Device

java.lang.Object

com.theme.finger.print.Device

All Implemented Interfaces:

ErrorCode

---

public final class **Device**
extends java.lang.Object
implements ErrorCode

All the functions of such a request for fingerprint module (USB mode and serial port mode), all methods use synchronized blocking access. To cancel the function which currently executing, you can call the method cancel() manually canceled. Due to the serial interface mode transmission fingerprint image is time-consuming, specific parameters "recvBitmap" as needed when you call the following function, Whether you need to transfer the fingerprint image: public synchronized void enroll(final boolean recvBitmap, final OnEnrollListener listen); public synchronized void Mark(final boolean recvBitmap, final OnVerifyListener listen); public synchronized void verify(final int id, final boolean recvBitmap, final OnVerifyListener listen); public synchronized void matchEx(final int id, final byte[] chars, final boolean recvBitmap, final OnMatchExListener listen);

Nested Class Summary	
static interface	Device.OnConnectionListener Device connect、disconnect callback
static interface	Device.OnEnrollListener Enroll fingerprint function enroll() callback parameters to prompt the user to operate
static interface	Device.OnMatchExListener
static interface	Device.OnRecvBitmapListener Get the fingerprint image callback
static interface	Device.OnRecvCharListener Read the fingerprint characteristic value callback
static interface	Device.OnSaveCharListener Save the fingerprint characteristic value callback
static interface	Device.OnVerifyListener Fingerprint matchs by 1:1 and 1:n on the callback connect

Field Summary	
static int	DEVICE_TYPE_UART Mark Device which is constructed by a UART communication type
static int	DEVICE_TYPE_USB Mark Device which is constructed by a USB communication type

Fields inherited from interface com.theme.fingerprint.ErrorCode
ERR_ALL_TMPL_EMPTY, ERR_BAD_QUALITY, ERR_BROKEN_ID_NOEXIST, ERR_CONNECTION, ERR_DUPLICATION_ID, ERR_EMPTY_ID_NOEXIST, ERR_FAIL, ERR_FP_CANCEL, ERR_FP_NOT_DETECTED, ERR_GEN_COUNT, ERR_MARK, ERR_INVALID_BUFFER_ID, ERR_INVALID_OPERATION_MODE, ERR_INVALID_PARAM, ERR_INVALID_TMPL_DATA, ERR_INVALID_TMPL_NO, ERR_MEMORY, ERR_MERGE_FAIL, ERR_NOT_AUTHORIZED, ERR_SUCCESS, ERR_TIME_OUT, ERR_TMPL_EMPTY, ERR_TMPL_NOT_EMPTY, ERR_VERIFY

Constructor Summary

Device(Context context, int type) Construct device examples by “type” type, all parameters have the specific default values：usb:Device(context, 0x2009, 0x7638); uart：Device(context, "/dev/ttySAC1", 115200);
Device(Context context, int usb_vid, int usb_pid) Construct Device by a usb drive
Device(Context context, java.lang.String serialPortName, int speed) Construct Device by a serial port driver

Method Summary

void	cancel() Cancel the operation currently being performed
boolean	close() Close related communication connections
boolean	connectionTest() Connectivity Test
void	destory() All operations have been completed, release relevant examples
boolean	detectFinger() Detect whether there is a fingerprint inputting status (whether the finger on the fingerprint reader or not) cmd = CMD_FINGER_DETECT(0x0021)
int	downChar(int ramBufId, byte[] templateBuf) Receive the fingerprint Template Data from Host and store in the specific Ram Buffer cmd = CMD_DOWN_CHAR(0x0043)
int	downLoadBitmap(byte[] bitmapBuf, int width, int height) Save the image data received from the Host in ImageBuffer.
void	enroll(boolean recvBitmap, Device.OnEnrollListener listen) Enroll fingerprint
int	generate(int ramBufId) Generate fingerprint template “Template” from the fingerprint image in ImageBuffer, and save in the specific Ram Buffer.
boolean	getBitmap() Collect fingerprint images cmd = CMD_GET_IMAGE(0x0020)
int	getBrokenIds(int startTemplateId, int endTemplateId,

