

KOZEN Financial

SDK Development Documentation

V2.1

Shanghai Xiangcheng Communication Technology Co., Ltd.

Table of Contents

- 1. Revision History 9
- 2. Overview 14

2.1 Introduction 14
2.2 Android version and IDE version supported by the SDK 14
2.3 Feature Introduction 14
2.3.1 Financial SDK Engine Module 14
2.3.2 Cardreader Module 14
2.3.3 EMV Module 14
2.3.4 General Module 15
2.3.5 Pinpad Module 15
2.3.6 Printer Module 15
2.3.7 Scanner Module 15
2.3.8 Security Module 15
2.3.9 ECR Module 16
2.4 Importing the Financial SDK 16
2.5 Initializing the Financial SDK 17

- 3. API Interface Introduction 18

➢ 3.1 Financial SDK initialization 18
3.1.1 Initialize the FinancialService instance 18
3.1.2 Initialization results 19
➢ 3.2 Scanner Operation module 19
-- Get scanner module - getScannerManager -- 19
3.2.1 Open the scanner module 20
3.2.2 Open the scanner module with a specific camera 20
3.2.3 Close the scanner module 20
3.2.4 Register a scanner result callback 20
3.2.5 Start scanning 21
3.2.6 Stop scanning 21
3.2.7 Enable or disable specific barcode types 21
3.2.8 Enable or disable all barcode types 21
3.2.9 Check if a barcode type is enabled 22
3.2.10 Toggle the flashlight 22
3.2.11 Enable or disable auto-focus (AF) mode and set fixed focus distance 22
3.2.12 Start camera scan and preview rendering to a SurfaceTexture 22
3.2.13 Trigger the decode action 23
3.2.14 Stop the decode action 23
3.2.15 Decode an image 23
3.2.16 Set zoom scale 23
3.2.17 Receive scan result and symbology type 24
3.2.18 Called when the connection to the scanner service is established 24
3.2.19 Called when the connection to the scanner service fails 25
3.2.20 Called when the connection to the scanner service is lost 25
3.2.21 Preview size change callback 25

3.2.22 Initialization success callback 25
3.2.23 Initialization failure callback 26
➢ 3.3 Card Reader Operation module 26
-- Get card reader operation module - getCardReaderManager -- 26
3.3.1 Power on the card 27
3.3.2 Power off the card 27
3.3.3 Start checking card 28
3.3.4 Stop checking card 31
3.3.5 Check whether the card is in the card reader slot 31
3.3.6 Transmit APDU command to card 32
3.3.7 write HCE data via NFC tag 32
3.3.8 Read NDEF data using HCE (Host Card Emulation) in blocking mode 33
3.3.9 Detect a single card (synchronously) 33
3.3.10 Detect a Felica card (single detection) 35
3.3.11 Detect a contactless card (single detection) 36
3.3.12 Mifare Card Single Card Detection 37
3.3.13 Mifare Card Authentication 37
3.3.14 Mifare Card Block Operations 38
3.3.15 Mifare Card Data Reading 38
3.3.16 Mifare Card Data Writing 39
➢ 3.4 EMV Operation module 39
-- Get EMV operation module - getEmvManager -- 39
3.4.1 Set the terminal information for the EMV kernel 41
3.4.2 Get the terminal information for the EMV kernel 41
3.4.3 Set AID 42
3.4.4 Delete AID 42
3.4.5 Retrieve AID List 42
3.4.6 Add CAPK 42
3.4.7 Delete CAPK 43
3.4.8 Retrieve the CAPK list 43
3.4.9 Set Exception File 43
3.4.10 Delete Exception File 43
3.4.11 Retrieve the configured Exception File 43
3.4.12 Set RevocationIPK 44
3.4.13 Delete RevocationIPK 44
3.4.14 Retrieve the configured CAPK Revocation parameters 44
3.4.15 Set Dynamic Reader Limits (DRL) configuration parameters 44
3.4.16 Delete Dynamic Reader Limits (DRL) configuration parameters 45
3.4.17 Retrieve Dynamic Reader Limits (DRL) configuration parameters 45
3.4.18 Set RuPay terminal parameters 45
3.4.19 Delete RuPay terminal parameters 45
3.4.20 Retrieve RuPay terminal parameters 46
3.4.21 Set Apple VAS transaction parameters 46
3.4.22 Retrieve Apple VAS transaction parameters 46
3.4.23 Set Apple Merchant transaction parameters 46

3.4.24 Delete Apple Merchant transaction parameters 47
3.4.25 Retrieve Apple Merchant transaction parameters 47
3.4.26 Retrieve Kernel version information 47
3.4.27 Start EMV transaction 47
3.4.28 Stop EMV transaction 48
3.4.29 Set EMV Tag/Object parameters 48
3.4.30 Retrieve EMV Tag/Object parameters 48
3.4.31 Set the result of multiple application selection 48
3.4.32 Set card information confirmation result 49
3.4.33 Set Pin input result 49
3.4.34 Set the online result 49
3.4.35 Detect card 50
3.4.36 Multiple application selection callback 50
3.4.37 Card information confirmation callback 50
3.4.38 Kernel card scheme type callback 51
3.4.39 Second tap card callback 52
3.4.40 Request PIN input callback 52
3.4.41 Request online process callback 52
3.4.42 Transaction result callback 52
3.4.43 Set AID List 53
3.4.44 Set CAPK List 53
➢ 3.5 General Operation function 53
-- Get general operation module - getGeneralManager -- 53
3.5.1 Set buzzer 54
3.5.2 Set system navigation bar 54
3.5.3 Set system status bar 55
3.5.4 Set system time 55
3.5.5 Get time from system 55
3.5.6 Get time zone 55
3.5.7 Set time zone 56
3.5.8 Reboot device 56
3.5.9 Shut down device 56
3.5.10 Set system property value 56
3.5.11 Get system property value 57
3.5.12 Configure the screen rotation function 57
3.5.13 Toggle LED light on/off 57
3.5.14 Show or hide the LED view 58
3.5.15 Check the dependency library version 58
3.5.16 Show or hide the notification quick settings panel 58
3.5.17 Set default visibility of the NFC logo 58
3.5.18 Control the device indicator light 59
3.5.19 Enable or disable the low battery alert for printing 59
3.5.20 Get the status of the low battery alert for printing 59
3.5.21 Enable or disable the paper-out alert for printing 60
3.5.22 Get the status of the paper-out alert for printing 60

3.5.23 Lock and unlock the installation function 60
➢ 3.6 PINPAD Operation module 61
-- Get pinpad module - getPinpadManager -- 61
3.6.1 Start Input PIN by provided view 61
3.6.2 Start Input PIN by default view 62
3.6.3 Cancel PIN entry 62
3.6.4 Check if blind keyboard mode is enabled 62
3.6.5 Toggle between blind and normal keyboard modes 62
3.6.6 Check whether the PINPAD is virtual or physical 63
3.6.7 Key press event 63
3.6.8 Error callback 64
3.6.9 Callback when PIN input is confirmed 64
3.6.10 Screen rotation callback when using PINPAD 64
➢ 3.7 Printing Operation module 64
-- Get printing operation module - getPrinterManager -- 64
3.7.1 Open the printer module 66
3.7.2 Get the current status of the printer 66
3.7.3 Feed paper before starting printing 66
3.7.4 Set print grayscale 67
3.7.5 Set print font 67
3.7.6 Set print line space 67
3.7.7 Add the text to the print cache 68
3.7.8 Add a line of text with different styles to the print cache 68
3.7.9 Add a blank line to the print cache 68
3.7.10 Add the barcode to the print cache 69
3.7.11 Add the bitmap to the print cache 69
3.7.12 Trigger the printing operation to print out the content in the print cache ... 69
3.7.13 Trigger the printing operation to print out the content in the print cache ... 70
3.7.14 Turn off the printer 70
3.7.15 Configure whether to clear print cache after paper-out warning 70
3.7.16 Set printer gray level by percentage 70
3.7.17 Get current printer gray level percentage 71
3.7.18 Set global font size for printing 71
3.7.19 Get current global font size 71
3.7.20 Set line spacing multiplier for printing 71
3.7.21 Get current line spacing multiplier 72
3.7.22 Callback when the printing finishes successfully 72
3.7.23 Callback when an error occurs during the printing process 72
3.7.24 Set the percentage threshold for not printing when the battery is low 72
3.7.25 Get the percentage threshold for not printing when the battery is low 73
3.7.26 Clear all pending print jobs and close the printer 73
➢ 3.8 Security module 74
-- Get security module - getSecurityManager -- 74
3.8.1 Write keys 77
3.8.2 Write DUKPT keys 78

3.8.3 Generate RSA keys 79
3.8.4 Write RSA keys 79
3.8.5 Read RSA keys 80
3.8.6 Get random numbers 80
3.8.7 Get KCV for the corresponding key type 80
3.8.8 Encrypt or decrypt data by DUKPT 81
3.8.9 RSA encryption or decryption 82
3.8.10 3DES data encryption and decryption 82
3.8.11 Calculate MAC 82
3.8.12 Calculate MAC by DUKPT 83
3.8.13 Get the KSN for the key at the specified index 84
3.8.14 Increase KSN 84
3.8.15 Calculate MAC using the DUKPT_TDES algorithm 85
3.8.16 Erase all keys 85
3.8.17 Write DUKPT_DES key 86
3.8.18 Write MK/SK key 86
3.8.19 Perform RSA encryption or decryption 87
3.8.20 Perform RSA encryption 87
3.8.21 Write TR31 key using extension method 88
3.8.22 Write MK/SK key with support for encryption algorithm and KCV mode 89
3.8.23 Write DUKPT_AES key (supports TLK and KCV mode) 90
3.8.24 Write DUKPT_DES key (supports KCV mode) 90
3.8.25 Read RSA key information 91
3.8.26 Encrypt or decrypt data using DUKPT_AES (supports ksnMode and usedKsn) 91
3.8.27 Compute MAC using DUKPT_AES key 92
3.8.28 Encrypt or decrypt data using SM4 algorithm 93
3.8.29 Delete a key of the specified type 93
3.8.30 Encrypt or decrypt data using DUKPT_AES (supports ksnMode, usedKsn, and padding modes) 94
3.8.31 Encrypt or decrypt data using DUKPT_DES, with support for padding modes 95
3.8.32 Encrypt or decrypt data using DUKPT_DES 96
➢ 3.9 ECR module 97
-- Get ECR module - getEcrManager -- 97
3.9.1 Create a connection to a device 98
3.9.2 Close connection(s) 98
3.9.3 Enumerate and return available ports 99
3.9.4 Hide all displayed pages 99
3.9.5 Register a connection listener 99
3.9.6 Unregister the connection listener 99
3.9.7 Get local BT/HOST information and display as QR code 100
3.9.8 Display QR scanning interface and obtain showHostByQR result 100
3.9.9 Check whether the device is connected to the specified port 100
3.9.10 Display the device' s BT/HOST information via NFC HCE tap 100
3.9.11 Display the UI via NFC HCE tap to receive information from showHostByNFC 101
3.9.12 Retrieve the default printer set in Settings 101

3.9.13 Enable or disable virtual COM port 101
3.9.14 Read data 102
3.9.15 Write data 102
3.9.16 Get connection information 102
3.9.17 Connection state callback 102
3.9.18 Called when a connection is established 103
- **4. Error Code Definition 103**
4.1 CardReaderError 103
4.2 CommonError 106
4.3 GeneralError 106
4.4 PinpadError 106
4.5 PrinterError 107
4.6 ScannerError 108
4.7 SecurityError 109
4.8 EmvError 113
4.9 PosEmvErrorCode 117
4.10 EcrError 118
- **5. Entity Class Definition 118**
5.1 com.kozen.financial.aidl.emv.EmvCapk 118
5.2 com.kozen.financial.aidl.emv.EmvAid 118
5.3 com.kozen.financial.aidl.emv.EmvRevocationIPK 121
5.4 com.kozen.financial.aidl.emv.EmvExceptionFile 122
5.5 ConstantCardReader 122
5.6 ConstantCardReader.CardType 122
5.7 ConstantEmv.POIEmvCoreManager 123
5.8 ConstantEmv.POIEmvCoreManager.AppleTerminalConstraints 124
5.9 ConstantEmv.POIEmvCoreManager.EmvCardInfoConstraints 125
5.10 ConstantEmv.POIEmvCoreManager.EmvDrlConstraints 125
5.11 ConstantEmv.POIEmvCoreManager.EmvOnlineConstraints 125
5.12 ConstantEmv.POIEmvCoreManager.EmvPinConstraints 126
5.13 ConstantEmv.POIEmvCoreManager.EmvResultConstraints 127
5.14 ConstantEmv.POIEmvCoreManager.EmvServiceConstraints 127
5.15 ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints 128
5.16 ConstantEmv.POIEmvCoreManager.EmvTransDataConstraints 130
5.17 ConstantPrinter 133
5.18 ConstantPrinter.Align 133
5.19 ConstantPrinter.BarcodeFormat 133
5.20 ConstantPrinter.PrintFailurePolicy 133
5.21 ConstantPrinter.FontSize 133
5.22 ConstantSecurity 133
5.23 ConstantScanner 136
5.24 ConstantScanner.BarcodeFormat 136
5.25 ConstantScanner.ScannerCameraType 137
5.26 ConstantScanner.BarcodeSupport 137
5.27 com.kozen.financial.pinpad.PinViewEnum 137

5.28 ConstantPrinter.GrayPercent 137
5.29 ConstantPrinter.GlobalFontSize 138
5.30 ConstantPrinter.LineSpaceMultiplier 138
5.31 ConstantEcr.ConnectType 138
5.32 ConstantEcr.ConnectState 138
5.33 ConstantGeneral.IndicatorType 138
5.34 ConstantCardReader.DetectMifareMode 138
5.35 ConstantCardReader.MifareBlockOperateType 139
5.36 ConstantCardReader.MifareCardAuthType 139
5.37 ConstantCardReader.DetectCardType 139
5.38 ConstantCardReader.DetectContactlessMode 139
5.39 ConstantCardReader.DetectFelicaRequestCode 139
5.40 ConstantPinpad.PinpadType 140
5.41 ConstantPrinter.PaperWidth 140
5.42 ConstantSecurity.RSA_BUNDLE_KEY 140
5.43 com.kozen.financial.aidl.cardreader.CustomByteArray 140
5.44 com.kozen.financial.aidl.printer.BarcodePrintLine 140
5.45 com.kozen.financial.aidl.printer.BitmapPrintLine 140
5.46 com.kozen.financial.aidl.printer.TextPrintLine 140
5.47 com.kozen.financial.aidl.printer.BitmapPrintLine 141
- **6. Access permission 141**

- 1. Revision History

Version	Release	Modify Record	Adapted SDK version	Author
2.1	2026/10/13	✓ Add support for setting the low-battery percentage threshold. ✓ Add support for Mifare card detection, authentication, block operations, block reading, and block writing. ✓ Add support for locking and unlocking the installation function. ✓ Add support for DUKPT_AES and DUKPT_DES encryption and decryption, including padding modes. ✓ Add support for setting all the buttons of the Navigation Bar. ✓ Add a card-check listener. ✓ Add a card-detection listener. ✓ Add EMV error codes, PIN pad error codes, Security error codes, and PosEMV error codes. ✓ Add methods for setting the AID and CAPK lists. ✓ Add a method for forcibly closing the printer. ✓ Add an overload of calcDukptDes. ✓ Add additional CardReaderError and ScannerError definitions. ✓ Add additional constants. ✓ Add and correct the Entity Class Definition. ✓ Correct inaccurate API return-value comments and parameter descriptions. ✓ Correct the ConstantGeneral.IndicatorType definition. ✓ Remove duplicate methods.	FinancialService1.7.x	Yiven Yao.Zhang
2.0	2026/07/17	✓ Add support for receiving the `POWER_ON` event from the physical keypad. ✓ Add a switch for the system low-battery	FinancialService1.6.x	Johnny

		reminder. ✓ Add an API to retrieve the system low-battery reminder status. ✓ Add a switch for printer voice notifications when paper is unavailable. ✓ Add an API to retrieve the printer voice notification switch status when paper is unavailable. ✓ Rename the input parameter for scanner fixed-focus settings. ✓ Add support for deleting a specified key.		
1.9	2026/05/12	✓ Added online dependency support without updating the service or AAR file.	FinancialService1.5.x	Johnny
1.8	2026/04/28	✓ Add single-frame image parsing for scanning ✓ Detect keyboard type; devices with physical keyboards can no longer use the on-screen keyboard by default ✓ Add physical keyboard support to startInputPin ✓ Add physical backlit silk-screen NFC logo ✓ Add device indicator light control interface ✓ Significantly improve scanning performance ✓ Add zoom ratio control interface during scanning ✓ Add support for ITF barcode format in scanning ✓ Add virtual port enable/disable functionality in ECR module	FinancialService1.5.x	Johnny
1.7	2026/02/03	✓ Add support to retrieve the AID and CAPK lists. ✓ Add a Mydebit card OPT-IN mode switch during transactions. ✓ Add standalone card checking. ✓ Add RSA encryption and decryption interfaces. ✓ Use BCD format to encrypt track data and PAN data. ✓ Add TR31 key writing. ✓ Add kcvMode when writing MK/SK keys.	FinancialService1.4.x	Johnny

		✓ Add parameters tlkIndex and kcvMode when writing DUKPT_AES keys. ✓ Add kcvMode when writing DUKPT_DES keys. ✓ Add a Bundle parameter when reading RSA keys. ✓ Add ksnMode to the DUKPT_AES encryption/decryption interface. ✓ Add ksnMode to the DUKPT_AES MAC calculation interface. ✓ Add SM4 encryption and decryption interfaces. ✓ Set print density based on the input percentage value. ✓ Get the printer density percentage. ✓ Set the global font size. ✓ Get the global font size. ✓ Add printer line spacing information. ✓ Add support to retrieve printer line spacing information. ✓ Modify the style of the print error popup. ✓ Adjust returning the out-of-paper status before printing. ✓ Add switching for blind keyboard mode. ✓ Add support to get whether the current mode is blind keyboard mode. ✓ Add disabling camera auto-focus during scanning. ✓ Add a custom camera scanning UI. ✓ Add camera scan decoding. ✓ Add stopping camera decoding. ✓ Modify the default preview scanning UI style. ✓ Support LED light on/off control. ✓ Support NFC tag reading. ✓ Add NFC HCE data reading. ✓ Return the device connection status of a specified port. ✓ Get the system default printer.		
1.6	2025/11/14	✓ Add printer cache-clearing settings ✓ Add RSA public/private key encryption and decryption support ✓ Add option to show/hide dropdown menus ✓ Add setting for displaying the NFC logo	FinancialService1.3.2+	Johnny

		mode ✓ Add TR31 support ✓ Supplement the input parameter descriptions for EMV CAPK and AID ✓ Bug fix		
1.5	2025/09/24	✓ Fix the problem of getCardExistStatus description	FinancialService1.2.x	Johnny
1.4	2025/09/15	✓ Fix documentation regarding EMVListener errors ✓ Add key-value enumeration description for PinViewEnum ✓ Add ECR pairing ✓ Add LED display ✓ Add interface to obtain library dependency versions ✓ Add PINPAD screen rotation support ✓ Add scanner default interface ✓ Add default PICC logo ✓ Add printer pop-up reminders for high/low temperature and paper shortage ✓ Add voice notifications for high/low temperature and paper shortage in the printer. ✓ Add support for SM4 (Only for the China region) ✓ Add permission control for each module	FinancialService1.2.x	Johnny
1.3	2025/08/22	✓ Add explanations for three DUKPT_DES functions ✓ Include instructions for MK/SK writing ✓ Fix the issue with ConstantSecurity in the previous version of the document.	FinancialService1.1.x	Johnny
1.2	2025/07/24	✓ Add NFC TAG, ECR (only includes serial port & USB-to-serial), Felica card support, front and rear camera scanning ✓ Support for separate updates of the Kozen SDK ✓ Fix the API level of the system environment to 23. ✓ getCardExistStatus adds a return value for card presence.	FinancialService1.1.x	Johnny
1.1	2025/03/31	✓ Supplemental printing & barcode module error codes/constants ✓ Remove duplicate error messages in EMV module	FinancialService1.0.x	Yue.Cui Yao.Zhang Tong.Liu Johnny

		Add Pinpad rotation support		
1.0	2025/03/10	✓ Add error code definition, entity definition, access permission ✓ Add EMV module ✓ Add SDK integration description	FinancialService1.0.x	Yao.zhang Sunan Johnny
0.5	2025/03/03	✓ Add a description of the financial SDK engine module	FinancialService1.0.x	Sunan Johnny
0.4	2025/02/27	✓ Update card reader, password keyboard, and some interfaces for general operations	FinancialService1.0.x	Johnny
0.3	2025/1/24	✓ Add EMV module, printer module, scanner module, and security module ✓ Update the API description for card reader, PINPAD, and general operation ✓ Add card detection and card positioning functions; Add PINPAD parameter configuration	FinancialService1.0.x	Johnny
0.2	2025/1/2	✓ Add object description	FinancialService1.0.x	Johnny
0.1	2024/11/12	✓ Initial version	FinancialService1.0.x	Johnny

- 2. Overview

2.1 Introduction

KozenFinancialService is a hardware firmware-based API SDK provided by KOZEN. Designed specifically for Java and Android developers. This SDK enables developers to quickly access hardware operation interfaces for KOZEN financial terminal, facilitating efficient business logic implementation.

The SDK primarily includes the following modules: Basic Device Information, Card Operations, PIN Pad, EMV, and Security Module.

This document serves as the KozenFinancialService API Reference.

