QuickStart Guide

Table of Contents

1.Introduction

Welcome to use BioID fingerprint SDK. This document will introduce how to use Fingerprint SDK for development on Android.

2.SDK Contents

The SDK contains libraries and demo needed for the development of SM-9X,Contains the following directories:

- /apk or /bin

  The `apk` or `bin` folder contains the compiled binary files, which can be installed directly on the Android device.

- /demo

  The `demo` folder is the source code corresponding to the apk file, which can be opened using Android Studio .

> NOTE:
>
> The library (.so & .jar & .aar) file is included in the demo and is not provided separately.

3.Development Guide

3.1 Configure your project

This section shows you how to create a new Android project using Android Studio and introduces some of the files in the project.

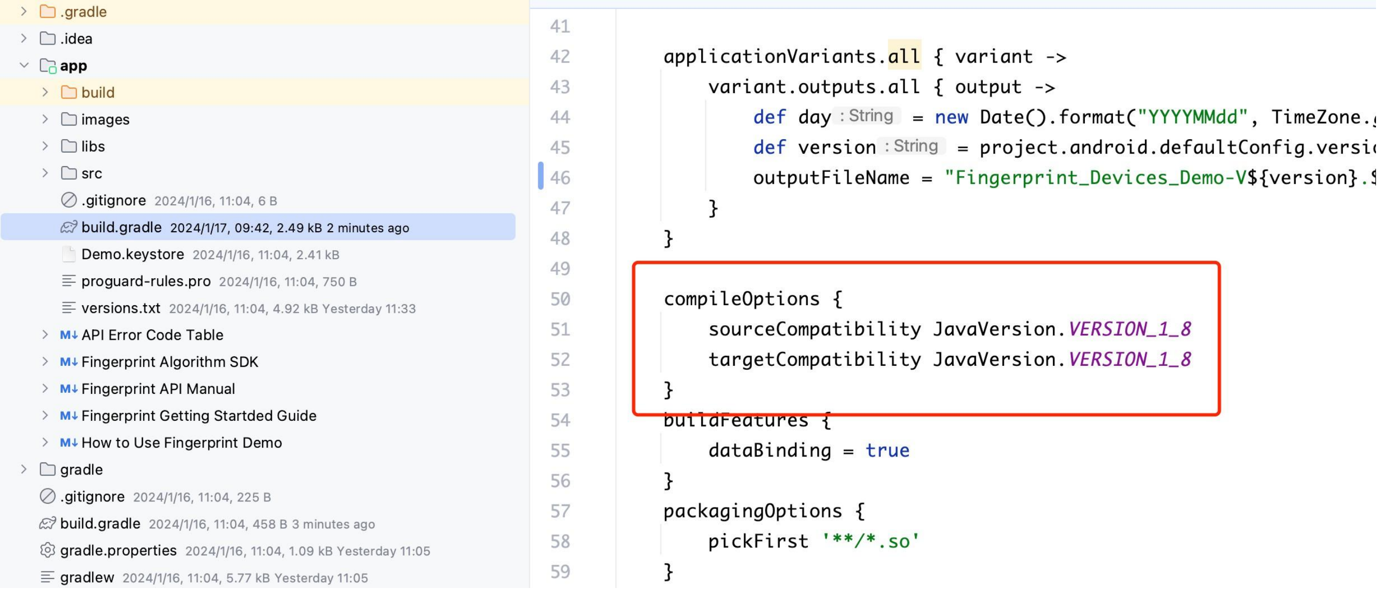
To create a new Android project, follow these steps:

3.1.1 Create a new Android project

Create a new Empty Activity application project,You can refer to Android Developer to create an Android project.