	int[] count, int[] firstTemplateId) Check whether the enrolled template in the specific range (the beginning Template number ~ the finishing Template number) is damage or not.
void	getChar(int id, Device.OnRecvCharListener listen) Read the characteristic value data corresponding to id in the fingerprint module (498Byte)
int	getCount() Get the enrolled fingerprint count cmd = CMD_GET_ENROLL_COUNT(0x0048)
java.lang.String	getDeviceInfo() Read the device information cmd = CMD_GET_DEVICE_INFO(0x0004)
int	getDeviceType() Get the currently instantial type {DEVICE_TYPE_USB DEVICE_TYPE_UART}
int	getEmptyId(int startTemplateId, int endTemplateId) Get the first Template number which can be enrolled(do not enroll Template) in the specific range(the beginning Template number ~ the finishing Template number).
java.lang.String	getIdNote(int id) Get the fingerprint remarks cmd = CMD_GET_ID_NOTE(0x0007)
int	getMaxCount() Get the maximum count of fingerprints which can be enrolled
java.lang.String	getModuleSn() Get the module serial number cmd = CMD_GET_MODULE_SN(0x0009)
int	getParam(int paramIndex) Get the set parameters of fingerprint module cmd = CMD_GET_PARAM(0x0003)
static java.util.List<java.lang.Integer>	getSerialPortBaundRates() Enumerate all the serial port baund rates
static java.util.List<java.lang.String>	getSerialPortNams() Get all the serial port names on the device
int	getStatus(int templateId) Get the enrollment status of Template in the specific number
void	identify(boolean recvBitmap, Device.OnVerifyListener listen) Identify fingerprints in 1:N
boolean	isOpend() Whether the communication connection is open or not
int	loadChar(int templateId, int ramBufId) Get the specifically numbered fingerprint template from the fingerprint base and store in the specific Ram Buffer temporarily.
int	match(int ramBufId0, int ramBufId1) Match with Templates from the two specific Ram Buffer.

void	matchEx(int id, byte[] chars, boolean recvBitmap, Device.OnMatchExListener listen) Match the fingerprint characteristics of host (the external of fingerprint module) with the characteristic value data corresponding to id that in the fingerprint module by 1:1
int	merge(int ramBufId, int count) Merge the templates which temporarily store in the Ram Buffer to generate template data and store in the specific Ram Buffer.
void	open(Device.OnConnectionListener l) Open the device communication connections
void	recvBitmap(Device.OnRecvBitmapListener listen) Get the current inputting fingerprint image
boolean	removeAll() Remove all the fingerprint information cmd = CMD_DEL_CHAR(0x0044)
boolean	removeId(int id) Remove the fingerprint information corresponding to the specific id cmd = CMD_DEL_CHAR(0x0044)
int	search(int ramBufId, int startId, int endId, int[] templateId, int[] learnResult) Match the fingerprint template in the specific Ram Buffer with all enrolled fingerprint templates in the specifically searched range(the beginning template number ~ the finishing template number) by 1:N and then return the result.
void	setChar(byte[] chars, Device.OnSaveCharListener listen) Save the fingerprint characteristic values in the fingerprint module
boolean	setIdNote(int id, java.lang.String note) Set the fingerprint remarks cmd = CMD_SET_ID_NOTE(0x0006)
boolean	setModuleSn(java.lang.String moduleSn) Set the module serial number cmd = CMD_GET_MODULE_SN(0x0008)
boolean	setParam(int paramIndex, int paramValue) Set the relevant parameters, which refer to the specific protocol document cmd = CMD_SET_PARAM(0x0002)
boolean	sledCtrl(int state) Control“on/off” of backlight of collector.
int	storeChar(int templateId, int ramBufId, int[] duplicationId) Save the template stored in the specific Ram Buffer in the specifically numbered fingerprint template base.
int	upChar(int ramBufId, byte[] templateBuf) Sent the Template in the specific Ram Buffer to Host.
int	upLoadBitmap(int type, byte[] bitmapBuf, int[] width, int[] height) Sent the image saved in the ImageBuffer to Host according to the specific Image Type.
void	verify(int id, boolean recvBitmap, Device.OnVerifyListener listen) Identify fingerprint in 1:1

int	verify(int templateId, int ramBufId, int[] learnResult) Match the template in the specific Ram Buffer with the specially numbered template from the database by 1:1 and then return the result.