2.2 Android version and IDE version supported by the SDK

System environment	Platform	Compile environment
Android 6.0 and above	ARM 64, ARM 32	Android Studio, IntelliJ

2.3 Feature Introduction

2.3.1 Financial SDK Engine Module

- This module handles SDK initialization and provides access to various module operation classes.
- Operation class object: FinancialEngine

2.3.2 Cardreader Module

- This module handles card reader functionality.
- Operation class object: ICardReaderManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getCardReaderManager()
Kotlin: FinancialEngine.cardReaderManager()

2.3.3 EMV Module

- This module handles EMV functionality.
- Operation class object: IEmvManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getEmvManager()
Kotlin: FinancialEngine.emvManager()

2.3.4 General Module

- This module handles basic device control functionalities.
- Operation class object: IGeneralManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getGeneralManager()
Kotlin: FinancialEngine.generalManager()

2.3.5 Pinpad Module

- This module handles Pinpad functionality.
- Operation class object: IPinpadManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getPinpadManager()
Kotlin: FinancialEngine.pinpadManager()

2.3.6 Printer Module

- This module handles printing functionality.
- Operation class object: IPrinterManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getPrinterManager()
Kotlin: FinancialEngine.printerManager()

2.3.7 Scanner Module

- This module handles scanning functionality.
- Operation class object: IScannerManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getScannerManager()
Kotlin: FinancialEngine.scannerManager()

2.3.8 Security Module

- This module handles encryption/decryption algorithms and key-related functionalities.
- Operation class object: ISecurityManager
- Example to get the module operation class:

JAVA: FinancialEngine.INSTANCE.getSecurityManager()
Kotlin: FinancialEngine.securityManager()

2.3.9 ECR Module

- This module handles Kiosk functionalities.
- Operation class object: IEcrManager
- Example to get the module operation class:

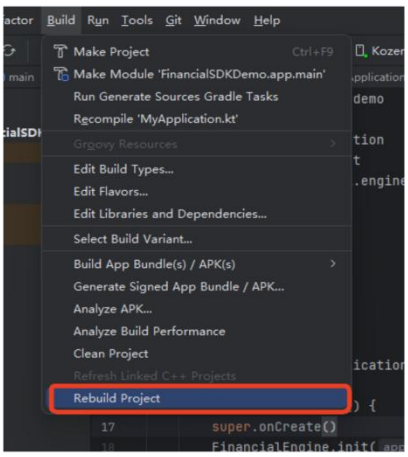
JAVA: FinancialEngine.INSTANCE.getEcrManager()
Kotlin: FinancialEngine.ecrManager()

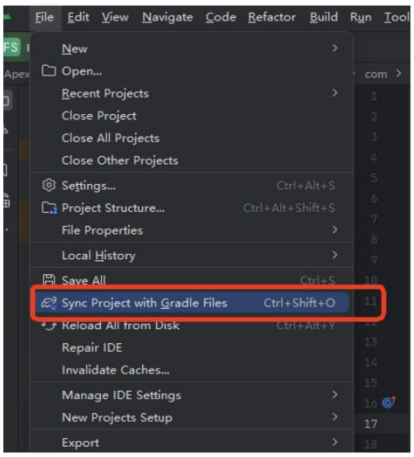
2.4 Importing the Financial SDK

Local Dependency:
Place the `FinancialLib-x.x.x-release.aar` file in the libs directory of your Android Studio project.
Add the following code to the build.gradle file:

```kotlin
dependencies {
    implementation(files("libs/FinancialLib-1.0.0-release.aar"))
}
```

After importing the `.aar` file, sync and rebuild the project.





Online Dependency:
Add the JitPack repository to the settings.gradle file of your Android Studio project:

```gradle
// settings.gradle
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        maven { url 'https://jitpack.io' }
    }
}
```

Add the following dependency `FinancialLib:1.x.x` to the app/build.gradle file:

```gradle
// app/build.gradle
dependencies {
    implementation 'com.github.KOZENTech-Repo.KozenSDK:FinancialLib:1.5.0'
}
```

After adding the dependency, click Sync Project with Gradle Files in Android Studio, then rebuild the project.

2.5 Initializing the Financial SDK

Please initialize the Financial SDK in your Application. Example:

```kotlin
class DemoApplication : Application() {

    override fun onCreate() {
        super.onCreate()

        /**
         * Example: Financial service initialization
         */
        FinancialEngine.init(this, object : InitListener {
            override fun onResult(result: Int, errorMsg: String?) {
                Toast.makeText(
                    this@DemoApplication,
                    if (result == 0) "success" else "failed, $errorMsg",
                    Toast.LENGTH_SHORT
                ).show()
            }
        })
    }
}
```

If initialization is successful, the callback will return `result == 0`.
If initialization fails, the callback will return `result == -1`.

After the Financial SDK is successfully initialized, use `FinancialEngine` to obtain the operation objects for each module. Otherwise, an error code `-10001` (Financial service not connected) will be thrown.

If the financial service is disconnected during use, the `InitListener` will be called again with `result == -1`. Upon receiving this callback, reinitialize the Financial SDK.

If an interface throws the error code `-10001` (Financial service not connected) during use, reinitialize the Financial SDK.

3. API Interface Introduction

void init(android.content.Context application, InitListener callback)	Financial SDK initialization
ICardReaderManager getCardReaderManager()	Card Reader Operation module
IEmvManager getEmvManager()	EMV Operation module
IGeneralManager getGeneralManager()	Device Basic Operation function
IPinpadManager getPinpadManager()	PINPAD Operation module
IPrinterManager getPrinterManager()	Printing Operation module
ISecurityManager getSecurityManager()	Security module
IEcrManager getEcrManager()	ECR Operation module
IScannerManager getScannerManager()	Scanner Operation module

3.1 Financial SDK initialization

3.1.1 Initialize the FinancialService instance

Prototype	void init(android.content.Context application, InitListener callback)
Function	Initialize the FinancialService instance
Parameters	Parameters: application - context callback - initialization callback
Return value	
Notes	

-- FinancialService instance callback - InitListener --

void onResult(int result, String errorMsg)	Initialization results

3.1.2 Initialization results

Prototype	void onResult(int result, String errorMsg)
Function	Initialization results
Parameters	Parameters: result - initialization result 0: success Others: failure - more details to see CommonError errorMsg - error message
Return value	
Notes	

➢ 3.2 Scanner Operation module

-- Get scanner module - getScannerManager --

void close()	Close the scanner module
boolean isBarcodeEnabled(ConstantScanner.BarcodeFormat type)	Check whether a specific barcode format is enabled
void open(ConstantScanner.ScannerCameraType cameraType, IConnectionStatusListener callback)	Open the scanner module with a specific camera type
void open(IConnectionStatusListener callback)	Open the scanner module
int registerResultCallback(IScannerResultCallback callback)	Register a callback to receive scan results
int setBarcodeEnable(boolean enable)	Enable/disable all supported barcode formats
int setBarcodeEnable(List<ConstantScanner.BarcodeFormat> types, boolean enable)	Enable/disable specified barcode formats
int startScan()	Start scanning
int stopScan()	Stop scanning
int switchLight()	Toggle the fill light on or off
void setAFModeEnable(boolean open, int fixDistance)	Enable or disable auto-focus mode, and set fixed focus distance when disabled
int startScan(android.graphics.SurfaceTexture surface, ICamScanInitStatusListener listener)	Start scanning with camera preview on the given surface
int startDecoding()	Trigger the decoding action for scanning
int stopDecoding()	Stop the decoding action without closing preview or camera
int decodeWithBitmap(android.graphics.Bitmap bitmap)	Image decoding
int setZoom(float zoomScale)	Set the scaling factor

3.2.1 Open the scanner module

Prototype	void open(IConnectionStatusListener callback)
Function	Open the scanner module
Parameters	Parameters: callback - Connection status callback
Return Value	
Notes	Initializes the scanner module. Other scanner interfaces should only be used after a successful initialization.

3.2.2 Open the scanner module with a specific camera

Prototype	void open(ConstantScanner.ScannerCameraType cameraType, IConnectionStatusListener callback)
Function	Open the scanner module with a specific camera
Parameters	Parameters: cameraType - Camera type for scanning: 1. CAMERA_REAR, 2. CAMERA_FRONT, 3. SCANNER callback - Connection status callback
Return Value	
Notes	Initializes the scanner with the selected camera. Other interfaces should only be used after successful initialization.

3.2.3 Close the scanner module

Prototype	void close()
Function	Close the scanner module
Parameters	
Return Value	
Notes	Unbinds from the scanner service and releases related resources.

3.2.4 Register a scanner result callback

Prototype	int registerResultCallback(IScannerResultCallback callback)
Function	Register a scanner result callback
Parameters	Parameters: callback - Callback for scan results including type and data
Return Value	Return: 0 - Success; Others - Failure. See ScannerError and CommonError for details
Notes	Must be called after open() is successfully executed.

3.2.5 Start scanning

Prototype	int startScan()
Function	Start scanning
Parameters	
Return Value	Return:0 - Success; Others - Failure. See ScannerError and CommonError for details
Notes	Triggers a scan. Must be called after open() is successfully executed.

3.2.6 Stop scanning

Prototype	int stopScan()
Function	Stop scanning
Parameters	
Return Value	Return:0 - Success; Others - Failure. See ScannerError and CommonError for details
Notes	Stops the current scan. Must be called after open() is successfully executed.

3.2.7 Enable or disable specific barcode types

Prototype	int setBarcodeEnable(List<ConstantScanner.BarcodeFormat> types, boolean enable)
Function	Enable or disable specific barcode types
Parameters	Parameters: types - List of barcode formats to enable/disable enable - true to enable, false to disable
Return Value	Return: 0 - Success; Others - Failure. See ScannerError and CommonError for details
Notes	Must be called after open() is successfully executed.

3.2.8 Enable or disable all barcode types

Prototype	int setBarcodeEnable(boolean enable)
Function	Enable or disable all barcode types
Parameters	Parameters: Enable - true to enable all types; false to disable all types
Return Value	Return: 0 - Success; Others - Failure. See ScannerError and CommonError for details
Notes	Enable or disable all supported code types. This method must be called after a successful open().

3.2.9 Check if a barcode type is enabled

Prototype	boolean isBarcodeEnabled(ConstantScanner.BarcodeFormat type)
Function	Check if a barcode type is enabled
Parameters	Parameters: type - The barcode format to check
Return Value	Return: true to enabled; false to disabled or SDK error
Notes	Must be called after open() is successfully executed.

3.2.10 Toggle the flashlight

Prototype	int switchLight()
Function	Toggle the flashlight
Parameters	
Return Value	Return: 0 - Success; Others - Failure. See ScannerError and CommonError for details
Notes	Requires hardware support; Not supported by SCANNER Must be called after open() is successfully executed.

3.2.11 Enable or disable auto-focus (AF) mode and set fixed focus distance

Prototype	void setAFModeEnable(boolean open, int fixDistance)
Function	Enable or disable auto-focus (AF) mode and set fixed focus distance
Parameters	Parameters: open - true: Enable AF mode; false: Disable AF and use fixed focusfixDistance - Fixed focus distance. Must be ⩾ 0 0 means use the minimum supported focus distance.
Return Value	
Notes	Applicable only if the hardware supports AF mode. When disabled, default focus is ~15cm (hardware-dependent).

3.2.12 Start camera scan and preview rendering to a SurfaceTexture

Prototype	int startScan(android.graphics.SurfaceTexture surface, ICamScanInitStatusListener listener)
Function	Start camera scan and preview rendering to a SurfaceTexture
Parameters	Parameters:

	surface - Target SurfaceTexture for preview listener - Initialization status callback for camera and decoder
Return Value	Return: 0 - Success Others - Failure (see ScannerError / CommonError)
Notes	Must be called after a successful camera open operation.

3.2.13 Trigger the decode action

Prototype	int startDecoding()
Function	Trigger the decode action
Parameters	
Return Value	Return: 0 - Success Others - Failure (see ScannerError / CommonError)
Notes	Only effective after successful startScan(...) initialization.

3.2.14 Stop the decode action

Prototype	int stopDecoding()
Function	Stop the decode action (camera and preview remain active)
Parameters	
Return Value	Return: 0 - Success Others - Failure (see ScannerError / CommonError)
Notes	Only effective after successful startScan(...) initialization.

3.2.15 Decode an image

Prototype	int decodeWithBitmap(android.graphics.Bitmap bitmap)
Function	Decode an image
Parameters	Parameter: bitmap - Image to be decoded. Recommended resolution: 960px × 540px to 1920px × 1080px. Image size should not exceed 20 MB.
Return Value	Return: 0 - Operation succeeded Others - Operation failed (see ScannerError and CommonError)
Notes	

3.2.16 Set zoom scale

Prototype	int setZoom(float zoomScale)

Function	Set zoom scale
Parameters	Parameter: zoomScale - Zoom factor (must be a float ≥ 1.0)
Return Value	Return: 0 - Operation succeeded Others - Operation failed (see ScannerError and CommonError)
Notes	

-- Scanner result callback - IScannerResultCallback--

void onResult(String sym, String barcode)	Retrieve the scan result and the barcode format

3.2.17 Receive scan result and symbology type

Prototype	void onResult(String sym, String barcode)
Function	Receive scan result and symbology type
Parameters	Parameters: sym - Barcode symbology type barcode - Scanned barcode result
Return Value	
Notes	This method is triggered upon a successful scan, returning the barcode data and type.

-- Connection Status Listener- IConnectionStatusListener--

void onConnected()	Called when the connection to the scanning service is established
void onDisconnected()	Called when the connection to the scanning service is lost
void onError(int error, String msg)	Called when the connection to the scanning service fails

3.2.18 Called when the connection to the scanner service is established

Prototype	void onConnected()
Function	Called when the connection to the scanner service is established
Parameters	
Return Value	
Notes	Indicates that the scanner module has been successfully initialized.

3.2.19 Called when the connection to the scanner service fails

Prototype	void onError(int error, String msg)
Function	Called when the connection to the scanner service fails
Parameters	Parameters: error - Error code. See ScannerError and CommonError for definitions msg - Error description
Return Value	
Notes	Triggered upon failure to establish a connection with the scanner service.

3.2.20 Called when the connection to the scanner service is lost

Prototype	void onDisconnected()
Function	Called when the connection to the scanner service is lost
Parameters	
Return Value	
Notes	Indicates that the scanner service is no longer connected.

-- ScanInit Status Listener- ICamScanInitStatusListener--

void onInitFailed(int errCode)	Initialization failed
void onInitSuccess()	Initialization successful
void updatePreviewSize(int previewWidth, int previewHeight)	Callback triggered when the preview size changes; you can adjust the preview interface size and aspect ratio here.

3.2.21 Preview size change callback

Prototype	void updatePreviewSize(int previewWidth, int previewHeight)
Function	Preview size change callback
Parameters	Parameter: previewWidth - Preview width previewHeight - Preview height
Return Value	
Notes	Adjust the preview UI size and aspect ratio when the preview size changes.

3.2.22 Initialization success callback

Prototype	void onInitSuccess()
Function	Initialization success callback

Parameters	
Return Value	
Notes	

3.2.23 Initialization failure callback

Prototype	void onInitFailed(int errCode)
Function	Initialization failure callback
Parameters	errCode - Failure error code. See ScannerError for details
Return Value	
Notes	

➢ 3.3 Card Reader Operation module

-- Get card reader operation module - getCardReaderManager --

int powerOff(int cardType)	Power off the card
int powerOn(int cardType)	Power on the card
int checkCard(int cardType, int timeout, ICheckCardListener callback)	Start checking card
int getCardExistStatus(int cardType)	Check whether the card is in the card reader slot
int stopCheck()	Stop checking card
int transmitApdu(int cardType, byte[] sendBuff, CustomByteArray rspBuf, CustomByteArray swBuf)	Transmit APDU command to card
int hceWrite(android.nfc.NdefMessage msg, int timeout)	Write NDEF data via HCE
byte[] hceRead(int timeout)	Read NDEF data via HCE
int detectCard(int cardType, IDetectCardListener callback)	Perform a single card detection
int detectContactlessCard(String mode, android.os.Bundle bundle)	Perform a single detection of a contactless card
int detectFelicaCard(byte[] systemCode, byte[] requestCode, byte[] timeSlot, android.os.Bundle bundle)	Perform a single detection of a Felica card
int detectMifareCard(int mifareType, int mode, Bundle bundle)	Mifare Card Single Card Detection

int mifareAuth(int authType, int blockNo, byte[] keyBuf, byte[] serialNum)	Mifare Card Authentication
int mifareBlockOperate(int operateType, int index, int srcBlockNo, int dstBlockNo, byte[] operateData)	Mifare Card Block Operations
int mifareBlockRead(int index, int blockNo, int len, CustomByteArray dataBuf);	Read data from the specified block of the Mifare card
int mifareBlockWrite(int index, int blockNo, byte[] buffer);	Write data to the specified block of the Mifare card

3.3.1 Power on the card

Prototype	int powerOn(int cardType)
Function	Card reader powered on
Parameters	Parameters: cardType - card type 1. Supports single card and multiple card types. If input ConstantCardReader.CardType.ALL, can detect all card types. 2. Multiple card types are detected such as follows: ConstantCardReader.CardType.CONTACT | ConstantCardReader.CardType.MAGNETIC 3. More card type details in ConstantCardReader.CardType
Return value	Returns: 0: Success <0: Failure - see CardReaderError
Notes	Note: 1. Due to protocol conflicts, Felica cards cannot be powered on simultaneously with other contactless cards. 2. When powering on cards, if the parameter ConstantCardReader.CardType.ALL is passed, only other contactless cards will be powered on by default, excluding Felica cards. 3. When powering on other contactless cards, Felica cards will be powered off; likewise, when powering on Felica cards, other contactless cards will be powered off.

3.3.2 Power off the card

Prototype	int powerOff(int cardType)
Function	Power off the card
Parameters	Parameters: cardType - card type 1. Supports single card and multiple card types. If input ConstantCardReader.CardType.ALL, can detect all card types. 2. Multiple card types are detected such as follows: ConstantCardReader.CardType.CONTACT | ConstantCardReader.CardType.MAGNETIC 3. More card type details in ConstantCardReader.CardType

Return value	Returns: 0: Success <0: Failure - see CardReaderError
Notes	

3.3.3 Start checking card

Prototype	int checkCard(int cardType, int timeout, ICheckCardListener callback)
Function	Start checking card
Parameters	Parameters: cardType - card type 1. Supports single card and multiple card types. If input ConstantCardReader.CardType.ALL, can detect all card types. 2. Multiple card types are detected such as follows: ConstantCardReader.CardType.CONTACT | ConstantCardReader.CardType.MAGNETIC 3. If both CardType.ALL and specific card types are passed, the specific card types take precedence. Example: ConstantCardReader.CardType.ALL | ConstantCardReader.CardType.CONTACT will only detect contact cards during card detection. 4. More card type details in ConstantCardReader.CardType timeout - Card detection timeout in milliseconds. A value of 0 uses the default timeout of 10,000 ms. callback - card check callback
Return value	Returns: 0: Success <0: Failure - see CardReaderError
Notes	Note: 1. The card will not be powered off automatically after the card checking is completed. Please note that the card must be powered off manually after the checking is completed; 2. During the card checking process, a card checking error will not terminate the process, but the onError callback will be called; 3. The checking process will continue until one type of card is detected or checking times out. 4. Contactless cards, FeliCa cards, and Mifare cards use the same NFC protocol, so two or more card types using the same NFC protocol cannot be detected simultaneously. For multi-type detection, only one of the above NFC protocols can be selected for detection. 5. During card detection, if the card type is passed as "detect all cards" (ConstantCardReader.CardType.ALL), the default is to detect contactless cards only, not FeliCa or Mifare cards.

-- Card check listener - ICheckCardListener --

void findContactCard(android.os.Bundle info)	Contact card found successfully
void findMagstripeCard(android.os.Bundle info)	Magstripe card found successfully
void findContactlessCard(android.os.Bundle info)	Contactless card found successfully
void findFelicaCard(android.os.Bundle info)	Felica card found successfully
void onError(int code, String message)	Card detection error
void onTimeout()	Card detection timeout
void findMifareCard(android.os.Bundle info)	Mifare Card found successfully

3.3.3.1 Contact card found successfully

Prototype	void findContactCard(android.os.Bundle info)
Function	Contact card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR

3.3.3.2 Magstripe card found successfully

Prototype	void findMagstripeCard(android.os.Bundle info)
Function	Magstripe card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR ConstantCardReader.TRACK1 - Track 1 data ConstantCardReader.TRACK2 - Track 2 data ConstantCardReader.TRACK3 - Track 3 data

3.3.3.3 Contactless card found successfully

Prototype	void findContactlessCard(android.os.Bundle info)

Function	Contactless card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR

3.3.3.4 Felica card found successfully

Prototype	void findFelicaCard(android.os.Bundle info)
Function	Felica card found successfully
Parameters	Parameters: info - return data
Return Value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.ID_FOR_MANUFACTURER - Card unique identifier ConstantCardReader.PARAMETER_FOR_MANUFACTURER - Card parameters ConstantCardReader.REQUEST_DATA - Command response

3.3.3.5 Card detection error

Prototype	void onError(int code, String message)
Function	Card detection error
Parameters	Parameters: code - error code, more details to see CardReaderError message - error description
Return value	
Notes	

3.3.3.6 Card detection timeout

Prototype	void onTimeout()
Function	Card detection timeout
Parameters	
Return value	
Notes	

3.3.3.7 Mifare card found successfully

Prototype	void findMifareCard(android.os.Bundle info)
Function	Mifare card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR

3.3.4 Stop checking card

Prototype	int stopCheck()
Function	Stop checking card
Parameters	
Return value	Returns: 0: Success <0: Failure - see CardReaderError
Notes	Note: 1. If you want to interrupt checking manually during the card checking process, please call this method to end the card inspection 2. In the normal process, such as card is found or timeout occurs, the process will automatically stop 3. Stop the card checking process will not power off card; 4. After this function, please use the poweroff function to avoid affecting the power consumption of the machine.

3.3.5 Check whether the card is in the card reader slot

Prototype	int getCardExistStatus(int cardType)
Function	Check whether the card is in the card reader slot
Parameters	cardType - card type 1. Only supports single card 2. Supported card types: 2.1 ConstantCardReader.CardType.CONTACT - contact card 2.2 ConstantCardReader.CardType.CONTACTLESS - contactless card 2.3 ConstantCardReader.CardType.FELICA - felica card
Return value	Return: 0: Card present -1: Card not present

	Other: Failure - see CardReaderError for details
Notes	Note: 1. This function does not support composite cards; it only supports a single card type; 2. This function will not automatically power on the card being checked; 3. The function must be called when the card is already powered on; otherwise, it will return an error code indicating the card is powered off.

