

Morpho 串口指纹 SDK API 说明 V1.0

Morpho UART Fingerprint SDK API Documentation V1.0

一、SDK概览

I. SDK Overview

二、详细API说明

II. Detailed API Documentation

2.1 init()

2.2 free()

2.3 setGrabCallBack(GrabCallBack)

2.4 startGrab(String, String)

2.5 setPtCaptureCallBack(PtCaptureCallBack)

2.6 setTemplateVerifyCallBack(TemplateVerifyCallBack)

2.7 setPtEnrollCallBack(EnrollCallBack)

2.8 setIdentificationCallBack(IdentificationCallBack)

2.9 startPtCapture(boolean)

2.10 startPtCapturePKComp(boolean)

2.11 startTemplateVerify(byte[], int)

2.12 startEnroll(int, String)

2.13 startEnroll(int, String, String, String)

2.14 startIdentification()

2.15 setMorphoLoadKs(byte[])

2.16 getMorphoDescriptor()

2.17 getMorphoPIDSN()

2.18 morphoEraseAllBase()

2.19 getMorphoSecurityLevel()

2.20 setMorphoSecurityLevel(int)

2.21 setTimeOut(int)

2.30 PtCaptureCallBack Interface

2.31 TemplateVerifyCallBack Interface

2.32 EnrollCallBack Interface

2.33 GrabCallBack Interface

2.34 IdentificationCallBack Interface

一、SDK 概览

I. SDK Overview

`com.rscja.deviceapi.FingerprintwithMorpho` 是基于 Android 平台的 Morpho 指纹模块操作类，采用单例模式设计，核心作用是为上层应用提供标准化的 API，实现指纹模块的初始化与资源释放、指纹图像采集、特征值生成与存储（缓冲区 / Flash）、指纹特征上传下载、指纹比对（精确比对 / 库内搜索）、模板管理（删除 / 清空）及指纹图像回调获取等全流程指纹识别功能。

`com.rscja.deviceapi.FingerprintwithMorpho` is a Morpho fingerprint module operation class based on the Android platform, designed with the singleton pattern. Its core function is to provide a standardized API for upper-layer applications, enabling the full-process fingerprint recognition functions of the fingerprint module including initialization and resource release, fingerprint image collection, feature generation and storage (buffer/Flash), fingerprint feature upload and download, fingerprint matching (exact matching/database search), template management (deletion/clearing), and callback acquisition of fingerprint images.

1. 简单易用的单例设计

1. Simple and Easy-to-Use Singleton Design

您无需手动创建该类的实例，只需通过 `getInstance()` 方法即可获取全局唯一的管理对象，确保调用简便且高效利用资源。

You do not need to manually create an instance of this class. You can obtain the globally unique management object only through the `getInstance()` method, ensuring that the calling process is simple and resource-efficient.

2. 使用步骤

2. Usage Steps

第一步: 通过 `init()` 初始化指纹相关资源。

Step 1: Initialize the fingerprint-related resources via `init()`.

第二步: 初始化成功后，根据需求执行以下操作：

Step 2: After successful initialization, perform the following operations as required:

- `startEnroll(int, String)` 注册指纹
- `startEnroll(int, String)` Register fingerprint
- `startIdentification()` 识别指纹
- `startIdentification()` Identify fingerprint
- `startGrab(String, String)` 生成指纹图片
- `startGrab(String, String)` Generate fingerprint image
- `startPtCapture(bool)` 下载模板
- `startPtCapture(bool)` Download template

第三步: 退出app调用 `free()` 释放资源。

Step 3: Call `free()` to release resources when exiting the app.