Methods inherited from " java.lang.Object"
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Details

DEVICE_TYPE_USB

public static final int DEVICE_TYPE_USB

Mark Device which is constructed by a USB communication type

Please refer to:
Constant Field Values

DEVICE_TYPE_UART

public static final int DEVICE_TYPE_UART

Mark Device which is constructed by a UART communication type

Please refer to:
Constant Field Values

Construction method details

Device

public Device(Context context,int type)

Construct Device examples by "type" type, all parameters have the specific default values:
usb:Device(context, 0x2009, 0x7638); uart: Device(context, "/dev/ttySAC1", 115200);

Parameters:
context -
type: {DEVICE_TYPE_USB, - DEVICE_TYPE_UART}

Device

public Device(Context context, int usb_vid,int usb_pid)

Construct Device by a USB drive

Parameters:

context -

usb_vid -

usb_pid -

Device

```
public Device(Context  context, java.lang.String  serialPortName,
              int  speed)
```

Construct Device by a serial port driver

Parameters:

context -

serialPortName -

speed –

Method details

Get Device Type

```
public int getDeviceType()
```

Get the currently instantial type{DEVICE_TYPE_USB， DEVICE_TYPE_UART}

Return:

{DEVICE_TYPE_USB， DEVICE_TYPE_UART}

---

open

```
public void open(Device.OnConnectionListener l)
```

Open the device communication connections

---

close

```
public boolean close()
```

Close the related communication connections

Return:

true: Close successfully, otherwise, the operation fails

---

destory

```
public void destory()
```

All operations have been completed, release the relevant examples

---

cancel

```
public void cancel()
```

Cancel the operation currently being performed

---

isOpend

public boolean **is Opend**()
Whether the communication connection is open or not
Return:

Connection Test

public boolean **connectionTest**()
Connectivity Test
Return:

---

enroll

public void **enroll**(boolean recvBitmap,
Device.OnEnrollListener listen)
Enroll fingerprint
Parameters:
recvBitmap:Indicate whether need to return the collected fingerprint image through onBitmap()-
listen -

---

identify

public void **identify**(boolean recvBitmap,
Device.OnVerifyListener listen)
Identify fingerprint in 1:N
Parameters:
recvBitmap: Indicate whether need to return the collected fingerprint image through onBitmap()-
listen -

---

verify

public void **verify**(int id,
boolean recvBitmap,
Device.OnVerifyListener listen)
Verify fingerprint in 1:1
Parameters:
id: the fingerprint id needed to verify -
recvBitmap: Indicate whether need to return the collected fingerprint image through onBitmap()-
listen -

recvBitmap

public void **recvBitmap**(Device.OnRecvBitmapListener listen)
Get the current inputting fingerprint image
Parameters:
listen -

Get Char

```
public void getChar(int  id,
                    Device.OnRecvCharListener listen)
```

Read the characteristic value data corresponding to id that in the fingerprint module (498Byte)

Parameters:

id -

---

setChar

```
public void setChar(byte[]  chars,
                    Device.OnSaveCharListener listen)
```

Save the fingerprint characteristic values in the fingerprint module

Parameters:

id -

chars -

Match Ex

```
public void match Ex(int  id,
                     byte[]  chars,
                     boolean  recvBitmap,
                     Device.OnMatchExListener listen)
```

Match the fingerprint characteristic values of host (the external of fingerprint module) with the characteristic values corresponding to id that in the fingerprint module by 1:1

Parameters:

id: the required comparative characteristic value id -

recvBitmap: Indicate whether need to return the collected fingerprint image through on Bitmap()-

chars:The fingerprint characteristic value data in host-

Remove Id

```
public boolean remove Id(int  id)
```

Remove the fingerprint information corresponding to the specific id cmd = CMD_DEL_CHAR (0x0044)