3.3.6 Transmit APDU command to card

Prototype	int transmitApdu(int cardType, byte[] sendBuff, CustomByteArray rspBuf, CustomByteArray swBuf)
Function	Transmit APDU command to card
Parameters	Parameters: cardType - card type 1. Only supports single card 2. Supported card types: 2.1 ConstantCardReader.CardType.CONTACT - contact card 2.2 ConstantCardReader.CardType.CONTACTLESS - contactless card 2.3 ConstantCardReader.CardType.FELICA - felica card sendBuff - data to be transparently transmitted to the card, maximum 1929B rspBuf - card response data swBuf - card response data - status word buffer
Return value	Return: 0: Success <0: Failure , see CardReaderError
Notes	Note: 1. This function does not support multiple cards, only single card types; 2. This function will power on the checked card automatically , but will not power off card after the detection is completed; 3. After this function, please use the poweroff function to avoid affecting the power consumption of the machine. 4. APDU command transmission is allowed only after successful card check; this method will fail if the card is powered on without card check

3.3.7 write HCE data via NFC tag

Prototype	int hceWrite(android.nfc.NdefMessage msg, int timeout)
Function	Write NDEF data via HCE
Parameters	Parameters: msg - Data of type NdefMessage , max 255 bytes

	timeout - Timeout in seconds (timeout <= 0 means no timeout)
Return Value	Return: 0: Success; <0: Failure. Refer to CardReaderError for details
Notes	This method writes NDEF data using Host Card Emulation.

3.3.8 Read NDEF data using HCE (Host Card Emulation) in blocking mode

Prototype	byte[] hceRead(int timeout)
Function	Read NDEF data using HCE (Host Card Emulation) in blocking mode
Parameters	Parameters: timeout - Read timeout in seconds
Return Value	Return: Byte array containing the read NDEF data
Notes	This is a blocking method; use with caution on UI or main threads to avoid freezing

3.3.9 Detect a single card (synchronously)

Prototype	int detectCard(int cardType, IDetectCardListener callback)
Function	Detect a single card (synchronously)
Parameters	Parameters: cardType - Type of card to detect. Only a single card type is supported. See ConstantCardReader.DetectCardType callback - Callback triggered upon detection
Return Value	Return: 0: Success <0: Failure. See CardReaderError for details
Notes	This method does not power on the card. Only supports single card types, not composite cards. You must power on the card before calling this method.

-- Card detect listener - IDetectCardListener --

void findContactCard(android.os.Bundle info)	Contact card found successfully
void findMagstripeCard(android.os.Bundle info)	Magstripe card found successfully
void findContactlessCard(android.os.Bundle info)	Contactless card found successfully
void findFelicaCard(android.os.Bundle info)	Felica card found successfully
void onError(int code, String message)	Card detection error
void findMifareCard(android.os.Bundle info)	Mifare Card found successfully

3.3.9.1 Contact card found successfully

Prototype	void findContactCard(android.os.Bundle info)
Function	Contact card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR

3.3.9.2 Magstripe card found successfully

Prototype	void findMagstripeCard(android.os.Bundle info)
Function	Magstripe card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR ConstantCardReader.TRACK1 - Track 1 data ConstantCardReader.TRACK2 - Track 2 data ConstantCardReader.TRACK3 - Track 3 data

3.3.9.3 Contactless card found successfully

Prototype	void findContactlessCard(android.os.Bundle info)
Function	Contactless card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR

3.3.9.4 Felica card found successfully

Prototype	void findFelicaCard(android.os.Bundle info)
Function	Felica card found successfully
Parameters	Parameters: info - return data
Return Value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.ID_FOR_MANUFACTURER - Card unique identifier ConstantCardReader.PARAMETER_FOR_MANUFACTURER - Card parameters ConstantCardReader.REQUEST_DATA - Command response

3.3.9.5 Card detection error

Prototype	void onError(int code, String message)
Function	Card detection error
Parameters	Parameters: code - error code, more details to see CardReaderError message - error description
Return value	
Notes	

3.3.9.6 Mifare card found successfully

Prototype	void findMifareCard(android.os.Bundle info)
Function	Mifare card found successfully
Parameters	Parameters: info - return data
Return value	
Notes	Parameter constant value The following data will be returned during the card detection process ConstantCardReader.CARD_CHANNEL - logical channel number ConstantCardReader.CARD_SERIAL_NUM - Card serial number ConstantCardReader.CARD_ATTRIBUTE - ATR

3.3.10 Detect a Felica card (single detection)

Prototype	int detectFelicaCard(byte[] systemCode, byte[] requestCode, byte[] timeSlot, android.os.Bundle bundle)

Function	Detect a Felica card (single detection)
Parameters	Parameters: systemCode - System code (default: 0xFFFF) requestCode - Request code (default: 0x00) values can be: ConstantCardReader.DetectFelicaRequestCode.NO_REQUEST ConstantCardReader.DetectFelicaRequestCode.SYSTEM_CODE_REQUEST ConstantCardReader.DetectFelicaRequestCode.COMMUNICATION_PERFORMANCE_REQUEST timeSlot - Maximum number of time slots (default: 0x03) bundle - Returned data on success, includes: ConstantCardReader.ID_FOR_MANUFACTURER ConstantCardReader.PARAMETER_FOR_MANUFACTURER ConstantCardReader.REQUEST_DATA
Return Value	Return: 0: Success <0: Failure. See CardReaderError for details
Notes	Used specifically for detecting Felica cards with custom system/request settings.

3.3.11 Detect a contactless card (single detection)

Prototype	int detectContactlessCard(String mode, android.os.Bundle bundle)
Function	Detect a contactless card (single detection)
Parameters	Parameters: mode - Card type list string (e.g. "1,A,B") Value options include: NULL ConstantCardReader.DetectContactlessMode.CARD_READER_DETECT_MODE_ISO14443 ConstantCardReader.DetectContactlessMode.CARD_READER_DETECT_MODE_EMV ConstantCardReader.DetectContactlessMode.CARD_READER_DETECT_MODE_A ConstantCardReader.DetectContactlessMode.CARD_READER_DETECT_MODE_B bundle - Returned data on success, includes: ConstantCardReader.CARD_CHANNEL ConstantCardReader.CARD_SERIAL_NUM ConstantCardReader.CARD_ATTRIBUTE
Return Value	Return: 0: Success <0: Failure. See CardReaderError for details
Notes	This method detects a non-contact (contactless) card by type list, returns card info in the bundle.

3.3.12 Mifare Card Single Card Detection

Prototype	int detectMifareCard(int mifareType, int mode, Bundle bundle)
Function	Mifare Card Single Card Detection
Parameters	Parameters: mifareType - Mifare card type, default value : MIFARE_CLASSIC Value item in list can be: ConstantCardReader.CardType.MIFARE_ULTRALIGHT ConstantCardReader.CardType.MIFARE_CLASSIC ConstantCardReader.CardType.MIFARE_PLUS ConstantCardReader.CardType.MIFARE_DESFIRE mode - Detect mode, default value : 'M' Value item in list can be: ConstantCardReader.DetectMifareMode.CARDREADER_DETECT_MODE_M ConstantCardReader.DetectMifareMode.CARDREADER_DETECT_MODE_N ConstantCardReader.DetectMifareMode.CARDREADER_DETECT_MODE_X bundle - Parameter value after successful card detection. Parameter Constant Value The following data is returned during the card detection process. ConstantCardReader.CARD_CHANNEL - Card logical channel number. ConstantCardReader.CARD_SERIAL_NUM - Card serial number. ConstantCardReader.CARD_ATTRIBUTE - ATR
Return Value	Return: 0: Success <0: Failure — See CardReaderError.
Notes	

3.3.13 Mifare Card Authentication

Prototype	int mifareAuth(int authType, int blockNo, byte[] keyBuf, byte[] serialNum)
Function	Mifare Card Authentication
Parameters	Parameters: authType - Authentication Key Type, Value Range ConstantCardReader.MifareCardAuthType.AUTH_KEY_TYPE_CLASSIC_A

	ConstantCardReader.MifareCardAuthType.AUTH_KEY_TYPE_CLASSIC_B ConstantCardReader.MifareCardAuthType.AUTH_KEY_TYPE_TDES ConstantCardReader.MifareCardAuthType.AUTH_KEY_TYPE_AES blockNo - Block Number. keyBuf - Key Data. serialNum - Card Serial Number.
Return Value	Return: 0: Success <0: Failure — See CardReaderError.
Notes	

3.3.14 Mifare Card Block Operations

Prototype	int mifareBlockOperate(int operateType, int index, int srcBlockNo, int dstBlockNo, byte[] operateData)
Function	Mifare Card Block Operations
Parameters	Parameters: operateType - Value item in list can be: ConstantCardReader.MifareBlockOperateType.OPERATE_BLK_TYPE_ADD ConstantCardReader.MifareBlockOperateType.OPERATE_BLK_TYPE_SUBSTRACT ConstantCardReader.MifareBlockOperateType.OPERATE_BLK_TYPE_TRANSFER index - Block Index srcBlockNo - source Block Number. dstBlockNo - Destination Block Number. operateData - Updated Data.
Return Value	Return: 0: Success <0: Failure — See CardReaderError.
Notes	Before calling this API, you need to power on and authenticate the Mifare card first. Operations on data blocks—value increment, value decrement, and transfer—can be performed only after authentication succeeds.

3.3.15 Mifare Card Data Reading

Prototype	int mifareBlockRead(int index,

	int blockNo, int len, CustomByteArray dataBuf)
Function	Read data from the specified block of the Mifare card
Parameters	Parameters: index - Block Index. blockNo - Block Number. len - Read Block Length. dataBuf - Read data : Len : Response Data Length. Buffer : Response Data.
Return Value	Return: 0: Success <0: Failure — See CardReaderError.
Notes	Before calling this API, you need to power on and authenticate the Mifare card first. Data can be read only after authentication succeeds.

3.3.16 Mifare Card Data Writing

Prototype	int mifareBlockWrite(int index, int blockNo, byte[] buffer)
Function	Write data to the specified block of the Mifare card
Parameters	Parameters: index - Block Index. blockNo - Block Number. buffer - Write Data.
Return Value	Return: 0: Success <0: Failure — See CardReaderError.
Notes	Before calling this API, you need to power on and authenticate the Mifare card first. Data can be written only after authentication succeeds.

➢ 3.4 EMV Operation module

-- Get EMV operation module - getEmvManager --

int deleteAid()	Delete AID
int deleteAppleMerchant()	Delete Apple Merchant transaction parameters

int deleteCapk()	Delete CAPK
int deleteDRL(int type)	Delete Dynamic Reader Limits (DRL) configuration parameters
int deleteExceptionFile()	Delete Exception File
int deleteRevocationIPK()	Delete Revocation IPK
int deleteService()	Delete RuPay payment terminal parameters
List<EmvAid> getAid()	Get AID list
int getAppleMerchant(android.os.Bundle bundle)	Get Apple Merchant transaction parameters
int getAppleTerminal(android.os.Bundle bundle)	Get Apple VAS transaction parameters
List<EmvCapk> getCapk()	Get CAPK list
int getDRL(int type, android.os.Bundle bundle)	Get Dynamic Reader Limits (DRL) configuration parameters
List<EmvExceptionFile> getExceptionFile()	Get configured Exception File
byte[] getKernel(String[] tags)	Get EMV Tag/Object parameters
List<EmvRevocationIPK> getRevocationIPK()	Get configured CAPK Revocation parameters
int getService(android.os.Bundle bundle)	Get RuPay payment terminal parameters
int getTerminal(int type, android.os.Bundle bundle)	Get EMV kernel terminal information
String getVersion(int type)	Get Kernel version information
int setAid(EmvAid emvAid)	Load AID
int setAppleMerchant(android.os.Bundle bundle)	Set Apple Merchant transaction parameters
int setAppleTerminal(android.os.Bundle bundle)	Set Apple VAS transaction parameters
int setCapk(EmvCapk emvCapk)	Add CAPK
int setCardInfoResponse(android.os.Bundle bundle)	Set card information confirmation result
int setDRL(int type, android.os.Bundle bundle)	Set Dynamic Reader Limits (DRL) configuration parameters
int setExceptionFile(EmvExceptionFile exceptionFile)	Set Exception File
int setKernel(byte[] tlv)	Set EMV Tag/Object parameters
int setOnlineResponse(android.os.Bundle bundle)	Set online transaction result
int setPinResponse(android.os.Bundle bundle)	Set PIN input result
int setRevocationIPK(EmvRevocationIPK revocationIPK)	Set Revocation IPK
int setSelectApplicationResponse(int position)	Set multiple application selection result
int setService(android.os.Bundle bundle)	Set RuPay terminal parameters
int setTerminal(int type, android.os.Bundle bundle)	Set EMV kernel terminal information
int startTransaction(android.os.Bundle bundle, IEmvListener callback)	Start EMV transaction
int stopTransaction()	Stop transaction
int setAidList(List<EmvAid> aidList)	Load AID List

int setCapkList(List<EmvCapk> capkList)	Load CAPK List

3.4.1 Set the terminal information for the EMV kernel

Prototype	int setTerminal(int type, android.os.Bundle bundle)
Function	Set the terminal information for the EMV kernel.
Parameters	type: Terminal type, with the following enumerated values: ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_TERMINAL ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_CONFIG ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_VISA ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_UNIONPAY ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_MASTERCARD ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_DISCOVER ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_AMEX ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_MIR ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_RUPAY ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_INTERAC bundle: Terminal parameters. For constant details, refer to ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.2 Get the terminal information for the EMV kernel

Prototype	int getTerminal(int type, android.os.Bundle bundle)
Function	Get the terminal information for the EMV kernel.
Parameters	type: Terminal type, with the following enumerated values: ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_TERMINAL ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_CONFIG ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_VISA ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_UNIONPAY ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_MASTERCARD ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_DISCOVER ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_AMEX ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_MIR ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_RUPAY ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.TYPE_INTERAC bundle: Terminal parameters. For details, refer to the class definition

	of ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.3 Set AID

Prototype	int setAid(EmvAid emvAid)
Function	Set AID
Parameters	emvAid: Parameter entity. For details, refer to EmvAid.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.4 Delete AID

Prototype	int deleteAid()
Function	Delete AID
Parameters	None
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.5 Retrieve AID List

Prototype	List<EmvAid> getAid()
Function	Retrieve the AID list
Parameters	None
Return value	List of EmvAid. For details, refer to EmvAid.
Notes	

3.4.6 Add CAPK

Prototype	int setCapk(EmvCapk emvCapk)
Function	Add CAPK
Parameters	emvCapk: Parameter entity. For details, refer to EmvCapk.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.7 Delete CAPK

Prototype	int deleteCapk()
Function	Delete CAPK
Parameters	None
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.8 Retrieve the CAPK list

Prototype	List<EmvCapk> getCapk()
Function	Retrieve the CAPK list
Parameters	None
Return value	List of EmvCapk. For details, refer to EmvCapk.
Notes	

3.4.9 Set Exception File

Prototype	int setExceptionFile(EmvExceptionFile exceptionFile)
Function	Set Exception File
Parameters	exceptionFile: Parameter entity. For details, refer to EmvExceptionFile.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.10 Delete Exception File

Prototype	int deleteExceptionFile()
Function	Delete Exception File
Parameters	None
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.11 Retrieve the configured Exception File

Prototype	List<EmvExceptionFile> getExceptionFile()
Function	Retrieve the configured Exception File
Parameters	None
Return value	List of EmvExceptionFile. For details, refer to EmvExceptionFile.
Notes	

3.4.12 Set RevocationIPK

Prototype	int setRevocationIPK(EmvRevocationIPK revocationIPK)
Function	Set RevocationIPK
Parameters	revocationIPK: Parameter entity. For details, refer to EmvRevocationIPK.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.13 Delete RevocationIPK

Prototype	int deleteRevocationIPK()
Function	Delete RevocationIPK
Parameters	None
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.14 Retrieve the configured CAPK Revocation parameters

Prototype	List<EmvRevocationIPK> getRevocationIPK()
Function	Retrieve the configured CAPK Revocation parameters
Parameters	None
Return value	List of EmvRevocationIPK. For details, refer to EmvRevocationIPK.
Notes	

3.4.15 Set Dynamic Reader Limits (DRL) configuration parameters

Prototype	int setDRL(int type,android.os.Bundle bundle)
Function	Set Dynamic Reader Limits (DRL) configuration parameters
Parameters	type: DRL type. Supported card scheme types are as follows: ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.TYPE_VISA ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.TYPE_AMEX bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.16 Delete Dynamic Reader Limits (DRL) configuration parameters

Prototype	int deleteDRL(int type)
Function	Delete Dynamic Reader Limits (DRL) configuration parameters
Parameters	type: DRL type. Supported card scheme types are as follows: ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.TYPE_VISA ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.TYPE_AMEX
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.17 Retrieve Dynamic Reader Limits (DRL) configuration parameters

Prototype	int getDRL(int type,android.os.Bundle bundle)
Function	Retrieve Dynamic Reader Limits (DRL) configuration parameters
Parameters	type: DRL type. Supported card scheme types are as follows: ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.TYPE_VISA ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.TYPE_AMEX bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.EmvDrlConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.18 Set RuPay terminal parameters

Prototype	int setService(android.os.Bundle bundle)
Function	Set RuPay terminal parameters
Parameters	bundle: Parameter values. Enumerated values are as follows: Bundle_Key: ConstantEmv.POIEmvCoreManager.EmvServiceConstraints.CONFIG Bundle_value: ByteArray in TLV format. For specific parameter values, refer to ConstantEmv.POIEmvCoreManager.EmvServiceConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.19 Delete RuPay terminal parameters

Prototype	int deleteService()
Function	Delete RuPay payment terminal parameters
Parameters	None

Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.20 Retrieve RuPay terminal parameters

Prototype	int getService(android.os.Bundle bundle)
Function	Retrieve RuPay payment terminal parameters
Parameters	bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.EmvServiceConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.21 Set Apple VAS transaction parameters

Prototype	int setAppleTerminal(android.os.Bundle bundle)
Function	Set Apple VAS transaction parameters
Parameters	bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.AppleTerminalConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.22 Retrieve Apple VAS transaction parameters

Prototype	int getAppleTerminal(android.os.Bundle bundle)
Function	Retrieve Apple VAS transaction parameters
Parameters	bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.AppleTerminalConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.23 Set Apple Merchant transaction parameters

Prototype	int setAppleMerchant(android.os.Bundle bundle)
Function	Set Apple Merchant transaction parameters
Parameters	bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.AppleTerminalConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.24 Delete Apple Merchant transaction parameters

Prototype	int deleteAppleMerchant()
Function	Delete Apple Merchant transaction parameters
Parameters	None
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.25 Retrieve Apple Merchant transaction parameters

Prototype	int getAppleMerchant(android.os.Bundle bundle)
Function	Retrieve Apple Merchant transaction parameters
Parameters	bundle: Parameter values. For details, refer to ConstantEmv.POIEmvCoreManager.AppleTerminalConstraints.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.26 Retrieve Kernel version information

Prototype	String getVersion(int type)
Function	Retrieve Kernel version information
Parameters	type: Kernel type. Enumerated values are as follows: ConstantEmv.POIEmvCoreManager.GET_LIB_VERSION ConstantEmv.POIEmvCoreManager.GET_VERSION_EMV ConstantEmv.POIEmvCoreManager.GET_VERSION_VISA ConstantEmv.POIEmvCoreManager.GET_VERSION_MASTERCARD ConstantEmv.POIEmvCoreManager.GET_VERSION_DISCOVER ConstantEmv.POIEmvCoreManager.GET_VERSION_AMEX ConstantEmv.POIEmvCoreManager.GET_VERSION_MIR ConstantEmv.POIEmvCoreManager.GET_VERSION_RUPAY ConstantEmv.POIEmvCoreManager.GET_VERSION_INTERAC ConstantEmv.POIEmvCoreManager.GET_VERSION_APPLE
Return value	Success:Version information Failure:Refer to EmvError.EMV_GET_KERNEL_VERSION_ERROR
Notes	

3.4.27 Start EMV transaction

Prototype	int startTransaction(android.os.Bundle bundle, IEmvListener callback)
Function	Start EMV transaction
Parameters	bundle: Transaction parameters. For details, refer

	to ConstantEmv.POIEmvCoreManager.EmvTransDataConstraints. callback: EMV process callback.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.28 Stop EMV transaction

Prototype	int stopTransaction()
Function	Stop EMV transaction
Parameters	None
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.29 Set EMV Tag/Object parameters

Prototype	int setKernel(byte[] tlv)
Function	Set EMV Tag/Object parameters
Parameters	tlv: TAG in TLV format. Example: HexUtil.parseHex("9F02060000000000119F0306000000000011").
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	After setting, this method is only used to change TAG values during the EMV transaction process. The set TAG values cannot be retrieved using getKernel.

3.4.30 Retrieve EMV Tag/Object parameters

Prototype	byte[] getKernel(String[] tags)
Function	Retrieve EMV Tag/Object parameters
Parameters	tags: Array of TAGs. Example: new String[]{"4F", "50", "87", "9F12"}.
Return value	TAG values in TLV format.
Notes	

3.4.31 Set the result of multiple application selection

Prototype	int setSelectApplicationResponse(int position)
Function	Set the result of multiple application selection
Parameters	position: The selected position in the multiple application selection callback data.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.32 Set card information confirmation result

Prototype	int setCardInfoResponse(android.os.Bundle bundle)
Function	Set card information confirmation result
Parameters	bundle - parameter value For details, see ConstantEmv.POIEmvCoreManager.EmvCardInfoConstraints
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.33 Set Pin input result

Prototype	int setPinResponse(android.os.Bundle bundle)
Function	Set Pin input result
Parameters	bundle - parameter value For details, see ConstantEmv.POIEmvCoreManager.EmvPinConstraints
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.34 Set the online result

Prototype	int setOnlineResponse(android.os.Bundle bundle)
Function	Set the online result
Parameters	bundle - parameter value For details, see ConstantEmv.POIEmvCoreManager.EmvOnlineConstraints
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

-- EMV Listener- IEmvListener--

void onConfirmCardInfo(int mode, android.os.Bundle info)	Callback for card information confirmation
void onEmvProcess(int type, android.os.Bundle info)	Callback when a card is detected
void onKernelType(int type)	Callback for kernel card scheme type
void onRequestInputPin(android.os.Bundle info)	Callback to request PIN input
void onRequestOnlineProcess(android.os.Bundle info)	Callback to request online processing
void onSecondTapCard()	Callback for second card tap
void onSelectApplication(List<String> appList, boolean isFirstSelect)	Callback for multiple application selection
void onTransactionResult(int resultCode,	Callback for transaction result

android.os.Bundle info)	

3.4.35 Detect card

Prototype	void onEmvProcess(int type, android.os.Bundle info)
Function	Detect card
Parameters	Parameter: type: Card type. Enumerated values are as follows: ConstantEmv.POIEmvCoreManager.DEVICE_CONTACT ConstantEmv.POIEmvCoreManager.DEVICE_CONTACTLESS ConstantEmv.POIEmvCoreManager.DEVICE_MAGSTRIPE ConstantEmv.POIEmvCoreManager.DEVICE_MIFARE_CLASSIC ConstantEmv.POIEmvCoreManager.DEVICE_MIFARE_ULTRALIGHT ConstantEmv.POIEmvCoreManager.DEVICE_MIFARE_PLUS ConstantEmv.POIEmvCoreManager.DEVICE_MIFARE_DESFIRE info: Card information parameters.
Return value	
Notes	If the transaction card type is a magnetic stripe card, the card data will be returned in this Bundle. For specific parameter constants, refer to ConstantEmv.POIEmvCoreManager.EmvCardInfoConstraints.