3.使用示例

3.Usage Examples

```
public class FingerprintDemo {
    public void test(){
        int result = mFingerprint.init();
        if(result != 0){
            //Initialization failed
            return;
        }
        //*****Register fingerprint*****
        mFingerprint.setPtEnrollCallBack(new FingerprintWithMorpho.EnrollCallBack() {
            @Override
            public void messageInfo(String s,int code) {
            }

            @Override
            public void onComplete(boolean result, int code) {
                if(result) {
                    //Enrollment succeeded
                    /* Fingerprint image processing
                    wsqUtils wsqUtils=new wsqUtils();
                    byte[] data=wsqUtils.decompressWSQ(new File(imgDirectory,imgName));
                    wsqUtils.generateBitmap(data,new File(imgDirectory,file_bmp));
                    */
                }else{
                    //Enrollment failed
                }
            }
        });
        /*Fingerprint image is required
        String file_bmp = "finger.bmp";
        String imgDirectory="sdcard/";
        String imgName="finger.wsq";
        mFingerprint.startEnroll(id,name,imgDirectory,imgName);
        */
        mFingerprint.startEnroll(id,name);

        //*****Identify fingerprint*****
        mFingerprint.setIdentificationCallBack(new FingerprintWithMorpho.IdentificationCallBack() {
            @Override
            public void messageInfo(String s,int code) { }

            @Override
            public void onComplete(boolean result, int id, String name, int Param) {
                if(result) {
                    //Identification succeeded
                }else{
                    //Identification failed
                }
            }
        });
        mFingerprint.startIdentification();

        //*****Acquire fingerprint image*****
        mFingerprint.setGrabCallBack(new FingerprintWithMorpho.GrabCallBack() {
            @Override
            public void messageInfo(String msg, int codeID) { }

            @Override
```

```

        public void onComplete(boolean result, int Param) {
            if (result) {
                //Image acquisition succeeded
                wsqUtils wsqUtils=new wsqUtils();
                byte[] data = wsqUtils.decompressWSQ(PATH + file_wsq);
                wsqUtils.generateBitmap(data, PATH + file_bmp);
                Bitmap bitmap = BitmapFactory.decodeFile(PATH + file_bmp);
            } else {
                //Image acquisition failed
            }
        }
    });
    String file_bmp = "finger.bmp";
    String file_wsq = "finger.wsq";
    String PATH = "/sdcard/";
    mFingerprint.startGrab(PATH, file_wsq);

    //*****Get fingerprint template*****
    mFingerprint.setPtCaptureCallBack(new FingerprintwithMorpho.PtCaptureCallBack() {
        @Override
        public void messageInfo(String msg,int code) {

        }

        @Override
        public void onComplete(boolean result, byte[] bytes, int i) {
            if(result) {
                //Template acquisition succeeded, bytes is the fingerprint template data
            }else{
                //Template acquisition failed
            }
        }
    });
    mFingerprint.startPtCapture(false);

    //*****Fingerprint template comparison*****
    mFingerprint.setTemplateVerifyCallBack(new FingerprintwithMorpho.TemplateVerifyCallBack() {
        @Override
        public void messageInfo(String msg, int Code) {
        }
        @Override
        public void onComplete(boolean result, int i) {
            if(result){
                //Template comparison succeeded
            }else{
                //Template comparison failed
            }
        }
    });
    byte[] templateData=readFile() ;
    mFingerprint.startTemplateVerify(templateData,0);

    //.....
    mFingerprint.free();
}
}

```

二、详细API说明

II. Detailed API Documentation

2.1 init()

项目 Item	说明 Description
原型 Prototype	boolean init()
描述 Description	初始化指纹读写器的相关资源 Initialize the relevant resources of the fingerprint reader
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

2.2 free()

项目 Item	说明 Description
原型 Prototype	boolean free()
描述 Description	释放指纹读写器已初始化的相关资源 Release the initialized relevant resources of the fingerprint reader
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

2.3 setGrabCallBack(GrabCallBack)