Parameters:

id -

Return:

Whether the operation succeeds or not

Remove All

```
public final boolean remove All()
```

Remove all the fingerprint information cmd = CMD_DEL_CHAR(0x0044)

Return:

Whether the operation succeeds or not

Get Count

```
public int get Count()
```

Get the enrolled fingerprint number cmd = CMD_GET_ENROLL_COUNT(0x0048)

Return:
Return -1 if get failure, otherwise return the actual number

Get Max Count

```
public int get Max Count()
```
Get the maximum count of fingerprints that can be enrolled
Return:

---

setParam

```
public boolean set Param(int  paramIndex,
                         int  paramValue)
```
Set the relevant parameters, which refer to the specific protocol document cmd = CMD_SET_PARAM(0x0002)
Parameters:
paramIndex -
paramValue -
Return:

Get Param

```
public int get Param(int  paramIndex)
```
Get the related setting parameters of fingerprint module cmd = CMD_GET_PARAM(0x0003)
Parameters:
paramIndex -
Return:
Failure to get the marks if less than 0, otherwise return the specific parameter values

---

Get Device Info

```
public java.lang.String get Device Info()
```
Read device information cmd = CMD_GET_DEVICE_INFO(0x0004)
Return:

Set Id Note

```
public boolean set Id Note(int  id,
                           java.lang.String  note)
```
Set the fingerprint remarks cmd = CMD_SET_ID_NOTE(0x0006)
Parameters:
id -
note - <= 16byte
Return:

Get Id Note

```
public java.lang.String get Id Note(int  id)
```
Get the fingerprint remarks cmd = CMD_GET_ID_NOTE(0x0007)
Parameters:
id -

Return:

Set Module Sn

```
public boolean set Module Sn(java.lang.String  moduleSn)
```

Set the module serial number cmd = CMD_GET_MODULE_SN(0x0008)

Parameters:

Module Sn -

Return:

Get Module Sn

```
public java.lang.String get Module Sn()
```

Get the module serial number cmd = CMD_GET_MODULE_SN(0x0009)

Return:

Get Bitmap

```
public boolean get Bitmap()
```

Collect the fingerprint image cmd = CMD_GET_IMAGE(0x0020)

Return:

Detect Finger

```
public boolean detect Finger()
```

Detect whether there is a fingerprint input status (whether finger on the fingerprint reader or not) cmd = CMD_FINGER_DETECT(0x0021)

Return:

---

upload Bitmap

```
public int upload Bitmap(int  type,
                         byte[]  bitmapBuf,
                         int[]  width,
                         int[]  height)
```

Sent the image saved in the Image Buffer to Host according to the specific Image Type. If the Image Type is 0, then send the whole map; if the Image Type is 1, then sent the 1/4 image (take a point from four points).cmd = CMD_UP_IMAGE(0x0022)

Parameters:

type -

bitmapBuf -

width -

height -

Return:

Return the error code

download Bitmap

```
public int download Bitmap(byte[]  bitmapBuf,
                           int  width,
                           int  height)
```

Save the image data received from the Host in Image Buffer. Host sent the image to Target by 496bytes unit. cmd = CMD_DOWN_IMAGE(0x0023)

Parameters:

bitmapBuf -

width -

height -

Return:

Sled Ctrl

```
public boolean sled Ctrl(int  state)
```

Control"on/off" of backlight of collector. cmd = CMD_SLED_CTRL(0x0024)

Parameters:

state -

Return:

Store Char

```
public int store Char(int  templateId,
                      int  ramBufId,
                      int[]  duplicationId)
```

Save the templates stored in the specific Ram Buffer in the specifically numbered fingerprint template base. cmd = CMD_STORE_CHAR(0x0040)

Parameters:

templateId -

ramBufId -

duplicationId -

Return:

Load Char

```
public int load Char(int  templateId,
                     int  ramBufId)
```