3.4.36 Multiple application selection callback

Prototype	void onSelectApplication(List<String> appList, boolean isFirstSelect)
Function	Multiple application selection callback
Parameters	Parameter: appList: Application selection list. isFirstSelect: Whether it is the first selection.
Return value	
Notes	

3.4.37 Card information confirmation callback

Prototype	void onConfirmCardInfo(int mode, android.os.Bundle info)
Function	Card information confirmation callback
Parameters	Parameter: mode: Current mode. Enumerated values are as follows: ConstantEmv.POIEmvCoreManager.CMD_TRY_OTHER_APPLICATION ConstantEmv.POIEmvCoreManager.CMD_AMOUNT_CONFIG

	ConstantEmv.POIEmvCoreManager.CMD_ISSUER_REFERRAL ConstantEmv.POIEmvCoreManager.CMD_GPO_FILTER ConstantEmv.POIEmvCoreManager.CMD_READ_RECORD_FILTER ConstantEmv.POIEmvCoreManager.CMD_SELECT_APPLICATION ConstantEmv.POIEmvCoreManager.CMD_READ_RECORD ConstantEmv.POIEmvCoreManager.CMD_GAC1 ConstantEmv.POIEmvCoreManager.CMD_GAC2 ConstantEmv.POIEmvCoreManager.CMD_SELECT_KERNEL ConstantEmv.POIEmvCoreManager.CMD_SELECT_AFTER ConstantEmv.POIEmvCoreManager.CMD_GPO_BEFORE ConstantEmv.POIEmvCoreManager.CMD_CARD_READ_SUCCESS ConstantEmv.POIEmvCoreManager.CMD_APDU_CMD_LOG ConstantEmv.POIEmvCoreManager.CMD_APDU_RESP_LOG ConstantEmv.POIEmvCoreManager.CMD_KERNEL_LOG info: Card information data. For specific parameter constants, refer to ConstantEmv.POIEmvCoreManager.EmvCardInfoConstraints.
Return value	
Notes	

3.4.38 Kernel card scheme type callback

Prototype	void onKernelType(int type)
Function	Kernel card scheme type callback
Parameters	Parameter: type: Kernel card scheme type. Enumerated values are as follows: ConstantEmv.POIEmvCoreManager.EMV_CARD_NOT ConstantEmv.POIEmvCoreManager.EMV_CARD_VISA ConstantEmv.POIEmvCoreManager.EMV_CARD_UNIONPAY ConstantEmv.POIEmvCoreManager.EMV_CARD_MASTERCARD ConstantEmv.POIEmvCoreManager.EMV_CARD_DISCOVER ConstantEmv.POIEmvCoreManager.EMV_CARD_AMEX ConstantEmv.POIEmvCoreManager.EMV_CARD_JCB ConstantEmv.POIEmvCoreManager.EMV_CARD_MIR ConstantEmv.POIEmvCoreManager.EMV_CARD_RUPAY ConstantEmv.POIEmvCoreManager.EMV_CARD_PURE ConstantEmv.POIEmvCoreManager.EMV_CARD_INTERAC ConstantEmv.POIEmvCoreManager.EMV_CARD_EFTPOS ConstantEmv.POIEmvCoreManager.EMV_CARD_MONYX
Return value	
Notes	1. This parameter is only returned for contactless card types. 2. For contact transactions and magnetic stripe transactions, this parameter will not be returned.

3.4.39 Second tap card callback

Prototype	void onSecondTapCard()
Function	Second tap card callback
Parameters	
Return value	
Notes	

3.4.40 Request PIN input callback

Prototype	void onRequestInputPin(android.os.Bundle info)
Function	Request PIN input callback
Parameters	Parameter: info: PIN parameters. For details, refer to ConstantEmv.POIEmvCoreManager.EmvPinConstraints.
Return value	
Notes	

3.4.41 Request online process callback

Prototype	void onRequestOnlineProcess(android.os.Bundle info)
Function	Request online process callback
Parameters	Parameter: info: Online parameters. For details, refer to ConstantEmv.POIEmvCoreManager.EmvOnlineConstraints.
Return value	
Notes	

3.4.42 Transaction result callback

Prototype	void onTransactionResult(int resultCode, android.os.Bundle info)
Function	Transaction result callback
Parameters	Parameter: resultCode: Transaction result code. 0: Success Others: Failure. Refer to ConstantEmv.PosEmvErrorCode/EmvError. info: Transaction data. For details, refer to ConstantEmv.POIEmvCoreManager.EmvResultConstraints.
Return value	
Notes	

3.4.43 Set AID List

Prototype	int setAidList(List<EmvAid> aidList)
Function	Set AID List
Parameters	aidList: Parameter entity list. For details, refer to EmvAid.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

3.4.44 Set CAPK List

Prototype	int setCapkList(List<EmvCapk> capkList)
Function	Set CAPK List
Parameters	capkList: Parameter entity list. For details, refer to EmvCapk.
Return value	0: Success <0: Failure. Refer to EmvError for error codes.
Notes	

➢ 3.5 General Operation function

-- Get general operation module - getGeneralManager --

int setTime(long time)	Set time
long getTime()	Get time
int setTimeZone(String timeZone)	Set time zone
String getTimeZone()	Get time zone
int setBeep(boolean enable, int times, int freq)	Set Beep
int setNavigationBar(int type, boolean isHide)	Set navigation bar
int setStatusBar(boolean isHide)	Set status bar
int shutdown()	Shut down device
String getSystemProperty(String key)	Get system property value
int setSystemProperty(String key, String value)	Set system property value
int reboot()	Reboot device
int setScreenRotation(boolean enable)	Configure screen rotation functionality
void led(int type, int status)	Switches the LED light between ON and OFF states
void setLedVisible(boolean visible)	Sets the visibility of the LedView

String checkDependencyVersion(int type)	Check the dependency library version
int setNotificationShade(boolean isEnable)	Set the Quick Settings dropdown menu to show/hide
int setNfcLogoVisible(boolean visible)	Default to hiding or showing the NFC logo
int setDeviceIndicator(int type, int brightness, boolean enable)	Device operation indicator light
boolean isLowBatteryAlertEnabled()	Get the status of the low battery alert for printing
boolean isPrintVoicePromptEnabled()	Get the status of the paper-out voice prompt for printing
int setLowBatteryAlertEnabled(boolean enabled)	Enable or disable the low battery alert for printing.
int setPrintVoicePromptEnabled(boolean enabled)	Enable or disable the paper-out voice prompt for printing.
int setAppInstallOrUpdateLocked(boolean lock, int timeout)	Lock and unlock the installation function.

3.5.1 Set buzzer

Prototype	int setBeep(boolean enable, int times, int freq)
Function	Set Beep
Parameters	Parameters: enable - begin or close buzzer. times - buzzer duration. freq - buzzer frequency.
Return value	Return: 0: success Others: failure - see GeneralError
Notes	

3.5.2 Set system navigation bar

Prototype	int setNavigationBar(int type, boolean isHide)
Function	Set system navigation bar
Parameters	Parameters: type - the type of navigation bar to be operated 0: All button; 1: back; 2: home; 3: recent. isHide - the state of the navigation bar to be operated. true: hidden false: displayed.
Return value	Return:

	0: success. Others: failure - see GeneralError.
Notes	

3.5.3 Set system status bar

Prototype	int setStatusBar(boolean isHide)
Function	Set system status bar
Parameters	isHide - the state of the status bar to be operated true: hidden false: displayed
Return value	Return: 0: success Others: failure - see GeneralError
Notes	

3.5.4 Set system time

Prototype	int setTime(long time)
Function	Set system time
Parameters	time - The timestamp in milliseconds since the UTC epoch.
Return value	Return: 0: success Others: failure - see GeneralError
Notes	

3.5.5 Get time from system

Prototype	long getTime()
Function	Get time from system
Parameters	
Return value	Return: The timestamp of the current time (the number of milliseconds since the UTC 1970-01-01 00:00:00 epoch).
Notes	

3.5.6 Get time zone

Prototype	String getTimeZone()
Function	Get time zone
Parameters	
Return value	Returns: Time zone
Notes	

3.5.7 Set time zone

Prototype	int setTimeZone(String timeZone)
Function	Set time zone
Parameters	Parameters: timeZone - time zone id, supports two formats: 1. Region/City 2. GMT Example 1: setTimeZone("Europe/Moscow") Example 2: setTimeZone("GMT+9")
Return value	Return: 0: success Others: failure - see GeneralError
Notes	

3.5.8 Reboot device

Prototype	int reboot()
Function	Reboot device
Parameters	
Return value	Return: 0: success Others: failure - see GeneralError
Notes	

3.5.9 Shut down device

Prototype	int shutdown()
Function	Shut down device
Parameters	
Return value	Return: 0: success Others: failure - see GeneralError
Notes	

3.5.10 Set system property value

Prototype	int setSystemProperty(String key, String value)
Function	Set system property value
Parameters	Parameters: key - the key of the system property(ro.product.model) value - the value of the system property
Return value	Return: 0: success

	Others: failure - see GeneralError
Notes	

3.5.11 Get system property value

Prototype	String getSystemProperty(String key)
Function	Get system property value
Parameters	Parameters: key - the key of the system property (ro.product.model)
Return value	Return: Return the value of the system property by string form, or null if it cannot be read
Notes	

3.5.12 Configure the screen rotation function

Prototype	int setScreenRotation(boolean enable)
Function	Configure the screen rotation function
Parameters	Parameters: enable - true: Enable screen rotation - false: Disable screen rotation
Return value	0: Success Others: failure - see GeneralError
Notes	

3.5.13 Toggle LED light on/off

Prototype	void led(int type, int status)
Function	Toggle LED light on/off
Parameters	Parameters: type - LED light type (range: 1-4): 1 - Blue 2 - Yellow 3 - Green 4 - Red status- Switch state: 1 - Turn on 0 - Turn off
Return Value	
Notes	This method is used to control the on/off state of different colored LED lights.

3.5.14 Show or hide the LED view

Prototype	void setLedVisible(boolean visible)
Function	Show or hide the LED view
Parameters	Parameters: visible true: show the LED view false: hide the LED view
Return Value	
Notes	Controls the visibility of the on-screen LED view component.

3.5.15 Check the dependency library version

Prototype	String checkDependencyVersion(int type)
Function	Check the dependency library version
Parameters	Parameters: type - Dependency type: 0 - EMV kernel 1 - POI 2 - UART 3 - Scanner box
Return Value	Return: Version string of the specified dependency
Notes	This method returns the version number of the selected module by type.

3.5.16 Show or hide the notification quick settings panel

Prototype	int setNotificationShade(boolean isEnable)
Function	Show or hide the notification quick settings panel
Parameters	Parameters: isEnable - true: Show the quick settings dropdown (default) false: Hide the quick settings dropdown
Return Value	Return: 0 - Operation succeeded Others - Operation failed (see CommonError for details)
Notes	Controls the visibility of the Android notification shade's quick settings panel.

3.5.17 Set default visibility of the NFC logo

Prototype	int setNfcLogoVisible(boolean visible)
Function	Set default visibility of the NFC logo
Parameters	Parameters:

	visible - true: Show NFC logo false: Hide NFC logo
Return Value	Return: 0 - Operation succeeded Others - Operation failed (see CommonError for details)
Notes	Sets whether the NFC logo is shown by default. This setting persists after reboot until changed again.

3.5.18 Control the device indicator light

Prototype	int setDeviceIndicator(int type, int brightness, boolean enable)
Function	Control the device indicator light
Parameters	Parameters: type - Indicator type: ConstantGeneral.IndicatorType.PINPAD_PHYSICAL brightness - Brightness level (range: 0-255, 0 turns off the indicator) enable - Indicator switch: true to enable, false to disable
Return Value	Return: 0 - Success Others - Failure (see GeneralError)
Notes	Controls brightness and on/off state of the specified device indicator light.

3.5.19 Enable or disable the low battery alert for printing

Prototype	int setLowBatteryAlertEnabled(boolean enabled)
Function	Enable or disable the low battery alert for printing
Parameters	Parameters: enabled - true: Printing is blocked and a low battery alert dialog is displayed when the battery is low. false: Printing is allowed and no low battery alert dialog is displayed.
Return Value	Return: 0 - Success Others - Failure (see GeneralError)
Notes	The low battery alert is enabled by default.

3.5.20 Get the status of the low battery alert for printing

Prototype	boolean isLowBatteryAlertEnabled()
Function	Get the status of the low battery alert for printing

Parameters	
Return Value	Return: true - Enabled false - Disabled
Notes	

3.5.21 Enable or disable the paper-out alert for printing

Prototype	int setPrintVoicePromptEnabled(boolean enabled)
Function	Enable or disable the paper-out alert for printing
Parameters	Parameters: enabled - true: A paper-out alert dialog is displayed when the printer is out of paper. false: No paper-out alert dialog is displayed.
Return Value	Return: 0 - Success Others - Failure (see GeneralError)
Notes	The paper-out alert is enabled by default.

3.5.22 Get the status of the paper-out alert for printing

Prototype	boolean isPrintVoicePromptEnabled()
Function	Get the status of the paper-out alert for printing
Parameters	
Return Value	Return: true - Enabled false - Disabled
Notes	

3.5.23 Lock and unlock the installation function

Prototype	int setAppInstallOrUpdateLocked(boolean lock, int timeout)
Function	Lock and unlock the installation function
Parameters	Parameters: lock - true: Lock the installation function; false: Unlock the installation function. timeout - The timeout period for locking the installation function. When lock is true, timeout must be greater than 0. When the timeout expires, the installation function is automatically unlocked. When lock is false, timeout is ignored, and the installation function is unlocked immediately.

Return Value	Return: 0: Success <0: Failure — See GeneralError.
Notes	Lock/Unlock the TerminalManager app installation/update interface functions. When locked, execution of the app installation/update interface functions is suspended. When unlocked, execution of the app installation/update interface functions resumes.

➢ 3.6 PINPAD Operation module

-- Get pinpad module - getPinpadManager --

void cancelInputPin()	Cancel PIN entry
void startInputPin(android.os.Bundle params, PinpadInputCallback callback)	Start Input PIN, take over TP according to the default view
void startInputPin(android.os.Bundle params, Map<String,android.view.View> keyViews, PinpadInputCallback callback)	Start Input PIN, take over TP according to the incoming view
boolean isBlindModeEnable()	Check if blind keyboard mode is enabled
int switchBlindMode()	Toggle between blind keyboard mode and normal keyboard mode
int getPinpadType()	Get the keyboard type

3.6.1 Start Input PIN by provided view

Prototype	void startInputPin(android.os.Bundle params, Map<String,android.view.View> keyViews, PinpadInputCallback callback)
Function	Start PIN entry and take over touch processing based on the provided views

Parameters	Parameters:
	params - PIN configuration parameters. See ConstantEmv.POIEmvCoreManager.EmvPinConstraints for details
	
	keyViews -- View map for PIN keys. Example:
	Map<String, View> keyViews = new HashMap<>();
	keyViews.put(PinViewEnum.BUTTON0.getType(), button0);
	keyViews.put(PinViewEnum.BUTTON1.getType(), button1);
	keyViews.put(PinViewEnum.BUTTON2.getType(), button2);
	keyViews.put(PinViewEnum.BUTTON3.getType(), button3);
	keyViews.put(PinViewEnum.BUTTON4.getType(), button4);
	keyViews.put(PinViewEnum.BUTTON5.getType(), button5);
	keyViews.put(PinViewEnum.BUTTON6.getType(), button6);
	keyViews.put(PinViewEnum.BUTTON7.getType(), button7);

	keyViews.put(PinViewEnum.BUTTON8.getType(), button8);
	keyViews.put(PinViewEnum.BUTTON9.getType(), button9);
	keyViews.put(PinViewEnum.BUTTON_ENTER.getType(), buttonEnter);
	keyViews.put(PinViewEnum.BUTTON_CLEAR.getType(), buttonBackspace);
	keyViews.put(PinViewEnum.BUTTON_ESC.getType(), buttonEsc);
	
	callback - the PIN input callback

Return value	
Notes	take over TP according to the provided view

3.6.2 Start Input PIN by default view

Prototype	void startInputPin(android.os.Bundle params, PinpadInputCallback callback)
Function	Start input PIN and take over TP according to the default view
Parameters	Parameters: params - PIN configuration parameters Detail information refer to ConstantEmv.POIEmvCoreManager.EmvPinConstraints callback - the PIN input callback
Return value	
Notes	take over TP according to the default view

3.6.3 Cancel PIN entry

Prototype	void cancelInputPin()
Function	Cancel PIN entry
Parameters	
Return value	
Notes	

3.6.4 Check if blind keyboard mode is enabled

Prototype	boolean isBlindModeEnable()
Function	Check if blind keyboard mode is enabled
Parameters	
Return Value	Return: true - Blind keyboard mode is enabled false - Normal keyboard mode
Notes	This method returns the current state of the keyboard mode.

3.6.5 Toggle between blind and normal keyboard modes

Prototype	int switchBlindMode()

Function	Toggle between blind and normal keyboard modes
Parameters	
Return Value	Return: 0: Success <0: Failure. See PinpadError for details
Notes	This method switches the keyboard mode. Typically used to assist visually impaired users.

3.6.6 Check whether the PINPAD is virtual or physical

Prototype	int getPinpadType()
Function	Check whether the PINPAD is virtual or physical
Parameters	
Return Value	Return: 0: Virtual PINPAD 1: Physical PINPAD Others - Failure (see PinpadError)
Notes	

-- PIN input callback - PinpadInputCallback --

void onInput(int len, int key)	Key press event
void onPinError(int verifyResult, int pinTryCntOut)	Callback on error
void onPinSuccess(int verifyResult, byte[] pinBlock, String ksn)	Invoked when the user confirms PIN entry
void onScreenRotation()	Screen rotation callback

3.6.7 Key press event

Prototype	void onInput(int len, int key)
Function	Key press event
Parameters	Parameters: len - The length of the password that has been entered key - The current key value, uniformly returns *
Return value	
Notes	

3.6.8 Error callback

Prototype	void onPinError(int verifyResult, int pinTryCntOut)
Function	Error callback
Parameters	Parameters: verifyResult - Error code, refer to ConstantEmv.POIEmvCoreManager.EmvPinConstraints pinTryCntOut - Number of PIN retry attempts
Return value	
Notes	

3.6.9 Callback when PIN input is confirmed

Prototype	void onPinSuccess(int verifyResult, byte[] pinBlock, String ksn)
Function	Callback when PIN input is confirmed
Parameters	Parameters: verifyResult - The result of the PIN confirmation, 0 means successful PIN confirmation. pinBlock - The ciphertext of the password calculated by PINPAD. ksn - Returned when DUKPT is used.
Return value	
Notes	

3.6.10 Screen rotation callback when using PINPAD

Prototype	void onScreenRotation()
Function	Screen rotation callback when using PINPAD
Parameters	
Return Value	
Notes	This method is triggered when the screen orientation changes.

➢ 3.7 Printing Operation module

-- Get printing operation module - getPrinterManager --

int addBarcode(BarcodePrintLine barcodeLine)	Add the barcode to the print cache.
int addBitmap(BitmapPrintLine bitmapLine)	Add the bitmap to the print cache.
int addText(TextPrintLine textLine)	Add the text to the print cache.
int addText(List<TextPrintLine> textLines)	Add a line of text with different

	styles to the print cache.
void close()	Turn off the printer.
int feedPaper(int lines)	Perform the paper feeding action before starting printing. The main purpose is to adjust the paper position of the printer to ensure that the printed content can be output accurately.
int getPrinterStatus()	Get the current status of the printer.
int open()	Turn on the printer.
int setFont(String path)	Set print font.
int setGray(int gray)	Set print grayscale.
int setLineSpace(int line)	Set print line space.
int startPrint(ConstantPrinter.PrintFailurePolicy failurePolicy, IPrintResultCallback callback)	Execute print job to print the buffered content
int startPrint(IPrintResultCallback callback)	Trigger the printing operation to print out the content in the print cache.
int wrapLine(int lines)	Add a blank line to the print cache, which serves the function of line breaking.
int setClearPrintCacheOnPaperOut(boolean isClear)	Set whether to clear the print cache after a paper-out warning
int setGrayByPercent(ConstantPrinter.GrayPercent percent)	Set print density
ConstantPrinter.GlobalFontSize getGlobalFontSize()	Get the global font size
ConstantPrinter.GrayPercent getGrayByPercent()	Get the current print density percentage
ConstantPrinter.LineSpaceMultiplier getLineSpaceByMultiplier()	Get line spacing information
int setGlobalFontSize(ConstantPrinter.GlobalFontSize size)	Set the global font size
int setLineSpaceByMultiplier(ConstantPrinter.LineSpaceMultiplier lineSpaceMultiplier)	Set the line spacing
int setLowBatteryPrintThreshold(int percent)	Set the percentage threshold for not printing when the battery is low
int getLowBatteryPrintThreshold()	Get the percentage threshold for not printing when the battery is low
void forcedClose()	Force close the printer

3.7.1 Open the printer module

Prototype	int open()
Function	Open the printer module
Parameters	
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	

3.7.2 Get the current status of the printer

Prototype	int getPrinterStatus()
Function	Get the current status of the printer.
Parameters	
Return value	Return: A negative return value: The operation fails; For the specific meaning of the error code, please refer to the definitions in the PrinterError and CommonError. A positive return value: The operation is successfully executed; For the specific meaning of the status code, please refer to the definitions in the ConstantPrinter.
Notes	

3.7.3 Feed paper before starting printing

Prototype	int feedPaper(int lines)
Function	It is used to feed paper before starting the printing process.
Parameters	Parameters: lines - The number of lines to feed (must be an integer greater than 0).
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	The main purpose is to adjust the printer's paper position, ensuring accurate print output. Must be used after the printer has been opened.