3.1.2 Enable Java 8

The SDK needs to work with **JDK 1.8**, open **build.gradle**, add a configuration inside, as shown in the figure:

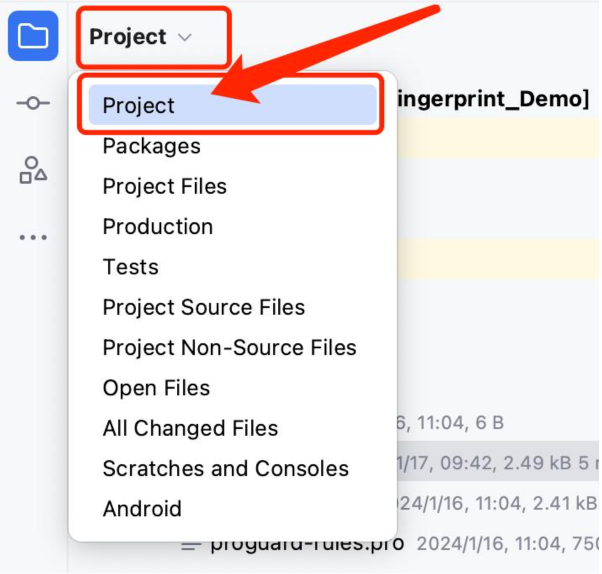


```
> .gradle
> .idea
v app
  > build
  > images
  > libs
  > src
    .gitignore  2024/1/16, 11:04, 6 B
    build.gradle  2024/1/17, 09:42, 2.49 kB 2 minutes ago
    Demo.keystore  2024/1/16, 11:04, 2.41 kB
    proguard-rules.pro  2024/1/16, 11:04, 750 B
    versions.txt  2024/1/16, 11:04, 4.92 kB Yesterday 11:33
  > API Error Code Table
  > Fingerprint Algorithm SDK
  > Fingerprint API Manual
  > Fingerprint Getting Startded Guide
  > How to Use Fingerprint Demo
> gradle
  .gitignore  2024/1/16, 11:04, 225 B
  build.gradle  2024/1/16, 11:04, 458 B 3 minutes ago
  gradle.properties  2024/1/16, 11:04, 1.09 kB Yesterday 11:05
  gradlew  2024/1/16, 11:04, 5.77 kB Yesterday 11:05
```

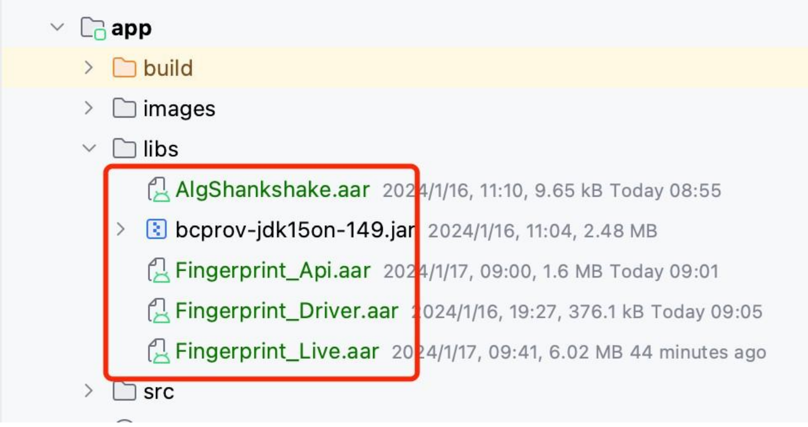
```
    applicationVariants.all { variant ->
        variant.outputs.all { output ->
            def day : String = new Date().format("YYYYMMdd", TimeZone.g
            def version : String = project.android.defaultConfig.versi
            outputFileName = "Fingerprint_Devices_Demo-V${version}.$
        }
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    buildFeatures {
        dataBinding = true
    }
    packagingOptions {
        pickFirst '**/*.so'
    }
```

3.1.3 Add aar file

First, switch to the project view. Click on `Android` and select `Project`.As shown below:



Then, copy the aar from the demo project to the lib directory of your project, as shown below:



```
app
  build
  images
  libs
    AlgShankshake.aar  2024/1/16, 11:10, 9.65 kB Today 08:55
    bcprov-jdk15on-149.jar  2024/1/16, 11:04, 2.48 MB
    Fingerprint_Api.aar  2024/1/17, 09:00, 1.6 MB Today 09:01
    Fingerprint_Driver.aar  2024/1/16, 19:27, 376.1 kB Today 09:05
    Fingerprint_Live.aar  2024/1/17, 09:41, 6.02 MB 44 minutes ago
  src
```

> NOTE:
>
> `Fingerprint_Api.aar` is used for fingerprint algorithm.
>
> `Fingerprint_Driver.aar` is used to access `bioID` Device.

3.1.4 Congratulations

Congratulations, you have completed all the preparations, then you can refer to the **Sample Code** to learn how to use our API.

3.2 Sample Code

3.2.1 Driver API

Get a instance of driver API :

```
SM92MSVcApi mDriverApi = FingerApiFactory.getInstance(getApplicationContext(),
FingerApiFactory.USB);
// do something with mDriverApi
```

Open/Close SM-92M-S device :

```
int fd =mDriverApi.openDevice();
if (fd >= 0) {
  // Open successfully !
  // 0 means it has been opened before
} else {
  // process error code
}
```

Capture fingerprint image from bioID device:

```
CaptureConfig captureConfig = new CaptureConfig.Builder()
                .setLfdLevel(0)
                .setLatentLevel(0)
                .setTimeout(8000)
                .setAreaScore(45)
                .setPreviewCallBack(previewCallBack)
                // AES/ECB/PKCS5Padding
                //.setAESConfig(new
AESConfig.Builder().setKey("1234567890123456").build())
                //.setAESStatus(CaptureConfig.AES_HOST)
                .build();
MxResult<MxImage> image = mDriverApi.getImage(captureConfig);
if (imageResult.isSuccess()) {
  // Capture successfully !
  MxImage mxImage = imageResult.data;
  // Convert raw image to Android bitmap
  byte[] imageDate = new byte[mxImage.width * mxImage.height + 1078];
  BmpLoader.Raw2Bmp(imageDate, mxImage.data, mxImage.width, mxImage.height);
  Bitmap bitmap = BitmapFactory.decodeByteArray(imageDate, 0, imageDate.length);
  //show bitmap
  ImageView imageView ;//The imageView to show fingerprint image
  imageView.setImageBitmap(bitmap);
} else {
  //process error code
}
```

Refer to the **API Manual** for more information about driver api.

3.2.2 FingerAlg Api

Create a instance of FingerAlg Api :

```
FingerAlgAPI mFingerAlgApi = new FingerAlgAPI();
```

Create FMR :

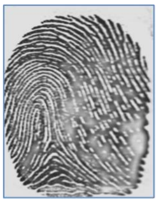
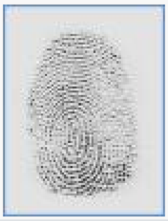
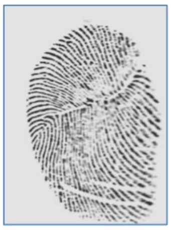
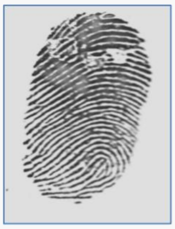
```
MxImage image ; // capture from Finger Device
byte[] fmrBuffer = new byte[1024];//Must be 1024 bytes
int result = mFingerAlgApi.createTemplateISO(image.data, image.width, image.height,
fmrBuffer);
if (result >= 0) {
  // successfully
}
// If you use ISO2011, please call the following function :
// mFingerAlgApi.createTemplateISO2011(...)
// If you use ANSI, please call the following function :
// mFingerAlgApi.createTemplateANSI(...)
```

Match two FMR :

```
byte[] fmrBufferA //Must be 1024 bytes
byte[] fmrBufferB //Must be 1024 bytes
int similarScore = mFingerAlgApi.compareTemplatesANSI(fmrBufferA, fmrBufferB);
if (score >= 45) { // Suggest 45 pass
  // Match passed
} else if (score >= 0) {
  // Match not passed
} else {
  // Process error code
}
```

> NOTE: When FingerAlg works with bioID reader, there is no need to call initialization
>
> Refer to the **Fingerprint Algorithm SDK.pdf** for more FingerAlg apis.