项目 Item	说明 Description
原型 Prototype	void setGrabCallBack(FingerprintWithMorpho.GrabCallBack callBack)
描述 Description	设置指纹图像采集的回调接口，用于接收指纹图像采集过程中的实时消息和采集结果。 Set up the callback interface for fingerprint image acquisition to receive real-time messages and results during the acquisition process.
参数 Parameters	callBack：GrabCallBack回调接口实例，用于处理图像采集的回调数据 callBack: GrabCallBack callback interface instance, used to process callback data of image acquisition
返回 Return	void

2.4 startGrab(String, String)

项目 Item	说明 Description
原型 Prototype	void startGrab(String imgDirectory, String imgName)
描述 Description	启动指纹图像采集流程，采集的图像将保存到指定目录和文件名路径下，需先调用 setGrabCa11Back 方法设置回调接口以接收采集数据 Start the fingerprint image acquisition process. The acquired image will be saved to the path of the specified directory and file name. You need to call the <code>setGrabCa11Back</code> method first to set up a callback interface for receiving acquisition data
参数 Parameters	imgDirectory: 指纹图像保存的目录路径 imgDirectory: Directory path for saving fingerprint images imgName: 指纹图像的文件名 imgName: File name of the fingerprint image
返回 Return	void
备注 Remark	可通过调用 <code>setTimeout(int)</code> 方法设置该采集流程的超时时间 You can set the timeout period for the acquisition process by calling the <code>setTimeout(int)</code> method

2.5 setPtCaptureCallBack(PtCaptureCallBack)

项目 Item	说明 Description
原型 Prototype	void setPtCaptureCallBack(FingerprintWithMorpho.PtCaptureCallBack callBack)
描述 Description	设置指纹模板获取的回调接口，用于接收指纹模板获取过程中的实时消息和获取结果。 Set up the callback interface for fingerprint template acquisition to receive real-time messages and results during the acquisition process.
参数 Parameters	callBack: PtCaptureCallBack回调接口实例，用于处理模板获取的回调数据 callBack: PtCaptureCallBack callback interface instance, used to process callback data of template acquisition
返回 Return	void

2.6 setTemplateVerifyCallBack(TemplateVerifyCallBack)

项目 Item	说明 Description
原型 Prototype	void setTemplateVerifyCallBack(FingerprintWithMorpho.TemplateVerifyCallBack callBack)
描述 Description	设置指纹模板比对的回调接口，用于接收模板比对过程中的实时消息和比对结果。 Set up the callback interface for fingerprint template comparison to receive real-time messages and results during the comparison process.
参数 Parameters	callBack: TemplateVerifyCallBack回调接口实例，用于处理模板比对的回调数据 callBack: TemplateVerifyCallBack callback interface instance, used to process callback data of template comparison

项目 Item	说明 Description
返回 Return	void

2.7 setPtEnrollCallBack(EnrollCallBack)

项目 Item	说明 Description
原型 Prototype	void setPtEnrollCallBack(FingerprintWithMorpho.EnrollCallBack callBack)
描述 Description	设置指纹采集注册的回调接口，用于接收指纹采集注册过程中的实时消息和采集结果。 Set up the callback interface for fingerprint enrollment to receive real-time messages and results during the enrollment process.
参数 Parameters	callBack: EnrollCallBack回调接口实例，用于处理指纹采集注册的回调数据 callBack: EnrollCallBack callback interface instance, used to process callback data of fingerprint enrollment
返回 Return	void

2.8 setIdentificationCallBack(IdentificationCallBack)

项目 Item	说明 Description
原型 Prototype	void setIdentificationCallBack(FingerprintWithMorpho.IdentificationCallBack callBack)
描述 Description	设置指纹验证的回调接口，用于接收指纹验证过程中的实时消息和验证结果。 Set up the callback interface for fingerprint verification to receive real-time messages and results during the verification process.
参数 Parameters	callBack: IdentificationCallBack回调接口实例，用于处理指纹验证的回调数据 callBack: IdentificationCallBack callback interface instance, used to process callback data of fingerprint verification
返回 Return	void

2.9 startPtCapture(boolean)