3.7.4 Set print grayscale

Prototype	int setGray(int gray)
Function	Set print grayscale
Parameters	Parameters: gray - Default: 1200. Smaller values make prints lighter; larger values make prints darker. Adjust in steps of 200 for consistent results.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened

3.7.5 Set print font

Prototype	int setFont(String path)
Function	Set print font
Parameters	Parameters: path - The path to the font resource package.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.6 Set print line space

Prototype	int setLineSpace(int line)
Function	Set print line space
Parameters	Parameters: line - An integer greater than 0.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.7 Add the text to the print cache

Prototype	int addText(TextPrintLine textLine)
Function	Add the text to the print cache
Parameters	Parameters: textLine - TextPrintLine type(used to define the text content and style for printing).
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.8 Add a line of text with different styles to the print cache

Prototype	int addText(List<TextPrintLine> textLines)
Function	Add a line of text with different styles to the print cache
Parameters	Parameters: textLines - An array of the TextPrintLine type. Recommend to use the TextPrintLineHelper utility class to construct the TextPrintLine array. This can simplify the development process and make the calls clearer and more straightforward.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.9 Add a blank line to the print cache

Prototype	int wrapLine(int lines)
Function	Add a blank line to the print cache, which serves the function of line breaking
Parameters	Parameters: lines - Number of lines, an integer greater than 0.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.10 Add the barcode to the print cache

Prototype	int addBarcode(BarcodePrintLine barcodeLine)
Function	Add the barcode to the print cache
Parameters	Parameters: barcodeLine - BarcodePrintLine Type. Used to define the print content, type, size, and style of the barcode.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.11 Add the bitmap to the print cache

Prototype	int addBitmap(BitmapPrintLine bitmapLine)
Function	Add the bitmap to the print cache
Parameters	Parameters: bitmapLine - BitmapPrintLine Type. Used to define the picture to be printed and its style.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.12 Trigger the printing operation to print out the content in the print cache

Prototype	int startPrint(ConstantPrinter.PrintFailurePolicy failurePolicy, IPrintResultCallback callback)
Function	Execute print job to print the buffered content
Parameters	Parameters: failurePolicy - Policy for handling failed prints (how to process remaining tasks after a single print failure); callback - Callback for monitoring print status.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened

3.7.13 Trigger the printing operation to print out the content in the print cache

Prototype	int startPrint(IPrintResultCallback callback)
Function	Trigger the printing operation to print out the content in the print cache
Parameters	Parameters: callback - Monitor the status callbacks of the printing process.
Return value	Return: 0: The operation is successfully executed; Others: The operation fails. For the specific meaning of the error code, please refer to the definitions in PrinterError and CommonError
Notes	Must be used after the printer has been opened.

3.7.14 Turn off the printer

Prototype	void close()
Function	Turn off the printer
Parameters	
Return value	
Notes	Turn off the printer and release related resources after the printing operation is completed.

3.7.15 Configure whether to clear print cache after paper-out warning

Prototype	int setClearPrintCacheOnPaperOut(boolean isClear)
Function	Configure whether to clear print cache after paper-out warning
Parameters	Parameters: isClear - • true: Warning pops up only once. After clicking confirm, it will not show again. Print cache is cleared. (Default behavior, persists after reboot) • false: Warning persists until the paper-out condition is resolved. After the warning closes, printing resumes without clearing the cached data.
Return Value	Return: 0 - Operation succeeded Others - Operation failed. Refer to PrinterError and CommonError for error details
Notes	Only applies to paper-out warnings; does not affect high/low temperature alerts.

3.7.16 Set printer gray level by percentage

Prototype	int setGrayByPercent(ConstantPrinter.GrayPercent percent)
Function	Set printer gray level by percentage
Parameters	Parameters:

	percent - Gray level percentage (see ConstantPrinter.GrayPercent)
Return Value	Return: 0 - Success Others - Failure (see PrinterError / CommonError)
Notes	Adjusts the print density using predefined percentage levels

3.7.17 Get current printer gray level percentage

Prototype	ConstantPrinter.GrayPercent getGrayByPercent()
Function	Get current printer gray level percentage
Parameters	
Return Value	Return: Gray level percentage (GrayPercent enum)
Notes	Retrieves the current print density setting

3.7.18 Set global font size for printing

Prototype	int setGlobalFontSize(ConstantPrinter.GlobalFontSize size)
Function	Set global font size for printing
Parameters	Parameters: size - Font size (see ConstantPrinter.GlobalFontSize)
Return Value	Return: 0 - Success Others - Failure (see PrinterError / CommonError)
Notes	Sets the default font size used in all printing operations

3.7.19 Get current global font size

Prototype	ConstantPrinter.GlobalFontSize getGlobalFontSize()
Function	Get current global font size
Parameters	
Return Value	Return:Font size (GlobalFontSize enum)
Notes	Returns the currently applied global font size

3.7.20 Set line spacing multiplier for printing

Prototype	int setLineSpaceByMultiplier(ConstantPrinter.LineSpaceMultiplier lineSpaceMultiplier)
Function	Set line spacing multiplier for printing
Parameters	Parameters: lineSpaceMultiplier - Line spacing setting (see LineSpaceMultiplier enum)
Return Value	Return: 0 - Success

	Others - Failure (see PrinterError / CommonError)
Notes	Adjusts the vertical spacing between printed lines

3.7.21 Get current line spacing multiplier

Prototype	ConstantPrinter.LineSpaceMultiplier getLineSpaceByMultiplier()
Function	Get current line spacing multiplier
Parameters	
Return Value	Return: Line spacing multiplier (LineSpaceMultiplier enum)
Notes	Retrieves the current line spacing configuration

-- Print result callback - IPrintResultCallback --

void onFinish()	Callback when the printing finishes successfully.
void onError(int error, String msg)	Callback when an error occurs during the printing process.

3.7.22 Callback when the printing finishes successfully

Prototype	void onFinish()
Function	Callback when the printing finishes successfully
Parameters	
Return value	
Notes	Developers can handle the post - printing - success logic in this method, such as releasing resources, updating the interface status, and closing the printer using the close method.

3.7.23 Callback when an error occurs during the printing process

Prototype	void onError(int error,String msg)
Function	Callback when an error occurs during the printing process
Parameters	Parameters: error - Error code. Refer to PrinterError. msg - Error message.
Return value	
Notes	When receiving the PRINTER_ERROR_PRINT error type, you can try closing the printer first, then reopening it and retrying.

3.7.24 Set the percentage threshold for not printing when the battery is low

Prototype	int setLowBatteryPrintThreshold(int percent)
Function	Set the percentage threshold for not printing when the battery is low

Parameters	Parameters: percent - The battery level percentage threshold. Printing is disabled and a low-battery warning dialog is displayed when the battery level falls below this value. Input range: 0 - 10. A value less than or equal to 0 means no restriction; a value greater than 10 is treated as 10.
Return value	Return: 0: Success <0: Failure (For details about the error codes, see the definitions in PrinterError and CommonError)
Notes	Note: This setting takes effect only when setLowBatteryAlertEnabled is set to true.

3.7.25 Get the percentage threshold for not printing when the battery is low

Prototype	int getLowBatteryPrintThreshold()
Function	Get the percentage threshold for not printing when the battery is low
Parameters	
Return value	Return: 0: No restriction >0: Success (The returned integer represents the low-battery threshold. Printing is disabled and a low-battery warning dialog is displayed when the battery level falls below this value) <0: Failure (For details about the error codes, see the definitions in PrinterError and CommonError)
Notes	Note: This setting takes effect only when setLowBatteryAlertEnabled is set to true.

3.7.26 Clear all pending print jobs and close the printer

Prototype	void forcedClose()
Function	Clear all pending print jobs and close the printer
Parameters	
Return value	
Notes	

➢ 3.8 Security module

-- Get security module - getSecurityManager --

int calcDes(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)	3DES data encryption and decryption
int calcDukpt(int index, int keyMode, int algType, int mode, byte[] dataIn, byte[] aesIv, byte[] dataOut)	Encrypt or decrypt data using DUKPT
int calcMac(int index, int operationMode, byte[] initVector, byte[] dataIn, byte[] dataOut)	Calculate MAC
int calcMacDukpt(int index, int keyType, int mode, byte[] dataIn, byte[] dataOut)	Calculate MAC using DUKPT
int calcRsa(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)	RSA encryption or decryption
int generateRsaKey(int pubKeyIndex, int priKeyIndex, int size)	Generate RSA keys
int getKCV(int keyIndex, int keyType, byte[] kcvOut)	Get the KCV for the corresponding key type
int getKsnDukpt(int index, byte[] ksnOut)	Get the KSN value by specified index
int getRandom(int length, byte[] keyOut)	Get random numbers compliant with NIST SP800-90A
int increaseKsnDukpt(int index)	Increase KSN
int readRsaKey(int keyIndex, byte[] keyOut, byte[] modulusOut, byte[] exponentOut)	Read RSA keys
int writeKey(int srcKeyType,	Write keys

int srcKeyIndex, int keyType, int index, byte[] keyIn, byte[] kcv)	
int writeKeyDukpt(int index, byte[] keyIn, byte[] ksnIn, byte[] kcv)	Write DUKPT keys
int writeRsaKey(int keyIndex, byte[] modulus, byte[] exponent)	Write RSA keys
int calcMacDukptDes(int tikIndex, int operationMode, int ksnMode, int macAlgorithm, byte[] dataIn, byte[] aesIv, byte[] dataOut, byte[] usedKsn)	Calculate MAC using DUKPT_TDES algorithm
int eraseAllKey()	Erase all keys
int writeKeyDukptDes(int tikIndex, int tlkIndex, byte[] keyIn, byte[] ksnIn, byte[] kcv)	Write DUKPT_DES keys
int writeKeyMKSK(int srcKeyType, int srcKeyIndex, int keyType, int encryptionAlgorithm, int index, byte[] keyIn, byte[] kcv)	Inject MK/SK keys
int calcRsaDecrypt(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)	Perform RSA decryption (supports public/private key decryption)
int calcRsaEncrypt(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)	Perform RSA encryption (supports public/private key encryption)
int writeKeyTR31(String tr31KeyBlock, int srcKeyType, int srcKeyIndex, int writeKeyType, int writeKeyIndex,	Write TR31 key using the extended method

int writeKeyAlgorithm)	
int writeKeyMKSK(int srcKeyType, int srcKeyIndex, int keyType, int encryptionAlgorithm, int index, byte[] keyIn, int kcvMode, byte[] kcv)	Write MK/SK key (supports encryption algorithm type and KCV mode)
int writeKeyDukptAes(int tikIndex, int tlkIndex, byte[] keyIn, byte[] ksnIn, int kcvMode, byte[] kcv)	Write DUKPT_AES key (supports TLK and KCV mode)
int writeKeyDukptDes(int tikIndex, int tlkIndex, byte[] keyIn, byte[] ksnIn, int kcvMode, byte[] kcv)	Write DUKPT_DES key (supports KCV mode)
int readRsaKey(int keyIndex, android.os.Bundle bundle)	Read RSA key
int calcDukptAes(int tikIndex, int keyUsage, int algType, byte[] initVector, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] aesIv, byte[] dataOut, byte[] usedKsn)	Encrypt or decrypt data using DUKPT_AES (supports ksnMode and usedKsn)
int calcMacDukptAes(int tikIndex, int keyUsage, int algType, int macAlgorithm, int ksnMode, byte[] dataIn, byte[] dataOut, byte[] usedKsn)	Calculate MAC using DUKPT_AES
int calcSM4(int keyIndex, int mode, byte[] dataIn,	Encrypt or decrypt data using SM4

byte[] dataOut)	
int eraseKey(int srcKeyType, int srcKeyIndex)	Delete the key of the specified type and index
int calcDukptAes(int tikIndex, int keyUsage, int algType, byte[] initVector, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] aesIv, byte[] dataOut, byte[] usedKsn, int paddingMode)	Encrypt or decrypt data using DUKPT_AES (supports ksnMode, usedKsn, and padding modes).
int calcDukptDes(int index, int dukptKeyType, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] initVector, byte[] dataOut, byte[] usedKsn, int paddingMode)	Encrypt or decrypt data using DUKPT_DES, with support for padding modes.
int calcDukptDes(int index, int dukptKeyType, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] initVector, byte[] dataOut, byte[] usedKsn)	Encrypt or decrypt data using DUKPT_DES, without support for padding modes.

3.8.1 Write keys

Prototype	int writeKey(int srcKeyType, int srcKeyIndex, int keyType, int index, byte[] keyIn, byte[] kcv)
Function	Write keys

Parameters	Parameters: srcKeyType - Source key type Such as ConstantSecurity.PED_TLK, ConstantSecurity.PED_TMK srcKeyIndex - Source key index the key index used to decrypt the target key. TLK: Only 1 group is supported, and the index range is [1,1]. TMK: Supports 64 groups, the index range is [1, 64]. TPK: Supports 64 groups, the index range is [1, 64]. TAK: Supports 64 groups, the index range is [1, 64]. TDK: Supports 64 groups, the index range is [1, 64]. TEK: Supports 64 groups, the index range is [1, 64]. TTK: Supports 64 groups, the index range is [1, 64]. Note: TPK, TAK, TDK, TEK, TTK share the index space. Key indexes can not be duplicated. If duplication occurs, the later written key will overwrite the previously injected key. keyType - Key type, as follows: ConstantSecurity.PED_TLK ConstantSecurity.PED_TMK ConstantSecurity.PED_TPK ConstantSecurity.PED_TAK ConstantSecurity.PED_TDK ConstantSecurity.PED_TEK ConstantSecurity.PED_TTK index - Key index, as follows: TLK: Only 1 group is supported, and the index range is [1,1]. TMK: Supports 64 groups, the index range is [1, 64]. TPK: Supports 64 groups, the index range is [1, 64]. TAK: Supports 64 groups, the index range is [1, 64]. TDK: Supports 64 groups, the index range is [1, 64]. TEK: Supports 64 groups, the index range is [1, 64]. TTK: Supports 64 groups, the index range is [1, 64]. Note: TPK, TAK, TDK, TEK, TTK share the index space. Key indexes should not be duplicated. If duplication occurs, the later written key will overwrite the previously injected key. keyIn - Key information, supported key lengths: [8, 16, 24, 32] kcv - KCV value
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.2 Write DUKPT keys

Prototype	int writeKeyDukpt(int index,

	byte[] keyIn, byte[] ksnIn, byte[] kcv)
Function	Write DUKPT keys
Parameters	Parameters: index - Key index (supported range 1～10) keyIn - Key data ksnIn - Initialization KSN kcv - KCV value
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.3 Generate RSA keys

Prototype	int generateRsaKey(int pubKeyIndex, int priKeyIndex, int size)
Function	Generate RSA keys
Parameters	Parameters: pubKeyIndex - Public key index priKeyIndex - Private key index size - Key size
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	Note: The key index range is 1～4, only support two sets of public and private keys.

3.8.4 Write RSA keys

Prototype	int writeRsaKey(int keyIndex, byte[] modulus, byte[] exponent)
Function	Write RSA keys
Parameters	Parameters: keyIndex - Key index modulus - Modulus. exponent - Exponent.
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	Note: If the length of the exponent is less than the length of the modulus, the

	key type written is a public key. If the length of the exponent is equal to the length of the modulus, the key type written is a private key.

3.8.5 Read RSA keys

Prototype	int readRsaKey(int keyIndex, byte[] keyOut, byte[] modulusOut, byte[] exponentOut)
Function	Read RSA keys
Parameters	Parameters: keyIndex - Key index keyOut - Key information modulusOut - Key modulus exponentOut - Key exponent
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.6 Get random numbers

Prototype	int getRandom(int length, byte[] keyOut)
Function	Get random numbers compliant with NIST SP800-90A
Parameters	Parameters: length - Length of the random number keyOut - Random number
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.7 Get KCV for the corresponding key type

Prototype	int getKCV(int keyIndex, int keyType, byte[] kcvOut)
Function	Get KCV for the corresponding key type
Parameters	Parameters: keyIndex - Key index keyType - Key type, as follows: ConstantSecurity.PED_TLK

	ConstantSecurity.PED_TMK ConstantSecurity.PED_TPK ConstantSecurity.PED_TAK ConstantSecurity.PED_TDK ConstantSecurity.PED_TEK ConstantSecurity.PED_TIK ConstantSecurity.PED_TTK kcvOut - KCV value
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.8 Encrypt or decrypt data by DUKPT

Prototype	int calcDukpt(int index, int keyMode, int algType, int mode, byte[] dataIn, byte[] aesIv, byte[] dataOut)
Function	Encrypt or decrypt data by DUKPT
Parameters	Parameters: index - Key index keyMode - Dukpt mode, can be as follows ConstantSecurity.PED_CALC_DUKPT_MODE_DEC: 0x01 ConstantSecurity.PED_CALC_DUKPT_MODE_ENC: 0x02 algType - Algorithm type, can be as follows ConstantSecurity.KEY_ALG_TYPE_2TDEA ConstantSecurity.KEY_ALG_TYPE_3TDEA ConstantSecurity.KEY_ALG_TYPE_AES_128 ConstantSecurity.KEY_ALG_TYPE_AES_192 ConstantSecurity.KEY_ALG_TYPE_AES_256 mode - Operation Mode, can be as follows ConstantSecurity.DUKPT_MAC_MODE_ECB: ECB Mode. ConstantSecurity.DUKPT_MAC_MODE_CBC: CBC Mode. dataIn - Input data aesIv - Initialization vector, pass null for ECB mode, or pass an 8-byte vector for other encryption modes dataOut - Encrypted data
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information

Notes	

3.8.9 RSA encryption or decryption

Prototype	int calcRsa(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)
Function	RSA encryption or decryption
Parameters	Parameters: keyIndex - Key index mode - RSA padding Mode, can be as follows ConstantSecurity.PED_CALC_RSA_MODE_NO_PADDING ConstantSecurity.PED_CALC_RSA_MODE_PKCS1_PADDING ConstantSecurity.PED_CALC_RSA_MODE_OAEP_PADDING dataIn - Input data dataOut - Encrypted or decrypted data
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.10 3DES data encryption and decryption

Prototype	int calcDes(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)
Function	3DES data encryption and decryption
Parameters	Parameters: keyIndex - Index of TDK. mode - Encryption and Decryption Mode, can be as follows ConstantSecurity.PED_CALC_DES_MODE_ECB_DEC ConstantSecurity.PED_CALC_DES_MODE_ECB_ENC ConstantSecurity.PED_CALC_DES_MODE_CBC_DEC ConstantSecurity.PED_CALC_DES_MODE_CBC_ENC dataIn - Input data dataOut - Encrypted or decrypted data
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.11 Calculate MAC

Prototype	int calcMac(int index,

	int operationMode, byte[] initVector, byte[] dataIn, byte[] dataOut)
Function	Calculate MAC
Parameters	Parameters: index - TAK key index operationMode- MAC algorithm type, can be as follows ConstantSecurity.MAC_ALGORITHM_CBC ConstantSecurity.MAC_ALGORITHM_XOR_ECB_MAC ConstantSecurity.MAC_ALGORITHM_ANSI_X9_19 ConstantSecurity.MAC_ALGORITHM_ANSI_X9_9 ConstantSecurity.MAC_ALGORITHM_CMAC initVector - Initialization vector dataIn - input data dataOut - Calculated data
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.12 Calculate MAC by DUKPT

Prototype	int calcMacDukpt(int index, int keyType, int mode, byte[] dataIn, byte[] dataOut)
Function	Calculate MAC by DUKPT
Parameters	index - dukpt key index keyType - Key Type : This Code is Coded by Two Part: X | Y E.g 0x00 | 0x40. X (Key Usage) can be as follows 0x00 : ConstantSecurity.AUTHENTICATION_GENERATION. 0x01 : ConstantSecurity.AUTHENTICATION_VERIFICATION. 0x02 : ConstantSecurity.AUTHENTICATION_BOTH. Y (Derive Key Algorithm Type) can be as follows 0x00 : ConstantSecurity.KEY_ALG_TYPE_2TDEA. 0x10: ConstantSecurity.KEY_ALG_TYPE_3TDEA. 0x20: ConstantSecurity.KEY_ALG_TYPE_AES_128. 0x30: ConstantSecurity.KEY_ALG_TYPE_AES_192. 0x40: ConstantSecurity.KEY_ALG_TYPE_AES_256. mode - MAC Operation Control Code.