Get the specifically numbered fingerprint template from the fingerprint base and store in the specific Ram Buffer temporarily. cmd = CMD_LOAD_CHAR(0x0041)

Parameters:

templateId -

ramBufId -

Return:

Up Char

```
public int up Char(int  ramBufId,
                   byte[]  templateBuf)
```

Sent the Template in the specific Ram Buffer to Host.
cmd = CMD_UP_CHAR(0x0042)

Parameters:

ramBufId -

templateBuf -

Return:

Down Char

public int **down Char**(int ramBufId,
byte[] templateBuf)

Receive the fingerprint Template Data from Host and store in the specific Ram Buffer. cmd = CMD_DOWN_CHAR(0x0043)

Parameters:

ramBufId -

templateBuf -

Return:

Get Empty Id

public int **get Empty Id**(int start Template Id,
int end Template Id)

Get the first Template number which can be enrolled(do not enroll Template) in the specific range (the beginning Template number ~ the finishing Template number). cmd = CMD_GET_EMPTY_ID(0x0045)

Parameters:

Start Template Id -

End Template Id -

Return:

Return the relevant id, if less than 0, means get failure.

Get Status

public int **get Status**(int template Id)

Get the enroll status of Template in the specific number. cmd = CMD_GET_STATUS(0x0046)

Parameters:

Template Id -

Return:

Get Broken Ids

public int **get Broken Ids**(int start Template Id,
int end Template Id,
int[] count,
int[] first Template Id)

Check whether the enrolled template in the specific range (the beginning Template number ~ the finishing Template number) is damage or not. cmd = CMD_GET_BROKEN_ID(0x0047)

Parameters:

Start Template Id -

End Template Id -

count -

first Template Id -

Return:

---

generate

public int **generate**(int ramBuf Id)

Generate fingerprint template "Template" from the fingerprint image in ImageBuffer, and save in the specific Ram Buffer. cmd = CMD_GENERATE(0x0060)

Parameters:

ramBufId -
Return:

---

merge

```
public int merge(int  ramBufId,
                 int  count)
```

Merge the templates which temporarily store in the Ram Buffer to generate template data and store in the specific Ram Buffer. The merge count can be 2 or 3: It will be generated to Template of Ram Buffer0 and Ram Buffer1 if the count is 2. It will be generated toTemplate of Ram Buffer0、Ram Buffer1 and Ram Buffer2 if the count is 3. cmd = CMD_MERGE(0x0061)

Parameters:

ramBufId -
count -

Return:

---

match

```
public int match(int  ramBufId0,
                 int  ramBufId1)
```

Match with Templates from the two specific Ram Buffer. cmd = CMD_MATCH(0x0062)

Parameters:

ramBufId0 -
ramBufId1 -

Return:

---

search

```
public int search(int  ramBufId,
                  int  startId,
                  int  endId,
                  int[]  templateId,
                  int[]  learnResult)
```

Match in 1:N of the fingerprint template in the specific Ram Buffer and all enrolled fingerprint templates in the specifically searched range(the beginning template number ~ the finishing template number） and then return the result. cmd = CMD_SEARCH(0x0063)

Parameters:

ramBufId -
startId -
endId -
templateId -
learnResult -

Return:

---

verify

```
public int verify(int  templateId,
                  int  ramBufId,
                  int[]  learnResult)
```

Match in 1:1 of the template in the specific Ram Buffer and the specially numbered template from the database and then return the result. cmd = CMD_VERIFY(0x0064)

Parameters:
templateId -
ramBufId -
learnResult -
Return:

Get Serial Port Nams

```
public static java.util.List<java.lang.String> getSerialPortNams()
```
Get all the serial port names on the device
Return:

Get SerialPort BaundRates

```
public static java.util.List<java.lang.Integer> getSerialPortBaundRates()
```
Enumerate all the serial port baund rates

Soft Packages com.theme.finger.print

Connection Summary	
Device.OnConnectionListener	Device connect, disconnect callback
Device.OnEnrollListener	Enroll fingerprint function enroll() callback parameters to prompt the user to operate
Device.OnMatchExListener	
Device.OnRecvBitmapListener	Get the fingerprint image callback
Device.OnRecvCharListener	Read the fingerprint characteristic values callback
Device.OnSaveCharListener	Save the fingerprint characteristic values callback
Device.OnVerifyListener	Fingerprints match in 1:1 and 1:n on the callback interface
ErrorCode	

Class Summary	
Device	All the functions of such a request for fingerprint module (USB mode and serial ports mode), all methods use synchronized blocking access.