项目 Item	说明 Description
原型 Prototype	void startPtCapture(boolean encryptflag)
描述 Description	启动指纹模板获取流程，可指定是否对获取的模板数据进行加密，需先调用 setPtCaptureCa11Back 方法设置回调接口以接收模板数据 Start the fingerprint template acquisition process. You can specify whether to encrypt the acquired template data. You need to call the setPtCaptureCa11Back method first to set up a callback interface for receiving template data
参数 Parameters	encryptflag: 是否加密指纹模板数据 (true=加密, false=不加密) encryptflag: Whether to encrypt fingerprint template data (true=Encrypt, false=Do not encrypt)

项目 Item	说明 Description
返回 Return	void
备注 Remark	可通过调用 <code>setTimeout(int)</code> 方法设置该指纹模板获取流程的超时时间 You can set the timeout period for the fingerprint template acquisition process by calling the <code>setTimeout(int)</code> method

2.10 startPtCapturePKComp(boolean)

项目 Item	说明 Description
原型 Prototype	void startPtCapturePKComp(boolean encryptflag)
描述 Description	启动PKComp模式的指纹模板获取流程，可指定是否对获取的模板数据进行加密，需先调用 <code>setPtCaptureCallback</code> 方法设置回调接口以接收模板数据 Start the fingerprint template acquisition process in PKComp mode. You can specify whether to encrypt the acquired template data. You need to call the <code>setPtCaptureCallback</code> method first to set up a callback interface for receiving template data
参数 Parameters	encryptflag: 是否加密PKComp模式下的指纹模板数据 (true=加密, false=不加密) encryptflag: Whether to encrypt fingerprint template data in PKComp mode (true=Encrypt, false=Do not encrypt)
返回 Return	void
备注 Remark	可通过调用 <code>setTimeout(int)</code> 方法设置该指纹模板获取流程的超时时间 You can set the timeout period for the fingerprint template acquisition process by calling the <code>setTimeout(int)</code> method

2.11 startTemplateVerify(byte[], int)

项目 Item	说明 Description
原型 Prototype	void startTemplateVerify(byte[] template, int type)
描述 Description	启动指纹模板比对验证流程，需先调用 <code>setTemplateVerifyCallback</code> 方法设置回调接口以接收比对结果数据 Start the fingerprint template comparison and verification process. You need to call the <code>setTemplateVerifyCallback</code> method first to set up a callback interface for receiving comparison result data
参数 Parameters	template: 待比对的指纹模板数据字节数组 template: Byte array of fingerprint template data to be compared type: 模板比对类型 (0=PKComp模式) type: Template comparison type (0=PKComp mode)
返回 Return	void
备注 Remark	可通过调用 <code>setTimeout(int)</code> 方法设置该比对验证流程的超时时间 You can set the timeout period for the comparison and verification process by calling the <code>setTimeout(int)</code> method

2.12 startEnroll(int, String)

项目 Item	说明 Description
原型 Prototype	void startEnroll(int id, String name)
描述 Description	启动指纹采集注册流程，为采集的指纹绑定指定的ID和姓名信息，需先调用 setPtEnrollCallback 方法设置回调接口以接收采集数据 Start the fingerprint enrollment process and bind the specified ID and name information to the acquired fingerprint. You need to call the setPtEnrollCallback method first to set up a callback interface for receiving acquisition data
参数 Parameters	id: 指纹注册的唯一标识ID id: Unique identification ID for fingerprint enrollment name: 指纹关联的姓名信息 name: Name information associated with the fingerprint
返回 Return	void
备注 Remark	可通过调用 setTimeout(int) 方法设置该采集注册流程的超时时间 You can set the timeout period for the enrollment process by calling the setTimeout(int) method

2.13 startEnroll(int, String, String, String)