	This Code is Coded by Three Part: X | Y | Z. E.g 0x00 | 0x04 | 0x80. X (Algorithm Type) can be as follows 0x00 : ConstantSecurity.MAC_MODE_CBC. 0x01 : ConstantSecurity.MAC_MODE_XOR_ECB_MAC. 0x02 : ConstantSecurity.MAC_MODE_ANSI_X9_19. Y (KSN Self Increasing Mode) can be as follows 0x00 : ConstantSecurity.NOT_SELF_INCREASING. 0x40: ConstantSecurity.SELF_INCREASING. Z (Dukpt Mode,default is 0x80) can be as follows 0x80 : ConstantSecurity.DUKPT_MODE_AES_MODE. dataIn - data involved in the calculation dataOut - data after calculation
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.13 Get the KSN for the key at the specified index

Prototype	int getKsnDukpt(int index, byte[] ksnOut)
Function	Get the KSN for the key at the specified index
Parameters	Parameters: index - Key index ksnOut - KSN value
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.14 Increase KSN

Prototype	int increaseKsnDukpt(int index)
Function	Increase KSN
Parameters	Parameters: index - Key index
Return value	Returns: 0: The operation is successfully executed; <0: The operation fails. Please refer to SecurityError for more information
Notes	

3.8.15 Calculate MAC using the DUKPT_TDES algorithm

Prototype	int calcMacDukptDes(int tikIndex, int operationMode, int ksnMode, int macAlgorithm, byte[] dataIn, byte[] initVector, byte[] dataOut, byte[] usedKsn)
Function	Calculate MAC using the DUKPT_TDES algorithm
Parameters	Parameters: tikIndex - DUKPT key index operationMode - Encryption mode: • ConstantSecurity.OPERATION_MODE_ECB • ConstantSecurity.OPERATION_MODE_CBC ksnMode - KSN mode: • ConstantSecurity.KSN_AUTO_INCREASING_BY_DUKPT_TDES_MAC_BOTH_KEY • ConstantSecurity.KSN_NOT_AUTO_INCREASING_BY_DUKPT_TDES_MAC_BOTH_KEY • ConstantSecurity.KSN_NOT_AUTO_INCREASING_BY_DUKPT_TDES_MAC_RSP_KEY macAlgorithm - MAC algorithm: • ConstantSecurity.MAC_ALGORITHM_CBC • ConstantSecurity.MAC_ALGORITHM_XOR_ECB_MAC • ConstantSecurity.MAC_ALGORITHM_ANSI_X9_19 (TDES only) • ConstantSecurity.MAC_ALGORITHM_ANSI_X9_9 (TDES only) dataIn - Input data to be used in MAC calculation initVector - Initialization vector (8 bytes for CBC; null for ECB) dataOut - Output MAC result (8 bytes) usedKsn - KSN used in calculation
Return Value	Return: 0: Success; <0: Failure. See SecurityError for details
Notes	This method calculates a MAC value using the DUKPT TDES algorithm with the provided input, mode, and algorithm settings.

3.8.16 Erase all keys

Prototype	int eraseAllKey()
Function	Erase all keys
Parameters	
Return Value	Return: 0: Success; <0: Failure. See SecurityError for details

Notes	This method deletes all stored cryptographic keys.

3.8.17 Write DUKPT_DES key

Prototype	int writeKeyDukptDes(int tikIndex, int tlkIndex, byte[] keyIn, byte[] ksnIn, byte[] kcv)
Function	Write DUKPT_DES key
Parameters	Parameters: tikIndex - TIK key index (valid range: 1 - 10) tlkIndex - TLK key index (set to 0 if TLK is not used) keyIn - Key data ksnIn - Initial KSN kcv - Key Check Value
Return Value	Return: 0: Success; <0: Failure. See SecurityError for details
Notes	This method writes a DUKPT DES key along with its initial KSN and KCV to the specified index.

3.8.18 Write MK/SK key

Prototype	int writeKeyMKSK(int srcKeyType, int srcKeyIndex, int keyType, int encryptionAlgorithm, int index, byte[] keyIn, byte[] kcv)
Function	Write MK/SK key
Parameters	Parameters: srcKeyType - Source key type: • ConstantSecurity.PED_TLK • ConstantSecurity.PED_TMK srcKeyIndex - Source key index: • ConstantSecurity.TLK: only index 1 is supported • ConstantSecurity.TMK: index range [1, 64] keyType - Target key type: • ConstantSecurity.PED_TLK • ConstantSecurity.PED_TMK • ConstantSecurity.PED_TPK • ConstantSecurity.PED_TAK

	• ConstantSecurity.PED_TDK • ConstantSecurity.PED_TEK • ConstantSecurity.PED_TTK encryptionAlgorithm - Encryption algorithm: • ConstantSecurity.ENCRYPTION_ALGORITHM_TDES • ConstantSecurity.ENCRYPTION_ALGORITHM_AES • ConstantSecurity.ENCRYPTION_ALGORITHM_SM4 index - Key index: • TLK: [1,1]; others (TMK, TPK, etc.): [1,64] Note: TPK, TAK, TDK, TEK, TTK share the same index space; avoid duplication to prevent overwriting. keyIn - Key data (supported lengths: 8, 16, 24, 32 bytes) kcv - Key check value
Return Value	Return: 0: Success; <0: Failure. See SecurityError for details
Notes	This method writes Master or Session keys (MK/SK). Not applicable for DUKPT key injection.

3.8.19 Perform RSA encryption or decryption

Prototype	int calcRsaDecrypt(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)
Function	Perform RSA encryption or decryption (supports both public/private key decryption)
Parameters	Parameters: keyIndex - Key index used for encryption/decryption mode - Padding mode: • ConstantSecurity.PED_CALC_RSA_MODE_NO_PADDING • ConstantSecurity.PED_CALC_RSA_MODE_PKCS1_PADDING • ConstantSecurity.PED_CALC_RSA_MODE_OAEP_PADDING dataIn - Input data dataOut - Output data after encryption/decryption
Return Value	Return: >= 0 - Length of valid data in data Out< 0 - Failure. See error code definition
Notes	This method performs RSA operations using the specified key and padding mode. Decryption with both public and private keys is supported.

3.8.20 Perform RSA encryption

Prototype	int calcRsaEncrypt(int keyIndex,

	int mode, byte[] dataIn, byte[] dataOut)
Function	Perform RSA encryption (supports public/private key encryption)
Parameters	Parameters: keyIndex - Key index used for encryption mode - Padding mode: • ConstantSecurity.PED_CALC_RSA_MODE_NO_PADDING • ConstantSecurity.PED_CALC_RSA_MODE_PKCS1_PADDING • ConstantSecurity.PED_CALC_RSA_MODE_OAEP_PADDING dataIn - Original input data dataOut - Encrypted output data
Return Value	Return: >= 0 - Length of valid data in data Out< 0 - Error code (operation failed)
Notes	This method performs RSA encryption using the specified key and padding mode. Both public and private key encryption are supported.

3.8.21 Write TR31 key using extension method

Prototype	int writeKeyTR31(String tr31KeyBlock, int srcKeyType, int srcKeyIndex, int writeKeyType, int writeKeyIndex, int writeKeyAlgorithm)
Function	Write TR31 key using extension method
Parameters	Parameters: tr31KeyBlock - TR31 format key data srcKeyType - Source key type: • ConstantSecurity.PED_TLK • ConstantSecurity.PED_TMK srcKeyIndex - Source key index writeKeyType - Target key type: • ConstantSecurity.PED_TMK • ConstantSecurity.PED_TPK • ConstantSecurity.PED_TAK • ConstantSecurity.PED_TDK • ConstantSecurity.PED_TEK • ConstantSecurity.PED_TTK writeKeyIndex - Target key index writeKeyAlgorithm - Target key algorithm: • ConstantSecurity.ENCRYPTION_ALGORITHM_TDES • ConstantSecurity.ENCRYPTION_ALGORITHM_AES

Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	

3.8.22 Write MK/SK key with support for encryption algorithm and KCV mode

Prototype	int writeKeyMKSK(int srcKeyType, int srcKeyIndex, int keyType, int encryptionAlgorithm, int index, byte[] keyIn, int kcvMode, byte[] kcv)
Function	Write MK/SK key with support for encryption algorithm and KCV mode
Parameters	Parameters: srcKeyType - Source key type: • ConstantSecurity.PED_TLK • ConstantSecurity.PED_TMK srcKeyIndex - Index of the source key used to encrypt the target key • TLK: Index range [1,1] • TMK: Index range [1,64] keyType - Target key type: • ConstantSecurity.PED_TLK, • ConstantSecurity.PED_TMK, • ConstantSecurity.PED_TPK, • ConstantSecurity.PED_TAK, • ConstantSecurity.PED_TDK, • ConstantSecurity.PED_TEK, • ConstantSecurity.PED_TTK encryptionAlgorithm - Algorithm type: • ConstantSecurity.ENCRYPTION_ALGORITHM_TDES • ConstantSecurity.ENCRYPTION_ALGORITHM_AES • ConstantSecurity.ENCRYPTION_ALGORITHM_SM4 index - Key index: • TLK: [1,1] • Others (TMK, TPK, TAK, TDK, TEK, TTK): [1,64] *Note:* TPK, TAK, TDK, TEK, TTK share index space; newer entries overwrite previous ones at the same index. keyIn - Key value (length: 8, 16, 24, or 32 bytes) kcvMode - KCV validation mode: • ConstantSecurity.KCV_MODE_NO_VERIFY • ConstantSecurity.KCV_MODE_CHK_0 • ConstantSecurity.KCV_MODE_CHK_ODD

	• ConstantSecurity.KCV_MODE_CHK_EVEN kcv - KCV value (required if kcvMode != 0, length: 8 bytes)
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Extended version of MK/SK key injection with encryption algorithm and KCV verification support

3.8.23 Write DUKPT_AES key (supports TLK and KCV mode)

Prototype	int writeKeyDukptAes(int tikIndex, int tlkIndex, byte[] keyIn, byte[] ksnIn, int kcvMode, byte[] kcv)
Function	Write DUKPT_AES key (supports TLK and KCV mode)
Parameters	Parameters: tikIndex - TIK key index (range: 1 - 10) tlkIndex - TLK key index (set to 0 if TLK not used) keyIn - Key value (supported lengths: 16 / 24 / 32 bytes) ksnIn - Initial KSN (12 bytes) kcvMode - KCV validation mode: • ConstantSecurity.KCV_MODE_NO_VERIFY • ConstantSecurity.KCV_MODE_CHK_0 • ConstantSecurity.KCV_MODE_CHK_ODD • ConstantSecurity.KCV_MODE_CHK_EVEN kcv - KCV value (required if kcvMode != 0, length: 8 bytes)
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Used to inject AES-based DUKPT keys with optional TLK and KCV validation support

3.8.24 Write DUKPT_DES key (supports KCV mode)

Prototype	int writeKeyDukptDes(int tikIndex, int tlkIndex, byte[] keyIn, byte[] ksnIn, int kcvMode, byte[] kcv)
Function	Write DUKPT_DES key (supports KCV mode)
Parameters	Parameters: tikIndex - TIK key index (range: 1 - 10)

	tlkIndex - TLK key index (set to 0 if TLK not used) keyIn - Key value ksnIn - Initial KSN (10 bytes) kcvMode - KCV validation mode: • ConstantSecurity.KCV_MODE_NO_VERIFY • ConstantSecurity.KCV_MODE_CHK_0 • ConstantSecurity.KCV_MODE_CHK_ODD • ConstantSecurity.KCV_MODE_CHK_EVEN kcv - KCV value (required if kcvMode != 0, length: 8 bytes)
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Used for injecting DUKPT DES keys with optional TLK and KCV validation support.

3.8.25 Read RSA key information

Prototype	int readRsaKey(int keyIndex, android.os.Bundle bundle)
Function	Read RSA key information
Parameters	Parameters: keyIndex - RSA key index bundle - Output container for key data. Keys are defined in ConstantSecurity.RSA_BUNDLE_KEY
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Used to retrieve RSA key data from a specific key index.

3.8.26 Encrypt or decrypt data using DUKPT_AES (supports ksnMode and usedKsn)

Prototype	int calcDukptAes(int tikIndex, int keyUsage, int algType, byte[] initVector, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] aesIv, byte[] dataOut, byte[] usedKsn)
Function	Encrypt or decrypt data using DUKPT_AES (supports ksnMode and usedKsn)
Parameters	Parameters: tikIndex - Key index

	keyUsage - Key usage mode • ConstantSecurity.USE_DATA_ENCRYPT_KEY • ConstantSecurity.USE_DATA_DECRYPT_KEY • ConstantSecurity.USE_BOTH_WAYS_KEY algType - AES algorithm type • ConstantSecurity.KEY_ALG_TYPE_AES_128 • ConstantSecurity.KEY_ALG_TYPE_AES_192 • ConstantSecurity.KEY_ALG_TYPE_AES_256 initVector - Initial vector (8 bytes; pass empty if ECB mode) operationDirection - Direction of operation: encrypt or decrypt • ConstantSecurity.OPERATION_DIRECTION_ENCRYPT • ConstantSecurity.OPERATION_DIRECTION_DECRYPT operationMode - Operation mode • ConstantSecurity.OPERATION_MODE_ECB • ConstantSecurity.OPERATION_MODE_CBC ksnMode - KSN handling mode • ConstantSecurity.SELF_INCREASING • ConstantSecurity.NOT_SELF_INCREASING dataIn - Data to encrypt/decrypt aesIv - 16-byte AES IV dataOut - Output buffer for result usedKsn - Output: used KSN (12 bytes)
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Performs AES encryption or decryption using a DUKPT key and configurable modes

3.8.27 Compute MAC using DUKPT_AES key

Prototype	int calcMacDukptAes(int tikIndex, int keyUsage, int algType, int macAlgorithm, int ksnMode, byte[] dataIn, byte[] dataOut, byte[] usedKsn)
Function	Compute MAC using DUKPT_AES key
Parameters	Parameters: tikIndex - DUKPT key index keyUsage - Key usage mode • ConstantSecurity.AUTHENTICATION_GENERATION • ConstantSecurity.AUTHENTICATION_VERIFICATION • ConstantSecurity.AUTHENTICATION_BOTH

	algType - AES algorithm type • ConstantSecurity.KEY_ALG_TYPE_AES_128 • ConstantSecurity.KEY_ALG_TYPE_AES_192 • ConstantSecurity.KEY_ALG_TYPE_AES_256 macAlgorithm - MAC algorithm • ConstantSecurity.MAC_ALGORITHM_CBC • ConstantSecurity.MAC_ALGORITHM_XOR_ECB_MAC • ConstantSecurity.MAC_ALGORITHM_ANSI_X9_19 ksnMode - KSN handling mode • ConstantSecurity.SELF_INCREASING • ConstantSecurity.NOT_SELF_INCREASING dataIn - Data to compute MAC from dataOut - Output MAC data (16 bytes) usedKsn - Output: used KSN (12 bytes)
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Computes MAC using the specified AES DUKPT key, algorithm, and mode.

3.8.28 Encrypt or decrypt data using SM4 algorithm

Prototype	int calcSM4(int keyIndex, int mode, byte[] dataIn, byte[] dataOut)
Function	Encrypt or decrypt data using SM4 algorithm
Parameters	Parameters: keyIndex - Index of the key used for encryption/decryption mode - Operation mode and direction: • ConstantSecurity.PED_CALC_DES_MODE_ECB_DEC • ConstantSecurity.PED_CALC_DES_MODE_ECB_ENC • ConstantSecurity.PED_CALC_DES_MODE_CBC_DEC • ConstantSecurity.PED_CALC_DES_MODE_CBC_ENC dataIn - Input data dataOut - Output data (same length as input)
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	Supports ECB and CBC modes for encryption and decryption with SM4

3.8.29 Delete a key of the specified type

Prototype	int eraseKey(int srcKeyType, int srcKeyIndex)

Function	Delete a key of the specified type
Parameters	Parameters: srcKeyType - Key type: MK/SK key types: • ConstantSecurity.PED_TLK • ConstantSecurity.PED_TMK • ConstantSecurity.PED_TPK • ConstantSecurity.PED_TAK • ConstantSecurity.PED_TDK • ConstantSecurity.PED_TEK • ConstantSecurity.PED_TTK RSA key types: • ConstantSecurity.PED_RSA DUKPT key type: • ConstantSecurity.PED_TIK srcKeyIndex - Key index
Return Value	Return: 0: Success <0: Failure (see SecurityError)
Notes	

3.8.30 Encrypt or decrypt data using DUKPT_AES (supports ksnMode, usedKsn, and padding modes)

Prototype	int calcDukptAes(int tikIndex, int keyUsage, int algType, byte[] initVector, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] aesIv, byte[] dataOut, byte[] usedKsn, int paddingMode)
Function	Encrypt or decrypt data using DUKPT_AES (supports ksnMode, usedKsn, and padding modes)
Parameters	Parameters: tikIndex - Key index keyUsage - Possible values: ConstantSecurity.USE_DATA_ENCRYPT_KEY ConstantSecurity.USE_DATA_DECRYPT_KEY ConstantSecurity.USE_BOTH_WAYS_KEY

	algType - Key algorithm type Possible values: ConstantSecurity.KEY_ALG_TYPE_AES_128 ConstantSecurity.KEY_ALG_TYPE_AES_192 ConstantSecurity.KEY_ALG_TYPE_AES_256 initVector - Initialization vector (pass an empty value when the encryption mode is ECB; for other encryption modes, pass an 8-byte vector) operationDirection - Operation direction Possible values: ConstantSecurity.OPERATION_DIRECTION_DECRYPT ConstantSecurity.OPERATION_DIRECTION_ENCRYPT operationMode - Operation mode Possible values: ConstantSecurity.OPERATION_MODE_ECB ConstantSecurity.OPERATION_MODE_CBC ksnMode - KSN mode Possible values: ConstantSecurity.SELF_INCREASING ConstantSecurity.NOT_SELF_INCREASING dataIn - Data to be encrypted aesIv - AES initialization vector (16 bytes) dataOut - Encrypted data (must be a multiple of 16 bytes) usedKsn - KSN used for encryption (12 bytes in length) paddingMode - Data padding mode for encryption Possible values: ConstantSecurity.DUKPT_PADDING_MODE_NONE ConstantSecurity.DUKPT_PADDING_MODE_9797_1_M1 ConstantSecurity.DUKPT_PADDING_MODE_9797_1_M2 ConstantSecurity.DUKPT_PADDING_MODE_PKCS7
Return Value	Return: 0: Success <0: Failure — See SecurityError.
Notes	

3.8.31 Encrypt or decrypt data using DUKPT_DES, with support for padding modes

Prototype	int calcDukptDes(int index, int dukptKeyType, int operationDirection, int operationMode, int ksnMode, byte[] dataIn, byte[] initVector, byte[] dataOut, byte[] usedKsn,

	int paddingMode)
Function	Encrypt or decrypt data using DUKPT_DES, with support for padding modes.
Parameters	Parameters: index - DUKPT key index dukptKeyType - DUKPT key type Possible values: ConstantSecurity.DUKPT_KEY_SELECT_MAC_REQUEST_OR_RESPONSE ConstantSecurity.DUKPT_KEY_SELECT_DATA_REQUEST ConstantSecurity.DUKPT_KEY_SELECT_DATA_RESPONSE ConstantSecurity.DUKPT_KEY_SELECT_PIN_ENCRYPTION operationDirection - Operation direction Possible values: ConstantSecurity.OPERATION_DIRECTION_DECRYPT ConstantSecurity.OPERATION_DIRECTION_ENCRYPT operationMode - Operation mode Possible values: ConstantSecurity.OPERATION_MODE_ECB ConstantSecurity.OPERATION_MODE_CBC ksnMode - KSN mode Possible values: ConstantSecurity.SELF_INCREASING ConstantSecurity.NOT_SELF_INCREASING dataIn - Data to be encrypted initVector - Initialization vector (pass an empty value when the encryption mode is ECB; for other encryption modes, pass an 8-byte vector) dataOut - Encrypted data (must be a multiple of 8 bytes) usedKsn - KSN used for encryption (10 bytes in length) paddingMode - Data padding mode for encryption Possible values: ConstantSecurity.DUKPT_PADDING_MODE_NONE ConstantSecurity.DUKPT_PADDING_MODE_9797_1_M1 ConstantSecurity.DUKPT_PADDING_MODE_9797_1_M2 ConstantSecurity.DUKPT_PADDING_MODE_PKCS7
Return Value	Return: 0: Success <0: Failure — See SecurityError.
Notes	

3.8.32 Encrypt or decrypt data using DUKPT_DES

Prototype	int calcDukptDes(int index, int dukptKeyType, int operationDirection, int operationMode,

	int ksnMode, byte[] dataIn, byte[] initVector, byte[] dataOut, byte[] usedKsn)
Function	Encrypt or decrypt data using DUKPT_DES
Parameters	Parameters: index - DUKPT key index dukptKeyType - DUKPT key type Possible values: ConstantSecurity.DUKPT_KEY_SELECT_MAC_REQUEST_OR_RESPONSE ConstantSecurity.DUKPT_KEY_SELECT_DATA_REQUEST ConstantSecurity.DUKPT_KEY_SELECT_DATA_RESPONSE ConstantSecurity.DUKPT_KEY_SELECT_PIN_ENCRYPTION operationDirection - Operation direction Possible values: ConstantSecurity.OPERATION_DIRECTION_DECRYPT ConstantSecurity.OPERATION_DIRECTION_ENCRYPT operationMode - Operation mode Possible values: ConstantSecurity.OPERATION_MODE_ECB ConstantSecurity.OPERATION_MODE_CBC ksnMode - KSN mode Possible values: ConstantSecurity.SELF_INCREASING ConstantSecurity.NOT_SELF_INCREASING dataIn - Data to be encrypted initVector - Initialization vector (pass an empty value when the encryption mode is ECB; for other encryption modes, pass an 8-byte vector) dataOut - Encrypted data (must be a multiple of 8 bytes) usedKsn - KSN used for encryption (10 bytes in length)
Return Value	Return: 0: Success <0: Failure — See SecurityError.
Notes	

➢ 3.9 ECR module

-- Get ECR module - getEcrManager --

int close(String port)	Close the connection on the specified port; pass 'null' to close all ports
ArrayList<String> getLocal()	Enumerate and return available port numbers

EcrConnector openOrConnect(String connection)	Create a connection; Supported types: Serial, USB-to-Serial, Cash Drawer
void hideHost()	Hide all displayed pages
void registerConnectionListener(ConnectionListener listener)	Register a connection listener
void unregisterConnectionListener()	Unregister the connection listener
String showClientByQR(int timeout, ConstantScanner.ScannerCameraType cameraType)	Display the interface using QR scan, and get the result by showHostByQR
String showHostByQR(ConstantEcr.ConnectType type)	Show the host info as a QR code through BT/HOST
EcrConnector getPrinterMaster()	Get the default printer added in the “Settings”
boolean isOpenOrConnect(String connection)	Return the connection status of the specified port
String showClientByNFC(int timeout)	Display the interface via NFC HCE tap to prepare for receiving data from showHostByNFC
String showHostByNFC(ConstantEcr.ConnectType type)	Retrieve the device’s BT/HOST info and display it via NFC HCE tap interface
void enableVirtualCom(boolean enable)	Enable the virtual serial port. After turning it on, the device will operate in slave mode.