1. Interface Device.OnConnectionListener

Inclusive category:

Device

---

public static interface Device.OnConnectionListener

Device connect, disconnect callback

Method Summary	
void	onConnected() Device connect successfully
void	onDisConnected() Disconnect, or not successfully connect to the device
void	onNotFound() No find the specific device
void	onPermissionDenied() Find the device, but the device permissions do not pass

Method Details

onConnected

void onConnected()

Device connect successfully

---

onDisConnected

void onDisConnected()

Disconnect, or not successfully connect to the device

---

onPermissionDenied

void onPermissionDenied()

Find the device, but the device permissions do not pass

---

onNotFound

void onNotFound()

No find the specific device

2. Interface Device.OnEnrollListener

Inclusive category:
Device

---

public static interface Device.OnEnrollListener

Enroll fingerprint function enroll() callback parameters to prompt the user to operate

---

Method Summary	
void	onEnrollFinished(int id) Finish enrollment
void	onEnrollStep(int step, int count, boolean retry)

Method Details

onEnrollStep

```
void onEnrollStep(int  step,
                  int  count,
                  boolean  retry)
```

Parameters:
step: - The current number of verification
count: Total number of times that require verification -
retry: Whether need to restart due to the acquisition of quality problems -

---

onEnrollFinished

```
void onEnrollFinished(int  id)
```

Finish enrollment

Parameters:
id - : Return the enrolled id

3. Interface Device.OnMatchExListener

All super interfaces:
Device.OnRecvBitmapListener

Inclusive category:
Device

---

public static interface Device.OnMatchExListener
extends Device.OnRecvBitmapListener

Method Summary	
void	onMatchSuccess(int id) Match success

Methods inherited from interface com.theme.finger.print.Device.OnRecvBitmapListener
onEndRecv, onStartRecv

Method Details

onMatchSuccess

```
void onMatchSuccess(int id)
```

Match success

Parameters:

id -

4. Interface Device.OnRecvBitmapListener

All known subinterfaces:

Device.OnMatchExListener

Inclusive category:

Device

```
public static interface Device.OnRecvBitmapListener
```

Get the fingerprint image callback

Method Summary	
void	onEndRecv() Tip the ending of inputting fingerprint
void	onStartRecv() Tip inputting fingerprint

Method Detail

onStartRecv

```
void onStartRecv()
```

Tip inputting fingerprint

---

onEndRecv

```
void onEndRecv()
```

Tip the ending of inputting fingerprint

5.Interface Device.OnRecvCharListener

Inclusive category:

Device

```
public static interface Device.OnRecvCharListener
```

Read the fingerprint characteristic values callback

Method Summary	
void	onRecvChar(int id, byte[] chars) Return to read the fingerprint characteristic value datas

Method Detail

onRecvChar

```
void onRecvChar(int id,
                byte[] chars)
```

Return to read the fingerprint characteristic value datas

6.Interface Device.OnSaveCharListener

Inclusive category:

Device

```
public static interface Device.OnSaveCharListener
```

Save the fingerprint characteristic values callback

Method Summary	
void	onSaveCharSuccess(int id) Save the fingerprint characteristic values successfully

Method Details

onSaveCharSuccess

```
void onSaveCharSuccess(int id)
```

Save the fingerprint characteristic values successfully

Parameters:
id -

7.Interface Device.OnVerifyListener

Inclusive category:

Device

---

public static interface Device.OnVerifyListener

Fingerprint matchs by 1: 1 and 1: n on the callback interface

---

Method Summary

void	onEndVerify(int index) Tip the ending of inputting fingerprint
void	onStartVerify(int index) Tip inputting fingerprint
void	onVerifySuccess(int id, boolean updated, int useTime) Verify succeed