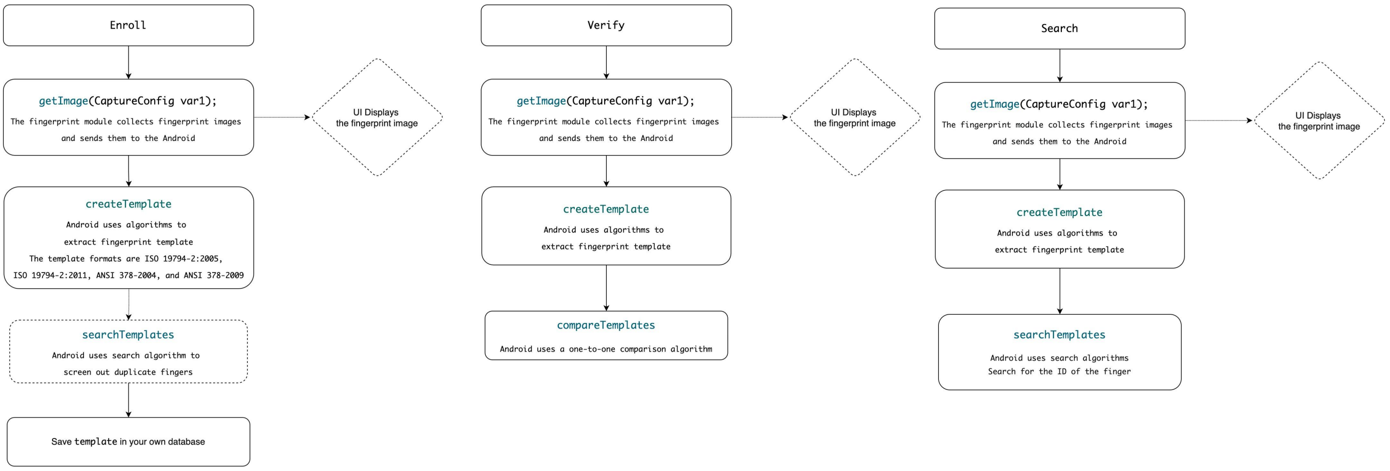
3.3 How to Get High-quality Fingerprints

Finger Type	Fingerprint Collected	How to get high-quality fingerprints
Wet finger		Need to remove the excessive moisture from your fingers and keep the sensor clean.
Dry finger		Use hand cream, wet wipes, blowing on his stiff fingers to moisturize.
Fingers have obvious scars		Use other fine fingers.
Finger peeling		Use other fine fingers.
Slight pressure	Like the image of dry fingers	Should apply moderate but steady pressure, and stay 1 second when place a finger on the fingerprint reader
Too much pressure	Like the image of wet fingers or torsion distortion	Should be relaxed and reduce the stress, apply moderate but steady pressure, and stay 1 second, about the same, pressure as pressing a button.