3.9.1 Create a connection to a device

Prototype	EcrConnector openOrConnect(String connection)
Function	Create a connection
Parameters	Parameters: connection - Connection parameter information. Supported formats: Serial: "SERIAL:<port>:<baudrate>" USB-to-serial: "SERIAL:<port>:<baudrate>" Cash drawer port: "SERIAL:<port>:<baudrate>" Bluetooth: "BT:<0|1>:<uuid>:<name|mac>" Network: "HOST:<0|1>:<IP>:<port>" USB device: "USB:VID<vid>:PID<pid>"
Return Value	Return: EcrConnector operation object
Notes	Supports serial, USB to serial, cash drawer port, Bluetooth, NET, and USB device connections.

3.9.2 Close connection(s)

Prototype	int close(String port)

Function	Close connection(s)
Parameters	Parameters: port - Port name to close, or 'null' to close all ports
Return Value	Return: 0 - Success; Others - Failure
Notes	Closes the specified port or all connections.

3.9.3 Enumerate and return available ports

Prototype	ArrayList<String> getLocal()
Function	Enumerate and return available ports
Parameters	
Return Value	Return: List of local port names (e.g., "tty0" for /dev/ttySx)
Notes	Used to detect available local communication ports.

3.9.4 Hide all displayed pages

Prototype	void hideHost()
Function	Hide all displayed pages
Parameters	
Return Value	
Notes	Cancels or clears all currently displayed pages from the host screen.

3.9.5 Register a connection listener

Prototype	void registerConnectionListener(ConnectionListener listener)
Function	Register a connection listener
Parameters	Parameters: listener - Listener for connection events
Return Value	
Notes	Registers a listener to receive callbacks when a connection is established.

3.9.6 Unregister the connection listener

Prototype	void unregisterConnectionListener()
Function	Unregister the connection listener
Parameters	
Return Value	
Notes	

3.9.7 Get local BT/HOST information and display as QR code

Prototype	String showHostByQR(ConstantEcr.ConnectType type)
Function	Get local BT/HOST information and display as QR code
Parameters	Parameters: type - Connection type: • ConstantEcr.ConnectType.BT - Bluetooth • ConstantEcr.ConnectType.HOST - Local network
Return Value	Return: Local BT or HOST information string
Notes	This method returns the device's Bluetooth or network info in a format suitable for QR code display.

3.9.8 Display QR scanning interface and obtain showHostByQR result

Prototype	String showClientByQR(int timeout, ConstantScanner.ScannerCameraType cameraType)
Function	Display QR scanning interface and obtain showHostByQR result
Parameters	Parameters: timeout - Timeout for reading (in seconds) cameraType - Camera used for scanning: • CAMERA_REAR - Rear camera • CAMERA_FRONT - Front camera • SCANNER - Dedicated scan engine
Return Value	Return: Result string from showHostByQR
Notes	Opens a QR code scanning UI to retrieve host information broadcast by showHostByQR.

3.9.9 Check whether the device is connected to the specified port

Prototype	boolean isOpenOrConnect(String connection)
Function	Check whether the device is connected to the specified port
Parameters	Parameters: connection - Connection information
Return Value	Return: true: Connected false: Not connected
Notes	

3.9.10 Display the device's BT/HOST information via NFC HCE tap

Prototype	String showHostByNFC(ConstantEcr.ConnectType type)

Function	Display the device’ s BT/HOST information via NFC HCE tap
Parameters	Parameters: type - Connection type: • ConstantEcr.ConnectType.BT: Bluetooth • ConstantEcr.ConnectType.HOST: Local network
Return Value	Return: BT or HOST information of the device
Notes	

3.9.11 Display the UI via NFC HCE tap to receive information from showHostByNFC

Prototype	String showClientByNFC(int timeout)
Function	Display the UI via NFC HCE tap to receive information from showHostByNFC
Parameters	Parameters: timeout - Timeout in seconds
Return Value	Return: Result from showHostByNFC
Notes	

3.9.12 Retrieve the default printer set in Settings

Prototype	EcrConnector getPrinterMaster()
Function	Retrieve the default printer set in Settings
Parameters	
Return Value	Return: Printer object, or null if not set
Notes	

3.9.13 Enable or disable virtual COM port

Prototype	void enableVirtualCom(boolean enable)
Function	Enable or disable virtual COM port
Parameters	enable - true: Enable virtual COM (device enters slave mode, ttyGSx appears on terminal and VCOM appears on PC) false: Disable virtual COM
Return Value	
Notes	When enabled, the device operates in slave mode and exposes a virtual serial interface to the PC.

-- ECR connector object- EcrConnector --

String getConnectInfo()	Return connection information

byte[] read()	Read data
void write(byte[] commands)	Write data

3.9.14 Read data

Prototype	byte[] read()
Function	Read data
Parameters	
Return Value	Return: Byte array containing the read data
Notes	Reads data from the scanner module or connected device.

3.9.15 Write data

Prototype	void write(byte[] commands)
Function	Write data
Parameters	Parameters: commands - Command data to write (max 4096 bytes)
Return Value	
Notes	Sends command data to the connected device or module.

3.9.16 Get connection information

Prototype	String getConnectInfo()
Function	Get connection information
Parameters	
Return Value	Return: Connection information string
Notes	Returns current connection details of the scanner or related module.

-- ECR connection listener - ConnectionListener--

void onConnected(String connectionInfo)	Will be called when a connection is established
void onState(int state, String msg, String connectionInfo)	Will be callback for connection state changes

3.9.17 Connection state callback

Prototype	void onState(int state,

	String msg, String connectionInfo)
Function	Connection state callback
Parameters	Parameters: state - Connection state value. See ConstantEcr.ConnectState for constants msg - Detailed description of the state connectionInfo - Device connection information
Return Value	
Notes	This method is triggered to notify the current connection state and a descriptive message.

3.9.18 Called when a connection is established

Prototype	void onConnected(String connectionInfo)
Function	Called when a connection is established
Parameters	Parameters: connectionInfo - Device connection information
Return Value	
Notes	This method is invoked when the device has successfully connected.

● 4. Error Code Definition

4.1 CardReaderError

Error Code	Error Description	Error Value
CARD_APDU_OTHER_ERROR	APDU Error	-70004
CARD_CHECK_OTHER_ERROR	Card Check Other Error	-70003
CARD_CONTACT_ATR_ERROR	Contact Card ATR Data Error	-72001
CARD_CONTACT_NO_SUPPORT	IC Card Not Supported	-72000
CARD_CONTACT_OTHERS_ERROR	Contact Card Insertion Failed	-72002
CARD_CONTACTLESS_ATS_ERROR	Contactless Card ATS Data Error	-73001
CARD_CONTACTLESS_NO_SUPPORT	Contactless Card Not Supported	-73000
CARD_CONTACTLESS_OTHERS_ERROR	Contactless Card Other Error	-73002
CARD_EXIST_STATUS_OTHER_ERROR	Card Present Status Other Error	-70005
CARD_FELICA_ATS_ERROR	Felica Card ATS Data Error	-74001
CARD_FELICA_NO_SUPPORT	Felica Card Not Supported	-74000
CARD_FELICA_OTHERS_ERROR	Felica Card Other Error	-74002
CARD_MAG_INVALID_ERROR	Magnetic Track or Data Verification Failed	-71003
CARD_MAG_NEED_RETRY_ERROR	Magnetic Card Swipe Failed, Please Retry	-71002
CARD_MAG_NO_DATA_ERROR	No Data in Magnetic Track	-71001

CARD_MAG_NO_SUPPORT	Magnetic Card Not Supported	-71000
CARD_MAG_OTHERS_ERROR	Magnetic Card Swipe Failed	-71004
CARD_READER_CONTACT_ALREADY_CLOSE	IC Card Already Powered Off	-72006
CARD_READER_CONTACT_CHECK_ERROR	Contact Card Check Failed	-72004
CARD_READER_CONTACT_OFF_ERROR	Contact Card Power-Off Failed	-72005
CARD_READER_CONTACT_OPEN_ERROR	Contact card detection open failed	-72003
CARD_READER_CONTACTLESS_ALREADY_CLOSE	Contactless Card Already Powered Off	-73006
CARD_READER_CONTACTLESS_CHECK_ERROR	Contactless Card Check Failed	-73004
CARD_READER_CONTACTLESS_MULTI_CARD	Multiple Contactless Cards Detected	-73007
CARD_READER_CONTACTLESS_OFF_ERROR	Contactless Card Power-Off Failed	-73005
CARD_READER_CONTACTLESS_OPEN_ERROR	Contactless card open failed	-73003
CARD_READER_FELICA_ALREADY_CLOSE	Felica Card Already Powered Off	-74006
CARD_READER_FELICA_CHECK_ERROR	Felica Card Check Failed	-74004
CARD_READER_FELICA_OFF_ERROR	Felica Card Power-Off Failed	-74005
CARD_READER_FELICA_OPEN_ERROR	Felica card open failed	-74003
CARD_READER_MAG_CHECK_ERROR	Magnetic Card Check Failed	-71006
CARD_READER_MAG_OFF_ERROR	Magnetic Card Power-Off Failed	-71007
CARD_READER_MAG_OPEN_ERROR	Magnetic card detection open failed	-71005
CARD_READER_OFF_OTHER_ERROR	Card Power-Off Other Error	-70002
CARD_READER_ON_OTHER_ERROR	Other error during card power-on	-70006
CARD_TYPE_ERROR	Card Type Error	-70001
NFC_TAG_ALREADY_CLOSE	NFC TAG already closed	-75006
NFC_TAG_NDEF_MESSAGE_ERROR	NFC TAG data error	-75004
NFC_TAG_NDEF_MESSAGE_LENGTH_ERROR	NDEFMessage data length exceeds limit (max: 255 bytes)	-75005
NFC_TAG_NO_SUPPORT	NFC TAG not supported	-75000
NFC_TAG_OFF_ERROR	NFC TAG close failed	-75002
NFC_TAG_OPEN_ERROR	NFC TAG open failed	-75001
NFC_TAG_OTHERS_ERROR	Other NFC TAG error	-75007
NFC_TAG_WRITE_MESSAGE_ERROR	NFC TAG write failed	-75003
NFC_TAG_READER_OPEN_ERROR	Failed to open NFC card emulation reader	-75008
NFC_TAG_READER_OFF_ERROR	Failed to close the NFC card emulation reader	-75009
NFC_TAG_READER_OTHERS_ERROR	Other NFC card emulation read error	-75010
NOT_SUPPORT_CHECK_CONTACTLESS_AND_FELICA_SIMULTANEOUS	Simultaneous contactless and Felica card check not supported	-73008
NOT_SUPPORT_OPEN_CONTACTLESS_AND_FELICA_SIMULTANEOUS	Simultaneous contactless and Felica card power-on not supported	-73009
NOT_SUPPORT_CHECK_CONTACTLESS_SIMULTANEOUS	Simultaneous detection of Contactless, FeliCa, and Mifare	-73010

	NFC cards is not supported.	
NOT_SUPPORT_OPEN_CONTACTLESS_SIMULTANEOUS	Simultaneous power-on of Contactless, FeliCa, NFC tag, and Mifare is not supported.	-73011
CARD_MIFARE_ALREADY_CLOSE	Mifare card has been powered off	-77006
CARD_MIFARE_ATS_ERROR	Mifare card ATS data error	-77001
CARD_MIFARE_AUTH_FAIL	Mifare card authentication failed	-77012
CARD_MIFARE_AUTH_TYPE_ERROR	Invalid Mifare card authentication type	-77011
CARD_MIFARE_BLOCK_OPERATE_ERROR	Mifare card block operation failed	-77010
CARD_MIFARE_BLOCK_WRITE_ERROR	Mifare card data writing failed	-77008
CARD_MIFARE_CHECK_ERROR	Mifare card detection failed	-77004
CARD_MIFARE_DETECT_MODE_ERROR	Invalid Mifare card detection mode	-77007
CARD_MIFARE_NO_SUPPORT	Mifare card is not supported	-77000
CARD_MIFARE_OFF_ERROR	Mifare card power-off failed	-77005
CARD_MIFARE_OPEN_ERROR	Failed to open Mifare card	-77003
CARD_MIFARE_OPERATE_BLOCK_TYPE_ERROR	Invalid Mifare card operation type	-77009
CARD_MIFARE_OTHERS_ERROR	Other Mifare card error	-77002
CARD_MIFARE_READ_FAIL	Mifare card read failed	-77013
CARD_MIFARE_READ_PARAM_ERROR	Invalid Mifare card read parameters	-77014
DETECT_MIFARE_CARD_NOT_FIND	Mifare card not detected	-76052
DETECT_MIFARE_CARD_NOT_OPEN_ERROR	Mifare card is not powered on	-76051
MIFARE_CARD_NOT_OPEN_ERROR	Mifare card is not powered on	-77016
NOT_SUPPORT_CHECK_MULTI_MIFARE_TYPE_SIMULTANEOUS	Simultaneous detection of multiple Mifare card types is not supported	-77015
DETECT_CARD_OTHER_ERROR	Other error during single card detection	-76999
DETECT_CARD_TYPE_NOT_SUPPORT	Card type not supported for single detection	-76001
DETECT_CONTACT_CARD_NOT_FIND	Contact card not detected	-76022
DETECT_CONTACT_CARD_NOT_OPEN_ERROR	Contact card not powered on	-76021
DETECT_CONTACTLESS_CARD_NOT_FIND	Contactless card not detected	-76032
DETECT_CONTACTLESS_CARD_NOT_OPEN_ERROR	Contactless card not powered on	-76031
DETECT_FELICA_CARD_NOT_FIND	Felica card not detected	-76042
DETECT_FELICA_CARD_NOT_OPEN_ERROR	Felica card not powered on	-76041
DETECT_MAG_CARD_NOT_FIND	Magnetic stripe card not detected	-76012
DETECT_MAG_CARD_NOT_OPEN_ERROR	Magnetic stripe card not powered on	-76011

4.2 CommonError

Error Code	Error Description	Error Value
FINANCIAL_PARAMETERS_INVALID	Illegal parameters	-10002
FINANCIAL_SERVICE_DISCONNECT	Financial services are not connected, please initialize financial services	-10001
FINANCIAL_VERSION_NOT_MATCH	Financial Services SDK version mismatch	-10000
FEATURE_NOPERMISSION	Permission not granted	-10010
FEATURE_UNSUPPORTED	Feature unsupported	-10011

4.3 GeneralError

Error Code	Error Description	Error Value
GENERAL_ERROR_INIT	General module initialization error	-20000
GENERAL_OTHER_ERROR	Other general error	-20001
GENERAL_PARAMETERS_INVALID	Invalid parameters	-20002
GENERAL_NAVIGATION_BUTTON_TYPE_INVALID	Invalid navigation button type	-20003
GENERAL_SCREEN_ROTATION_ERROR	Screen rotation error	-20004
GENERAL_SET_BEEP_ERROR	Failed to set beep	-20005
GENERAL_INDICATOR_TYPE_ERROR	Indicator type error	-20100
GENERAL_INDICATOR_TYPE_PINPAD_CAPACITIVE_ERROR	Capacitive PINPAD indicator type error	-20101
GENERAL_INSTALL_LOCKED_TIMEOUT_ERROR	Lock/unlock installation function timeout error	-20200

4.4 PinpadError

Error Code	Error Description	Error Value
PINPAD_OTHER_ERROR	Other PINPAD error	-60000
PINPAD_START_ERROR	Failed to start PINPAD	-60001
PINPAD_CANCEL_ERROR	PINPAD operation canceled	-60002
PINPAD_SCREEN_ORIENTATION_CHANGED	Screen orientation changed during PINPAD operation	-60003
PIN_KEY_COORDINATE_CALCULATION_ERROR	PIN key coordinate calculation error	-60004
PIN_SWITCH_BLIND_MODE_ERROR	Failed to switch to blind PIN mode	-60005
SP_ERROR_PHYSICAL_PINPAD_CALCULATE_KEY_POSITION_ERROR	Failed to calculate physical PINPAD key position	-60006
CURRENT_SP_NOT_SUPPORT_GET_PINPAD_TYPE_ERROR	Current service provider does not support getting PINPAD type	-60007
CURRENT_SP_NOT_SUPPORT_PHYSICAL_PINPAD_ERROR	Current service provider does not	-60008

	support physical PINPAD	
PHYSICAL_PINPAD_NOT_SUPPORT_KEY_POSITION_PARAM_ERROR	Physical PINPAD does not support key position parameters	-60009
PINPAD_KEY_VIEW_POSITION_PARAM_ERROR	Invalid key view position parameters	-60010
DEFAULT_PINPAD_START_ERROR	Failed to start default PINPAD	-60011
CUSTOM_PINPAD_START_ERROR	Failed to start custom PINPAD	-60012
BLIND_PINPAD_START_ERROR	Failed to start blind PINPAD	-60013
PINPAD_ERROR_NO_KEY	Key does not exist	-69001
PINPAD_ERROR_PIN_USER	User input error	-69002
PINPAD_ERROR_PASSWD	Incorrect password	-69003
PINPAD_ERROR_CHANGE_PASSWD	Password reset required	-69004
PINPAD_ERROR_TRY_PASSWD_EXCEED	Password attempts exceeded	-69005
PINPAD_ERROR_PIN_TIMEOUT	PIN input timeout	-69006
PINPAD_ERROR_INPUT_ESC	User pressed ESC to cancel	-69007
PINPAD_ERROR_PIN_ICC_OUT	IC card removed / exited	-69008
PINPAD_ERROR_PIN_INPUT_LEN	PIN length error	-69009
PINPAD_ERROR_PIN_KEYPAD	Keypad input error	-69010
PINPAD_ERROR_NO_PIN_INPUT	No PIN entered	-69011
PINPAD_ERROR_INPUT_CANCEL	User cancelled PIN input	-69012
PINPAD_ERROR_EXPLEN	PIN length descriptor error	-69013
PINPAD_ERROR_INPUT_TIMEOUT	PIN input timeout	-69014
PINPAD_ERROR_INPUT_CLEAR	User pressed CLEAR to exit PIN input	-69015
PINPAD_ERROR_PIN_BYPASS_BYFUNKEY	Skip PIN via function key	-69016
PINPAD_ERROR_PINBLOCK_FMT	PIN block format error	-69017
PINPAD_ERROR_PIN_LEN	PIN length error	-69018
PINPAD_ERROR_KBD_PARA	Keypad parameter error	-69019
PINPAD_ERROR_KBD_REJECT	Keypad operation rejected	-69020
PINPAD_ERROR_KBD_CLOSE	Keypad close error	-69021
PINPAD_ERROR_KBD_AUTHCODE	Keypad authentication code error	-69022
PINPAD_ERROR_KBD_ENCRYPT	Keypad encryption error	-69023
PINPAD_ERROR_VERIFY_ERROR	PIN verification failed -- process exception	-69024

4.5 PrinterError

Error Code	Error Description	Error Value
PRINTER_ERROR_INIT	Printer not initialized or initialization failed	-50000
PRINTER_ERROR_NO_PRINTER	No printer device	-50001
PRINTER_ERROR_NOT_OPENED	Printer not opened	-50002
PRINTER_ERROR_PRINT	Print failed	-50003
PRINTER_ERROR_OVERHEAT	Printer overheating	-50004

PRINTER_ERROR_NO_PAPER	Printer out of paper	-50005
PRINTER_ERROR_LOW_POWER	Printing is not possible when the battery level is low	-50006
PRINTER_ERROR_NO_CONTENT	No content to print	-50007
PRINTER_ERROR_OTHER	Other unknown error	-50008
PRINTER_ERROR_QUEUE_OVER_FLOW	Print queue overflow	-50009
PRINTER_ERROR_SCREEN_OFF	Screen off, unable to print	-50010
PRINTER_ERROR_NOT_SUPPORT	Feature or parameter not supported	-50011
PRINTER_ERROR_PRINTING	Configuration not allowed during printing	-50012

4.6 ScannerError

Error Code	Error Description	Error Value
SCANNER_SERVICE_NOT_FIND	Scanner service not found	-30000
SCANNER_SERVICE_UNAVAILABLE	Scanner service unavailable	-30001
SCANNER_SERVICE_CONNECT_FAILED	Failed to connect to scanner service	-30002
SCANNER_SERVICE_DISCONNECT	Scanner service not connected, please call open to initialize	-30003
SCANNER_OTHER_ERROR	Other unknown error	-30004
CAMERA_OCCUPIED	Camera is occupied, please call close to release resources first	-30005
CAMERA_UNKNOWN	Current device does not support this camera type	-30006
GET_CAMERA_INFO_ERROR	Failed to get camera information, please rebind Financial SDK	-30007
INIT_CAMERA_ERROR	Camera initialization failed, try close then open again	-30008
NOT_SUPPORT_LIGHT_CONTROL	Camera does not support light control	-30009
OPEN_LIGHT_FAILED	Failed to turn on fill light, must be called after scan UI is started	-30010
NO_SCAN_TYPE_SET	No decode type set, please set scan type first	-30011
SET_SCAN_TYPE_FAILED	Failed to enable or disable decode type, please check parameters	-30012
PARAMETER_EXCEPTION	Parameter exception	-30013
NOT_SUPPORTED_FEATURE	Current camera type does not support this feature	-30014
PREVIEW_MODE_EXCEPTION	Preview mode exception	-30015
CAMERA_SESSION_ERROR	Camera session error	-30016
CAMERA_CLOSED_ERROR	Camera was unexpectedly closed	-30017

DECODE_BMP_TRANS_FAILED	Image transfer failed	-30018
DECODE_BMP_EXCEPTION	Image decoding failed	-30019
DECODE_BMP_TOO_LARGE	Image too large, Maximum size is 20 MB	-30020
DECODE_BMP_NOT_AVAILABLE	Image unavailable	-30021