项目 Item	说明 Description
原型 Prototype	void startEnroll(int id, String name, String imgDirectory, String imgName) throws IOException
描述 Description	启动指纹采集注册流程，为采集的指纹绑定指定的唯一ID和姓名信息，同时将采集的指纹图像保存至指定目录（仅支持.wsq格式）；因指纹图像传输需占用数秒时间，本方法的执行耗时会高于 startEnroll(int id, String name) 方法，使用前需先调用 setPtEnrollCallback 方法设置回调接口以接收采集数据 Start the fingerprint enrollment process, bind the specified unique ID and name information to the acquired fingerprint, and save the acquired fingerprint image to the specified directory (only .wsq format is supported); since fingerprint image transmission takes several seconds, this method takes longer to execute compared to the startEnroll(int id, String name) method, and you need to call the setPtEnrollCallback method first to set up a callback interface for receiving acquisition data before use
参数 Parameters	id: 指纹注册的唯一标识ID id: Unique identification ID for fingerprint enrollment name: 指纹关联的姓名信息 name: Name information associated with the fingerprint imgDirectory: 指纹图像保存的目录路径 imgDirectory: Directory path for saving fingerprint images imgName: 指纹图像的文件名（仅支持.wsq格式） imgName: File name of the fingerprint image (only .wsq format is supported)
返回 Return	void
异常 Exception	IOException: 指纹图像保存过程中发生IO异常时抛出 IOException: Thrown when an IO exception occurs during the saving of fingerprint images

项目 Item	说明 Description
备注 Remark	可通过调用 <code>setTimeout(int)</code> 方法设置该采集注册流程的超时时间 You can set the timeout period for this fingerprint enrollment process by calling the <code>setTimeout(int)</code> method

2.14 startIdentification()

项目 Item	说明 Description
原型 Prototype	<code>void startIdentification()</code>
描述 Description	启动指纹验证流程，需先调用 <code>setIdentificationCallback</code> 方法设置回调接口以接收验证结果数据 Start the fingerprint verification process. You need to call the <code>setIdentificationCallback</code> method first to set up a callback interface for receiving verification result data
参数 Parameters	无 N/A
返回 Return	<code>void</code>
备注 Remark	可通过调用 <code>setTimeout(int)</code> 方法设置该指纹验证流程的超时时间 You can set the timeout period for the fingerprint verification process by calling the <code>setTimeout(int)</code> method

2.15 setMorphoLoadKs(byte[])

项目 Item	说明 Description
原型 Prototype	<code>boolean setMorphoLoadKs(byte[] keybuf)</code>
描述 Description	设置指纹模块加密数据的密钥，用于后续加密/解密指纹数据的密钥配置 Set the key for encrypted data of the fingerprint module, used to configure the key for subsequent encryption/decryption of fingerprint data

| 参数
Parameters | `keybuf`: 加密密钥字节数组，长度必须为24个字节
`keybuf`: Encryption key byte array (must be 24 bytes) |
| 返回
Return | `true`:成功 `false`:失败
`true`:success `false`:failure |

2.16 getMorphoDescriptor()

项目 Item	说明 Description
原型 Prototype	<code>String getMorphoDescriptor()</code>
描述 Description	获取指纹模块的版本信息，返回包含版本号、型号等相关的版本描述字符串 Acquire the version information of the fingerprint module, returning a version description string containing version number, model and other related information

项目 Item	说明 Description
参数 Parameters	无 N/A
返回 Return	String: 指纹模块的版本信息字符串，获取失败时返回null String: Version information string of the fingerprint module, returns null if acquisition fails

2.17 getMorphoPIDSN()

项目 Item	说明 Description
原型 Prototype	String getMorphoPIDSN()
描述 Description	获取指纹模块的序列号（SN），返回唯一标识该模块的SN字符串 Acquire the serial number (SN) of the fingerprint module, returns the SN string that uniquely identifies the module
参数 Parameters	无 N/A
返回 Return	String: 指纹模块的序列号字符串，获取失败时返回null String: Serial number string of the fingerprint module, returns null if acquisition fails