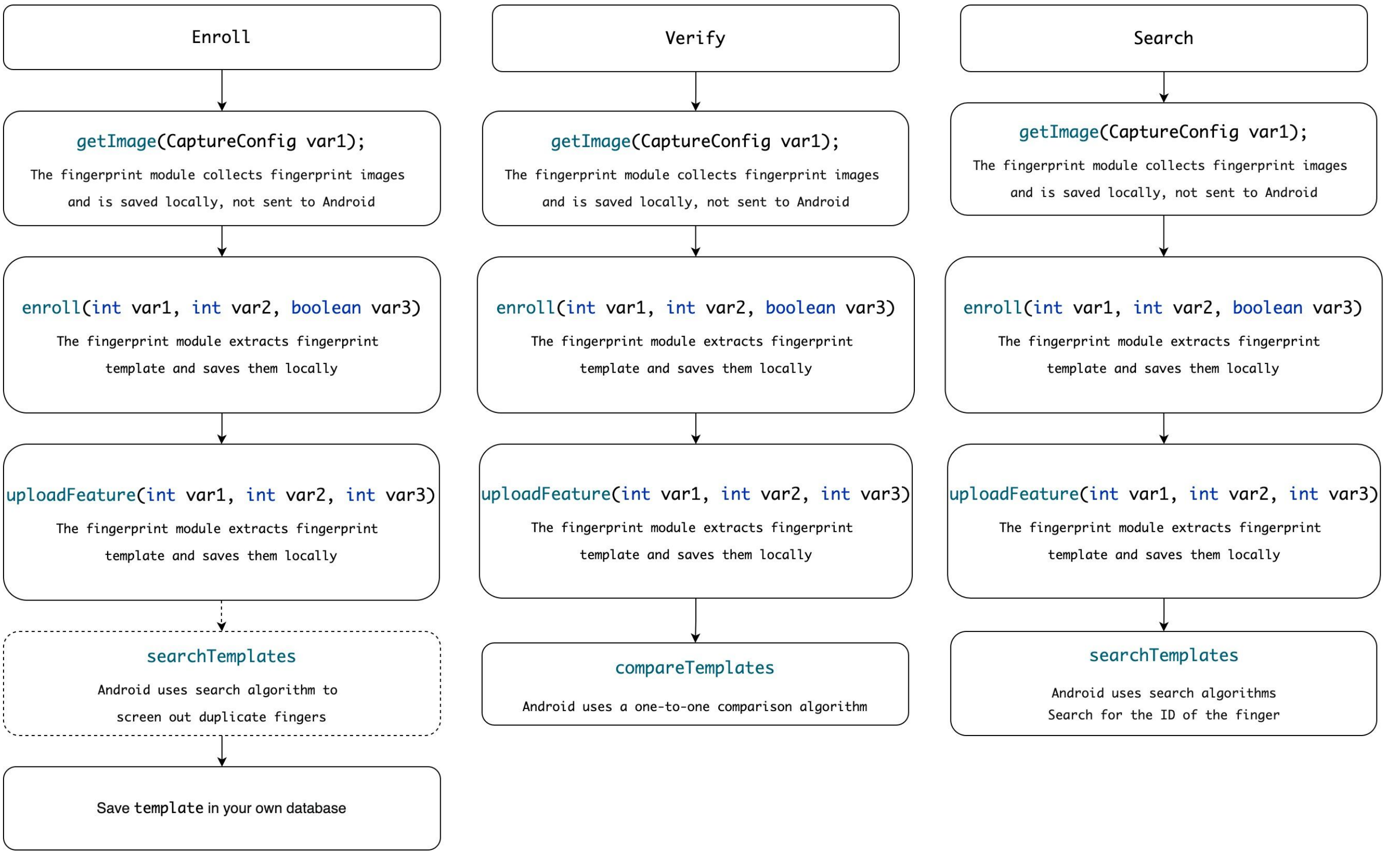
3.4 Development Reference

Fingerprint template data is stored in your database.