Method Details

onStartVerify

void onStartVerify(int index)
Tip inputting fingerprint
Parameters:
index: - The currently verified serial number

---

onEndVerify

void onEndVerify(int index)
Tip the ending of inputting fingerprint
Parameters:
index: - The currently verified serial number

---

onVerifySuccess

void onVerifySuccess(int id,
boolean updated,
int useTime)
Verify succeed
Parameters:
id: If not found the verified fingerprint id, return -1 -
updated: Whether to update the fingerprint database about the current collected fingerprint -
useTime:Time consuming -

8.Interface ErrorCode

All known implementing classes:

Device

public interface ErrorCode

Field Summary	
static int	ERR_ALL_TMPL_EMPTY Do not exist enrolled Template
static int	ERR_BAD_QUALITY The fingerprint image quality is not good
static int	ERR_BROKEN_ID_NOEXIST Do not exist damaged Template
static int	ERR_CONNECTION Device is disconnected
static int	ERR_DUPLICATION_ID The fingerprint has been enrolled
static int	ERR_EMPTY_ID_NOEXIST Do not exist the Template ID that can be enrolled
static int	ERR_FAIL Instruction processing fails
static int	ERR_FP_CANCEL Cancel manually
static int	ERR_FP_NOT_DETECTED No fingerprints input on the sensor
static int	ERR_GEN_COUNT The fingerprint merged count is invalid
static int	ERR_IDENTIFY It has been matched by 1: N, but the same Templates do not exist
static int	ERR_INVALID_BUFFER_ID Buffer ID is incorrect
static int	ERR_INVALID_OPERATION_MODE
static int	ERR_INVALID_PARAM Using an incorrect parameter
static int	ERR_INVALID_TMPL_DATA The specific Template Data is invalid
static int	ERR_INVALID_TMPL_NO The marked Template Data is invalid
static int	ERR_MEMORY External Flash programming errors
static int	ERR_MERGE_FAIL Template merge fails
static int	ERR_NOT_AUTHORIZED No communication password confirmation

static int	ERR_SUCCESS Instruction processing succeeds
static int	ERR_TIME_OUT No fingerprint inputting in the TimeOut time
static int	ERR_TMPL_EMPTY No enrolled Template exists in specific number
static int	ERR_TMPL_NOT_EMPTY Template exists in the specific number
static int	ERR_VERIFY Failure match with Templates in the specific number by 1:1

Field Details

ERR_SUCCESS

static final int ERR_SUCCESS
Instruction processing succeeds
Please refer to:
Constant Field Values

ERR_FAIL

static final int ERR_FAIL
Instruction processing fails
Please refer to:
Constant Field Values

ERR_CONNECTION

static final int ERR_CONNECTION
Device is disconnected
Please refer to:
Constant Field Values

ERR_VERIFY

static final int ERR_VERIFY
Failure match with Templates in the specific number by 1:1
Please refer to:
Constant Field Values

ERR_IDENTIFY

static final int ERR_IDENTIFY
It has been matched by 1: N, but the same Templates do not exist
Please refer to:
Constant Field Values

ERR_TMPL_EMPTY

static final int **ERR_TMPL_EMPTY**
No enrolled Template exists in specific number
Please refer to:
Constant Field Values

ERR_TMPL_NOT_EMPTY

static final int **ERR_TMPL_NOT_EMPTY**
Template exists in the specific number
Please refer to:
Constant Field Values

ERR_ALL_TMPL_EMPTY

static final int **ERR_ALL_TMPL_EMPTY**
Do not exist the enrolled Template
Please refer to:
Constant Field Values

ERR_EMPTY_ID_NOEXIST

static final int **ERR_EMPTY_ID_NOEXIST**
Do not exist the Template ID that can be enrolled
Please refer to:
Constant Field Values