2.18 morphoEraseAllBase()

项目 Item	说明 Description
原型 Prototype	boolean morphoEraseAllBase()
描述 Description	删除指纹模块中所有已注册的指纹信息（包括指纹模板、ID、姓名等关联数据） Delete all registered fingerprint information in the fingerprint module (including fingerprint templates, IDs, names, and other related data)
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

2.19 getMorphoSecurityLevel()

项目 Item	说明 Description
原型 Prototype	int getMorphoSecurityLevel()
描述 Description	获取指纹模块当前的安全等级，安全等级决定指纹验证的严格程度 Acquire the current security level of the fingerprint module; the security level determines the strictness of fingerprint verification
参数 Parameters	无 N/A

项目 Item	说明 Description
返回 Return	int: 0-2=对应的安全等级（0最低，2最高），-1=安全等级获取失败 int: 0-2=Corresponding security level (0 is the lowest, 2 is the highest), -1=Acquisition of security level failed

2.20 setMorphoSecurityLevel(int)

项目 Item	说明 Description
原型 Prototype	boolean setMorphoSecurityLevel(int level)
描述 Description	设置指纹模块的安全等级，安全等级会影响指纹验证的严格程度 Set the security level of the fingerprint module; the security level will affect the strictness of fingerprint verification
参数 Parameters	level: 待设置的安全等级，取值范围为0-2（0最低，2最高） level: Security level to be set, the value range is 0-2 (0 is the lowest, 2 is the highest)
返回 Return	boolean: true=安全等级设置成功，false=设置失败（如等级值超出0-2范围） boolean: true=Security level set successfully, false=Setup failure (e.g., the level value is out of the 0-2 range)

2.21 setTimeout(int)

项目 Item	说明 Description
原型 Prototype	boolean setTimeout(int timeOut)
描述 Description	设置上位机与指纹模块之间的通信超时时间，超时后未完成的各类指纹操作通信流程将自动终止 Set the communication timeout period between the host computer and the fingerprint module, and all unfinished fingerprint operation communication processes will be automatically terminated after the timeout
参数 Parameters	timeOut: 通信超时时间（单位：秒），该参数最小值为5秒 timeout: Communication timeout period (in seconds); minimum value is 5 seconds
返回 Return	true=设置成功，false=设置失败 true=Set successfully, false=Setup failed

2.30 PtCaptureCallBack 接口

2.30 PtCaptureCallBack Interface

指纹模板获取回调接口，用于实时反馈指纹模板获取过程中的消息及获取结果。

Callback interface for fingerprint template acquisition, used to provide real-time feedback of messages and acquisition results during fingerprint template acquisition.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg, int codeID)	接收指纹模板获取的实时状态消息 Receive real-time status messages of fingerprint template acquisition	msg: 指纹模板获取的实时描述信息 msg: Real-time description information of fingerprint template acquisition codeID: 消息对应的状态编码 codeID: Status code corresponding to the message	无 N/A
onComplete(boolean b, byte[] data, int Param)	指纹模板获取完成后的回调方法，返回获取结果 Callback method after fingerprint template acquisition is completed, returns acquisition result	b: true=获取成功，false=获取失败 b: true=Acquisition success, false=Acquisition failure data: 指纹模板数据字节数组（获取成功时有值，失败时为null） data: Byte array of fingerprint template data (valued if acquisition succeeds, null if failed) Param: 附加参数，返回获取失败的原因（成功时无实际意义） Param: Additional parameter returning the reason for failure (undefined if successful)	无 N/A

