p> <p>faceImageBase64: 匹配成功, 视频的截图</p>

5.错误码

以 json 字符串作为返回值，解析回调参数，内容包含：success:方法执行成功标志；errorMsg:失败信息；functionName:调用方法；成功标识为 1；