3.4.1 Host (USB/SPI communication)

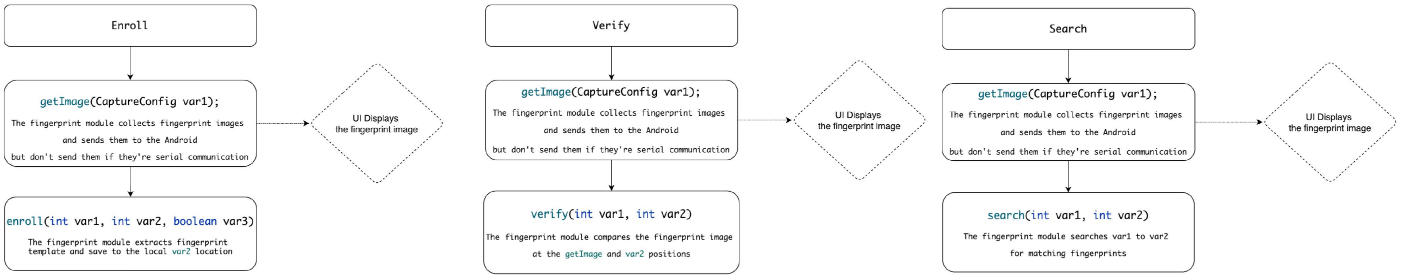

Enroll
getImage(CaptureConfig var1);
The fingerprint module collects fingerprint images
and sends them to the Android
UI Displays
the fingerprint image
createTemplate
Android uses algorithms to
extract fingerprint template
The template formats are ISO 19794-2:2005,
ISO 19794-2:2011, ANSI 378-2004, and ANSI 378-2009
searchTemplates
Android uses search algorithm to
screen out duplicate fingers
Save template in your own database
Verify
getImage(CaptureConfig var1);
The fingerprint module collects fingerprint images
and sends them to the Android
UI Displays
the fingerprint image
createTemplate
Android uses algorithms to
extract fingerprint template
compareTemplates
Android uses a one-to-one comparison algorithm
Search
getImage(CaptureConfig var1);
The fingerprint module collects fingerprint images
and sends them to the Android
UI Displays
the fingerprint image
createTemplate
Android uses algorithms to
extract fingerprint template
searchTemplates
Android uses search algorithms
Search for the ID of the finger

3.4.2 Host (Serial communication)


Enroll
getImage(CaptureConfig var1);
The fingerprint module collects fingerprint images
and is saved locally, not sent to Android
enroll(int var1, int var2, boolean var3)
The fingerprint module extracts fingerprint
template and saves them locally
uploadFeature(int var1, int var2, int var3)
The fingerprint module extracts fingerprint
template and saves them locally
searchTemplates
Android uses search algorithm to
screen out duplicate fingers
Save template in your own database
Verify
getImage(CaptureConfig var1);
The fingerprint module collects fingerprint images
and is saved locally, not sent to Android
enroll(int var1, int var2, boolean var3)
The fingerprint module extracts fingerprint
template and saves them locally
uploadFeature(int var1, int var2, int var3)
The fingerprint module extracts fingerprint
template and saves them locally
compareTemplates
Android uses a one-to-one comparison algorithm
Search
getImage(CaptureConfig var1);
The fingerprint module collects fingerprint images
and is saved locally, not sent to Android
enroll(int var1, int var2, boolean var3)
The fingerprint module extracts fingerprint
template and saves them locally
uploadFeature(int var1, int var2, int var3)
The fingerprint module extracts fingerprint
template and saves them locally
searchTemplates
Android uses search algorithms
Search for the ID of the finger

3.4.3 Module

The fingerprint template data is stored in the fingerprint module.



Enroll

getImage(CaptureConfig var1);

The fingerprint module collects fingerprint images and sends them to the Android but don't send them if they're serial communication

enroll(int var1, int var2, boolean var3)

The fingerprint module extracts fingerprint template and save to the local var2 location

Verify

getImage(CaptureConfig var1);

The fingerprint module collects fingerprint images and sends them to the Android but don't send them if they're serial communication

verify(int var1, int var2)

The fingerprint module compares the fingerprint image at the getImage and var2 positions

Search

getImage(CaptureConfig var1);

The fingerprint module collects fingerprint images and sends them to the Android but don't send them if they're serial communication

search(int var1, int var2)

The fingerprint module searches var1 to var2 for matching fingerprints

UI Displays the fingerprint image
UI Displays the fingerprint image
UI Displays the fingerprint image