2.31 TemplateVerifyCallBack 接口

2.31 TemplateVerifyCallBack Interface

指纹模板比对回调接口，用于实时反馈模板比对过程中的消息及比对结果。

Callback interface for fingerprint template comparison, used to provide real-time feedback of messages and comparison results during template comparison.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg, int codeID)	接收指纹模板比对的实时状态消息 Receive real-time status messages of fingerprint template comparison	msg: 指纹模板比对的实时描述信息 msg: Real-time description information of fingerprint template comparison codeID: 消息对应的状态编码 codeID: Status code corresponding to the message	无 N/A
onComplete(boolean b, int Param)	指纹模板比对完成后的回调方法，返回比对结果 Callback method after fingerprint template comparison is completed, returns comparison result	b: true=比对成功，false=比对失败 b: true=Comparison success, false=Comparison failure Param: 附加参数，返回比对失败的原因（成功时无实际意义） Param: Additional parameter returning the reason for failure (undefined if successful)	无 N/A

2.32 EnrollCallBack 接口

2.32 EnrollCallBack Interface

指纹采集注册回调接口，用于实时反馈指纹采集注册过程中的消息及采集结果。

Callback interface for fingerprint enrollment, used to provide real-time feedback of messages and acquisition results during fingerprint enrollment.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg, int codeID)	接收指纹采集注册的实时状态消息 Receive real-time status messages of fingerprint enrollment	msg: 指纹采集注册的实时描述信息 msg: Real-time description information of fingerprint enrollment codeID: 消息对应的状态编码 codeID: Status code corresponding to the message	无 N/A
onComplete(boolean b, int Param)	指纹采集注册完成后的回调方法，返回采集结果 Callback method after fingerprint enrollment is completed, returns acquisition result	b: true=采集成功，false=采集失败 b: true=Acquisition success, false=Acquisition failure Param: 附加参数，返回采集失败的原因（成功时无实际意义） Param: Additional parameter returning the reason for failure (undefined if successful)	无 N/A

2.33 GrabCallBack 接口

2.33 GrabCallBack Interface

指纹图像采集回调接口，用于实时反馈指纹图像采集过程中的消息及采集结果。

Callback interface for fingerprint image acquisition, used to provide real-time feedback of messages and acquisition results during fingerprint image acquisition.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg, int codeID)	接收指纹图像采集的实时状态消息 Receive real-time status messages of fingerprint image acquisition	msg: 指纹图像采集的实时描述信息 msg: Real-time description information of fingerprint image acquisition codeID: 消息对应的状态编码 codeID: Status code corresponding to the message	无 N/A
onComplete(boolean b, int Param)	指纹图像采集完成后的回调方法，返回采集结果 Callback method after fingerprint image acquisition is completed, returns acquisition result	b: true=采集成功，false=采集失败 b: true=Acquisition success, false=Acquisition failure Param: 附加参数，返回采集失败的原因（成功时无实际意义） Param: Additional parameter returning the reason for failure (undefined if successful)	无 N/A

2.34 IdentificationCallBack 接口

2.34 IdentificationCallBack Interface

指纹验证回调接口，用于实时反馈指纹验证过程中的消息及验证结果。

Callback interface for fingerprint identification, used to provide real-time feedback of messages and identification results during fingerprint identification.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg, int codeID)	接收指纹验证的实时状态消息 Receive real-time status messages of fingerprint identification	msg: 指纹验证的实时描述信息 msg: Real-time description information of fingerprint identification codeID: 消息对应的状态编码 codeID: Status code corresponding to the message	无 N/A
onComplete(boolean b, int id, String name, int Param)	指纹验证完成后的回调方法，返回验证结果及相关信息 Callback method after fingerprint identification is completed, returns identification result and related information	b: true=验证成功, false=验证失败 b: true=Identification success, false=Identification failure id: 指纹ID (>=0=验证成功, -1=验证失败, -2=操作取消, -3=未找到指纹) id: Fingerprint ID (>=0=Identification success, -1=Identification failure, -2=Operation cancelled, -3=Fingerprint not found) name: 指纹关联的名称信息 name: Name information associated with the fingerprint Param: 附加参数，返回验证失败的原因（成功时无实际意义） Param: Additional parameter returning the reason for failure (undefined if successful)	无 N/A