ERR_BROKEN_ID_NOEXIST

static final int **ERR_BROKEN_ID_NOEXIST**
Do not exist the damaged Template
Please refer to:
Constant Field Values

ERR_INVALID_TMPL_DATA

static final int **ERR_INVALID_TMPL_DATA**
The specific Template Data is invalid
Please refer to:
Constant Field Values

ERR_DUPLICATION_ID

static final int **ERR_DUPLICATION_ID**
The fingerprint has been enrolled
Please refer to:
Constant Field Values

ERR_BAD_QUALITY

static final int **ERR_BAD_QUALITY**
The fingerprint image quality is not good

Please refer to:
Constant Field Values

ERR_MERGE_FAIL

`static final int` **ERR_MERGE_FAIL**
Template merge fails
Please refer to:
Constant Field Values

ERR_NOT_AUTHORIZED

`static final int` **ERR_NOT_AUTHORIZED**
No communication password confirmation
Please refer to:
Constant Field Values

ERR_MEMORY

`static final int` **ERR_MEMORY**
External Flash programming errors
Please refer to:
Constant Field Values

ERR_INVALID_TMPL_NO

`static final int` **ERR_INVALID_TMPL_NO**
The marked template id is invalid
Please refer to:
Constant Field Values

ERR_INVALID_PARAM

`static final int` **ERR_INVALID_PARAM**
Using an incorrect parameter
Please refer to:
Constant Field Values

ERR_TIME_OUT

`static final int` **ERR_TIME_OUT**
No fingerprint inputting in the TimeOut time
Please refer to:
Constant Field Values

ERR_GEN_COUNT

`static final int` **ERR_GEN_COUNT**
The fingerprint merged count is invalid
Please refer to:
Constant Field Values

ERR_INVALID_BUFFER_ID

`static final int` **ERR_INVALID_BUFFER_ID**
Buffer ID is incorrect
Please refer to:
Constant Field Values

ERR_INVALID_OPERATION_MODE

`static final int` **ERR_INVALID_OPERATION_MODE**
Please refer to:
Constant Field Values

ERR_FP_NOT_DETECTED

`static final int` **ERR_FP_NOT_DETECTED**
No fingerprints input on the sensor
Please refer to:
Constant Field Values

ERR_FP_CANCEL

`static final int` **ERR_FP_CANCEL**
Cancel manually
Please refer to:
Constant Field Values

Constant Field Values

Contents

com.theme.*

com.theme.finger.print.Device

public	static	final	int	DEVICE_TYPE_UART	2
public	static	final	int	DEVICE_TYPE_USB	1

com.theme.finger.print.ErrorCode

public	static	final	int	ERR_ALL_TMPL_EMPTY	20
public	static	final	int	ERR_BAD_QUALITY	25
public	static	final	int	ERR_BROKEN_ID_NOEXIST	22
public	static	final	int	ERR_CONNECTION	2
public	static	final	int	ERR_DUPLICATION_ID	24
public	static	final	int	ERR_EMPTY_ID_NOEXIST	21
public	static	final	int	ERR_FAIL	1
public	static	final	int	ERR_FP_CANCEL	65
public	static	final	int	ERR_FP_NOT_DETECTED	40
public	static	final	int	ERR_GEN_COUNT	37
public	static	final	int	ERR_IDENTIFY	17
public	static	final	int	ERR_INVALID_BUFFER_ID	38
public	static	final	int	ERR_INVALID_OPERATION_MODE	39
public	static	final	int	ERR_INVALID_PARAM	34
public	static	final	int	ERR_INVALID_TMPL_DATA	23
public	static	final	int	ERR_INVALID_TMPL_NO	29
public	static	final	int	ERR_MEMORY	28
public	static	final	int	ERR_MERGE_FAIL	26
public	static	final	int	ERR_NOT_AUTHORIZED	27
public	static	final	int	ERR_SUCCESS	0
public	static	final	int	ERR_TIME_OUT	35
public	static	final	int	ERR_TMPL_EMPTY	18
public	static	final	int	ERR_TMPL_NOT_EMPTY	19
public	static	final	int	ERR_VERIFY	16