4.7 SecurityError

Error Code	Error Description	Error Value
SECURITY_ERROR_INIT	Security module not initialized	-40000
SECURITY_OTHER_ERROR	Other error (e.g., a system interface call exception)	-40001
SECURITY_PARAMETERS_INVALID	Invalid parameters (null or illegal format)	-40002
SECURITY_KEY_TYPE_OUT_OF_RANGE	Key type out of range	-40003
SECURITY_KEY_INDEX_OUT_OF_RANGE	Key index out of range	-40004
SECURITY_DATA_INDEX_OUT_OF_RANGE	Data index out of range	-40005
SECURITY_TLK_INDEX_OUT_OF_RANGE	TLK index out of range	-40006
SECURITY_KEY_IN_EMPTY_ERROR	Key input data is empty	-40007
SECURITY_DATA_IN_EMPTY_ERROR	Input data is empty	-40008
SECURITY_DATA_OUT_NULL_ERROR	Output data is null	-40009
SECURITY_DATA_OUT_LENGTH_ERROR	Output buffer length insufficient	-40010
SECURITY_KCV_MODE_ERROR	KCV mode error	-40011
SECURITY_KCV_ERROR	KCV parameter error	-40012
SECURITY_RANDOM_KEY_OUT_OF_RANGE	Random key output length out of range	-40013
SECURITY_KCV_VALUE_OUT_OF_RANGE	KCV output value out of range	-40014
SECURITY_MKSK_KEY_TYPE_ERROR	MK/SK key type out of range	-40100
SECURITY_MKSK_SRC_KEY_TYPE_ERROR	MK/SK source key type error	-40101
SECURITY_MKSK_KEY_INDEX_ERROR	MK/SK key index error	-40102
SECURITY_MKSK_SRC_KEY_INDEX_ERROR	MK/SK source key index error	-40103
SECURITY_MKSK_KEY_LENGTH_ERROR	MK/SK key length error	-40104
SECURITY_MKSK_ENCRYPTION_ALGORITHM_ERROR	MK/SK encryption algorithm error	-40105
SECURITY_MKSK_CALC_MODE_ERROR	MK/SK calculation mode error	-40106

SECURITY_MKSK_MAC_MODE_ERROR	MK/SK MAC mode error	-40107
SECURITY_MKSK_MAC_OUT_OF_RANGE	MK/SK MAC output length out of range	-40108
SECURITY_MKSK_CALC_OUT_OF_RANGE	MK/SK calculation output length out of range	-40109
SECURITY_SM4_CALC_MODE_ERROR	SM4 calculation mode error	-40110
SECURITY_SM4_CALC_OUT_OF_RANGE	SM4 calculation output length out of range	-40111
SECURITY_DUKPT_TIK_INDEX_ERROR	DUKPT TIK key index error	-40200
SECURITY_DUKPT_SRC_KEY_INDEX_ERROR	DUKPT source key index error	-40201
SECURITY_DUKPT_KEY_USAGE_ERROR	DUKPT key usage parameter error	-40202
SECURITY_DUKPT_KEY_ALG_TYPE_ERROR	DUKPT key algorithm type error	-40203
SECURITY_DUKPT_MAC_ALG_TYPE_ERROR	DUKPT MAC algorithm type error	-40204
SECURITY_DUKPT_INIT_VECTOR_ERROR	DUKPT init vector parameter error	-40205
SECURITY_DUKPT_AES_VECTOR_ERROR	DUKPT AES init vector parameter error	-40206
SECURITY_DUKPT_KEY_TYPE_ERROR	DUKPT key type parameter error	-40207
SECURITY_DUKPT_OPERATION_MODE_ERROR	DUKPT operation mode error	-40208
SECURITY_DUKPT_OPERATION_DIRECTION_ERROR	DUKPT operation direction error	-40209
SECURITY_DUKPT_KSN_MODE_ERROR	DUKPT KSN mode error	-40210
SECURITY_DUKPT_KEY_LENGTH_ERROR	DUKPT key length error	-40211
SECURITY_DUKPT_CALC_OUT_OF_RANGE	DUKPT calculation output length out of range	-40212
SECURITY_DUKPT_MAC_OUT_OF_RANGE	DUKPT MAC output length out of range	-40213
SECURITY_DUKPT_KSN_OUT_OF_RANGE	DUKPT KSN output length out of range	-40214
SECURITY_RSA_KEY_INDEX_OUT_OF_RANGE	RSA key index out of range	-40300
SECURITY_RSA_PADDING_MODE_ERROR	RSA padding mode error	-40301
SECURITY_RSA_RESPONSE_DATA_EMPTY	RSA response data is empty	-40302

SECURITY_RSA_OUT_DATA_LENGTH_OUT_OF_RANGE	RSA output data length out of range	-40303
SECURITY_TR31_SRC_KEY_TYPE_ERROR	TR31 source key type error	-40401
SECURITY_TR31_SRC_KEY_INDEX_ERROR	TR31 source key index error	-40402
SECURITY_TR31_KEY_TYPE_ERROR	TR31 key type error	-40403
SECURITY_TR31_KEY_INDEX_ERROR	TR31 key index error	-40404
SECURITY_TR31_KEY_BLOCK_EMPTY_ERROR	TR31 key block is empty	-40405
SECURITY_TR31_KEY_ALGORITHM_ERROR	TR31 key algorithm type error	-40406
SECURITY_ERASE_KEY_TYPE_ERROR	Key erase failed: key type error	-40501
SECURITY_DUKPT_PADDING_MODE_ERROR	Invalid DUKPT padding mode	-40215
SECURITY_ERROR_NO_KEY	Key does not exist	-49001
SECURITY_ERROR_KEYIDX_ERR	Key index error	-49002
SECURITY_ERROR_DERIVE_ERR	Key derivation error	-49003
SECURITY_ERROR_CHECK_KEY_FAIL	Key verification failed	-49004
SECURITY_ERROR_WAIT_INTERVAL	Insufficient call interval	-49005
SECURITY_ERROR_CHECK_MODE_ERR	KCV check mode error	-49006
SECURITY_ERROR_NO_RIGHT_USE	No permission to use this key	-49007
SECURITY_ERROR_KEY_TYPE_ERR	Key type error	-49008
SECURITY_ERROR_DSTKEY_IDX_ERR	Destination key index error	-49009
SECURITY_ERROR_SRCKEY_IDX_ERR	Source key index error	-49010
SECURITY_ERROR_KEY_LEN_ERR	Key length error	-49011
SECURITY_ERROR_NO_ICC	IC card does not exist	-49012
SECURITY_ERROR_ICC_NO_INIT	IC card not initialized	-49013
SECURITY_ERROR_GROUP_IDX_ERR	DUKPT group index error	-49014
SECURITY_ERROR_PARAM_PTR_NULL	Pointer parameter is null	-49015
SECURITY_ERROR_TAMPERED	PED is tampered / under attack	-49016
SECURITY_ERROR_PED	PED general error	-49017
SECURITY_ERROR_NOMORE_BUF	No available buffer	-49018
SECURITY_ERROR_NEED_ADMIN	Admin privileges required	-49019
SECURITY_ERROR_DUKPT_OVERFLOW	DUKPT overflow	-49020
SECURITY_ERROR_KCV_CHECK_FAIL	KCV verification failed	-49021
SECURITY_ERROR_SRCKEY_TYPE_ERR	Source key type error	-49022
SECURITY_ERROR_UNSPT_CMD	Command not supported	-49023

SECURITY_ERROR_COMM_ERR	Communication error	-49024
SECURITY_ERROR_PED_DATA_RW_FAIL	PED data read/write failed	-49025
SECURITY_ERROR_ICC_CMD_ERR	IC card operation error	-49026
SECURITY_ERROR_KEY_VALUE_INVALID	Invalid or all-zero key value	-49027
SECURITY_ERROR_KEY_VALUE_EXIST	Key value already exists	-49028
SECURITY_ERROR_KEY_INDEX_NOT_SELECT_OR_NOT_MATCH	Key index not selected or not matched	-49029
SECURITY_ERROR_DUKPT_NEED_INC_KSN	DUKPT KSN needs to be incremented first	-49030
SECURITY_ERROR_DUKPT_NO_KCV	DUKPT has no KCV	-49031
SECURITY_ERROR_MAC_ERR	MAC verification error	-49032
SECURITY_ERROR_CRC_ERR	CRC verification error	-49033
SECURITY_ERROR_ALG_ERR	Algorithm error	-49034
SECURITY_ERROR_STATE	PED state error	-49035
SECURITY_ERROR_REQ_SSA	Sensitive service authorization required	-49036
SECURITY_ERROR_NEED_RESET	SP needs to be reset	-49037
SECURITY_ERROR_RNG	Random number generation error	-49038
SECURITY_ERROR_NEED_RMT_AUTH	Remote authorization required	-49039
SECURITY_ERROR_TAMPER_HW	Tamper hardware error	-49040
SECURITY_ERROR_PAN_LEN	PAN length error	-49041
SECURITY_ERROR_DATA_LEN	Data length / block alignment error	-49042
SECURITY_ERROR_KEY_LEN	Key length error	-49043
SECURITY_ERROR_CMAC	CMAC calculation failed	-49044
SECURITY_ERROR_READ_CFG	Configuration file read error	-49045
SECURITY_ERROR_KEY_STRENGTH	Key strength error	-49046
SECURITY_ERROR_PARAM	Parameter error	-49047
SECURITY_ERROR_WORK_MODE	Work mode error	-49048
SECURITY_ERROR_KEY_INDEX	Key index error	-49049
SECURITY_ERROR_KEYBLOCK_VER	TR-31 KeyBlock version error	-49050
SECURITY_ERROR_FORBIDDEN	Operation forbidden	-49051
SECURITY_ERROR_INVALID_SIGN	Signature verification failed	-49052
SECURITY_ERROR_INVALID_USE	Invalid key usage	-49053
SECURITY_ERROR_RSA_GEN	RSA key pair generation failed	-49054

SECURITY_ERROR_RSA_CALC	RSA calculation error	-49055
SECURITY_ERROR_NOT_INIT	PED not initialized	-49056
SECURITY_ERROR_SEK_ACCESS_FAIL	SEK access failed	-49057
SECURITY_ERROR_SEK_INVALID	SEK validity verification failed	-49058
SECURITY_ERROR_SM2_INIT	SM2 initialization failed	-49059
SECURITY_ERROR_SM2_GENKEY	SM2 key pair generation failed	-49060
SECURITY_ERROR_SM2_WRITE	SM2 key write failed	-49061
SECURITY_ERROR_SM2_ERASE	SM2 key erase failed	-49062
SECURITY_ERROR_SM2_CALC	SM2 calculation error	-49063
SECURITY_ERROR_KSN_LEN	KSN length error	-49064
SECURITY_ERROR_KEY_USAGE	Key usage error	-49065
SECURITY_ERROR_KSN_OVERFLOW	KSN overflow	-49066
SECURITY_ERROR_KSN_MODE	KSN mode error	-49067
SECURITY_ERROR_DERIVED_TYPE	Derived key type error	-49068
SECURITY_ERROR_TYPE_INVALID	Derived key type invalid	-49069
SECURITY_ERROR_MODE	Invalid mode	-49070

4.8 EmvError

Error Code	Error Description	Error Value
EMV_AID_OTHER_ERROR	Other AID parameter error	-80100
EMV_APPLE_TERMINAL_OTHER_ERROR	Other Apple Terminal configuration error	-80400
EMV_CAPK_OTHER_ERROR	Other CAPK error	-80200
EMV_DELETE_AID_ERROR	Error deleting AID parameter	-80109
EMV_DELETE_APPLE_MERCHANT_ERROR	Error deleting Apple Merchant	-80405
EMV_DELETE_APPLE_TERMINAL_ERROR	Error deleting Apple Terminal	-80402
EMV_DELETE_CAPK_ERROR	Error deleting CAPK	-80209
EMV_DELETE_DRL_ERROR	Error deleting DRL configuration	-80310
EMV_DELETE_EXCEPTION_FILE_ERROR	Error deleting ExceptionFile	-80305
EMV_DELETE_REVOCATION_IPK_ERROR	Error deleting RevocationIPK	-80308
EMV_DELETE_SERVICE_ERROR	Error deleting RuPay Service	-80314
EMV_DRL_OTHER_ERROR	Other DRL configuration error	-80309
EMV_GET_APPLE_MERCHANT_ERROR	Error retrieving Apple Merchant	-80404
EMV_GET_DRL_ERROR	Error retrieving DRL configuration	-80311
EMV_GET_KERNEL_VERSION_ERROR	Error retrieving Kernel version	-80316
EMV_GET_SERVICE_ERROR	Error retrieving RuPay Service	-80315
EMV_GET_TERMINAL_ERROR	Error retrieving terminal parameters	-80302

EMV_OTHER_ERROR	Other EMV error	-80001
EMV_SERVICE_OTHER_ERROR	Other RuPay Service error	-80312
EMV_SET_AID_ERROR	Error setting AID parameter	-80108
EMV_SET_AID_ERROR_AID_NULL	Error loading AID parameter, AID is null	-80101
EMV_SET_AID_ERROR_DDOL_NULL	Error loading AID parameter, DDOL is null	-80103
EMV_SET_AID_ERROR_TACDEFAULT_NULL	Error loading AID parameter, TACDefault is null	-80107
EMV_SET_AID_ERROR_TACDENIAL_NULL	Error loading AID parameter, TACDenial is null	-80105
EMV_SET_AID_ERROR_TACONLINE_NULL	Error loading AID parameter, TACOnline is null	-80106
EMV_SET_AID_ERROR_TDOL_NULL	Error loading AID parameter, TDOL is null	-80104
EMV_SET_AID_ERROR_VERSION_NULL	Error loading AID parameter, Version is null	-80102
EMV_SET_APPLE_MERCHANT_ERROR	Error setting Apple Merchant	-80403
EMV_SET_APPLE_TERMINAL_ERROR	Error setting Apple Terminal	-80401
EMV_SET_CAPK_ERROR	Error setting CAPK	-80208
EMV_SET_CAPK_ERROR_ALGORITHM_IND_NULL	Error loading CAPK parameter, AlgorithmInd is null	-80206
EMV_SET_CAPK_ERROR_CAPK_INDEX_NULL	Error loading CAPK parameter, CapkIndex is null	-80202
EMV_SET_CAPK_ERROR_CHECK_SUM_NULL	Error loading CAPK parameter, Checksum is null	-80205
EMV_SET_CAPK_ERROR_Exponent_NULL	Error loading CAPK parameter, Exponent is null	-80204
EMV_SET_CAPK_ERROR_HASH_IND_NULL	Error loading CAPK parameter, HashInd is null	-80207
EMV_SET_CAPK_ERROR_MODULE_NULL	Error loading CAPK parameter, Module is null	-80203
EMV_SET_CAPK_ERROR_RID_NULL	Error loading CAPK parameter, RID is null	-80201
EMV_SET_CARD_INFO_RESPONSE_ERROR	Error setting card confirmation result	-81004
EMV_SET_DRL_ERROR	Error setting DRL configuration	-80309
EMV_SET_EXCEPTION_FILE_ERROR	Error setting ExceptionFile	-80304
EMV_SET_KERNEL_ERROR	Error setting KernelTag	-80317
EMV_SET_ONLINE_RESPONSE_ERROR	Error setting online result	-81006
EMV_SET_PIN_RESPONSE_ERROR	Error setting PIN result	-81005
EMV_SET_REVOCATION_IPK_ERROR	Error setting RevocationIPK	-80307
EMV_SET_SELECT_RESPONSE_ERROR	Error setting multi-application selection result	-81003

EMV_SET_SELECT_RESPONSE_POSITION_ERROR	Error confirming position in multi-application selection result	-81007
EMV_SET_SERVICE_ERROR	Error setting RuPay Service	-80313
EMV_SET_TERMINAL_ERROR	Error loading terminal parameters	-80301
EMV_START_TRANSACTION_ERROR	Error starting transaction	-81001
EMV_STOP_TRANSACTION_ERROR	Error stopping transaction	-81002
EMV_SUCCESS	Success	0
EMV_TERMINAL_EXCEPTION_FILE_OTHER_ERROR	Other ExceptionFile error	-80303
EMV_TERMINAL_OTHER_ERROR	Other terminal parameter error	-80300
EMV_TERMINAL_REVOCATION_IPK_OTHER_ERROR	Other RevocationIPK error	-80306
EMV_APPROVED	Transaction approved	1
EMV_APPROVED_ONLINE	Online approved	2
EMV_DECLINED	Transaction declined	3
EMV_FORCE_APPROVED	Force approved	4
EMV_DELAYED_APPROVED	Delayed approved	5
APPLE_VAS_APPROVED	Apple VAS approved	6
MONYX_READ_SUCCESS	Monyx read success	7
SAMSUNG_READ_SUCCESS	Samsung read success	8
EMV_EXCEPTION_ERROR	Exception error	-89001
EMV_PARAMETER_ERROR	Parameter error	-89002
EMV_CANCEL	Card detection phase, stopTransaction called to exit the transaction	-1
EMV_TIMEOUT	Card detection phase, no card found before timeout	-2
EMV_COMMAND_FAIL	APDU communication failed during CTLS transaction	-3
EMV_FALLBACK	ICC transaction, card damaged or inserted incorrectly causing reset failure (e.g. card inserted backwards). Can be treated as card damage, allowed to fall back to magnetic stripe	-4
EMV_MULTI_CONTACTLESS	CTLS transaction, multiple cards detected during card detection	-5
EMV_OTHER_ICC_INTERFACE	MAG transaction, track ServiceCode equals 02 or 06, magnetic stripe transaction not allowed, switch to ICC transaction. Usually occurs after contactless transaction failure	-6

EMV_APP_BLOCKED	ICC / CTLS transaction, SELECT command returns 6283. Card problem, card locked	-7
EMV_CARD_BLOCKED	ICC transaction, PSE command returns 6A81, or SELECT command returns 6A81	-8
EMV_APP_EMPTY	ICC / CTLS transaction, no matching AID parameter	-9
EMV_NOT_ALLOWED	ICC transaction, GAC command returns error 9F27	-10
EMV_NOT_ACCEPTED	ICC transaction, GAC command returns error 9F27	-11
EMV_TERMINATED	ICC / CTLS transaction, transaction terminated	-12
EMV_SEE_PHONE	CTLS transaction, SEE PHONE	-13
EMV_OTHER_INTERFACE	CTLS transaction, Try another interface. Swiping the magnetic stripe of a bank card with an IC chip is not allowed unless FALLBACK is set to true when starting the transaction	-14
EMV_APDU_FAIL	APDU Fail (unused)	-15
EMV_ENCRYPT_ERROR	Transaction encryption failed	-30
EMV_UNENCRYPTED	Transaction not encrypted	-31
EMV_PIN_MANUALLY_CANCEL	PIN input manually cancelled	-32
EMV_PIN_TIMEOUT	PIN input timeout	-33
EMV_TERMINATE_ON_CARD_READ_ERROR	Transaction terminated due to card read error	-34
EMV_TERMINATE_ON_CARD_READ_ERROR_THREE_TIMES	Transaction terminated due to three consecutive card read errors	-35
EMV_INTERRUPTED_ON_COMMUNICATION	Transaction aborted due to communication interruption	-36
APPLE_VAS_UNTREATED	Apple VAS untreated	-40
EMV_CARD_ERROR	Abecs Fallback	-40
APPLE_VAS_FAILED	Apple VAS failed	-41
APPLE_VAS_WAITING_INTERVENTION	Apple VAS waiting for manual intervention	-42
APPLE_VAS_WAITING_ACTIVATION	Apple VAS waiting for activation	-43
EMV_TERMINATED_ON_MCCS	Transaction terminated due to MCCS	-44
EMV_GPO_6985	GPO command returns 6985	-50
EMV_READ_RECORD_ERROR	Read Record command error	-80
EMV_DATA_ERROR	Card data error	-81

4.9 PosEmvErrorCode

Error Code	Error Description	Error Value
APPLE_VAS_APPROVED	Apple Vas Approved.	6
APPLE_VAS_FAILED	Apple Vas Failed.	-41
APPLE_VAS_UNTREATED	Apple Vas Untreated.	-40
APPLE_VAS_WAITING_ACTIVATION	Apple Vas Waiting Activation.	-43
APPLE_VAS_WAITING_INTERVENTION	Apple Vas Waiting Intervention.	-42
EMV_APP_BLOCKED	Application Blocked.	-7
EMV_APP_EMPTY	No Application.	-9
EMV_APPROVED	Transaction Approved.	1
EMV_APPROVED_ONLINE	Transaction Online Approved.	2
EMV_CANCEL	Trans Cancel.	-1
EMV_CARD_BLOCKED	Card Locked.	-8
EMV_CARD_ERROR	Abecs Fallback.	-40
EMV_COMMAND_FAIL	Read the Card Fail.	-3
EMV_DATA_ERROR	Card Data Error.	-81
EMV_DECLINED	Transaction Declined.	3
EMV_DELAYED_APPROVED	Transaction Delayed Approved.	5
EMV_ENCRYPT_ERROR	Trade Encrypt Error.	-30
EMV_FALLBACK	Fallback.	-4
EMV_FORCE_APPROVED	Transaction Force Approved.	4
EMV_GPO_6985	Command GPO 6985.	-50
EMV_MULTI_CONTACTLESS	Read Multi Contactless.	-5
EMV_NOT_ACCEPTED	Not Accepted.	-11
EMV_NOT_ALLOWED	Not Allowed.	-10
EMV_OK	Handle OK.	0
EMV_OTHER_ERROR	Other Error Exceptions.	-999
EMV_OTHER_ICC_INTERFACE	Not Pure Magnetic Strip.	-6
EMV_OTHER_INTERFACE	Try Another Interface.	-14
EMV_READ_RECORD_ERROR	Command Read Record Error.	-80
EMV_SEE_PHONE	See Phone.	-13
EMV_TERMINATED	Terminated.	-12
EMV_TERMINATED_ON_MCCS	Terminated On MCCS.	-44
EMV_TIMEOUT	Trans Timeout.	-2
EMV_UNENCRYPTED	Trade Unencrypted.	-31
EXCEPTION_ERROR	Exception Error.	255
PARAMETER_ERROR	Parameter Error.	254
MONYX_READ_SUCCESS	Monyx Read Success.	7
SAMSUNG_READ_SUCCESS	Samsung Read Success.	8
EMV_APDU_FAIL	Apdu Fail.	-15
EMV_PIN_MANUALLY_CANCEL	Pin Manually Cancel Input.	-32
EMV_PIN_TIMEOUT	Pin Timeout.	-33
EMV_TERMINATE_ON_CARD_READ_ERROR	Terminate On Card Read Error.	-34

4.10 EcrError

Error Code	Error Description	Error Value
ECR_CONNECT_CLOSE_FAILED	Failed to close connection	-90006
ECR_CONNECT_FAILED	Connection failed	-90003
ECR_CONNECT_INFO_EMPTY	Connection parameter info is empty	-90001
ECR_CONNECT_INFO_FORMAT_EXCEPTION	Connection parameter format exception	-90002
ECR_ERROR_INIT	ECR module not initialized	-90000
ECR_FD_EMPTY	File handle is null	-90004
ECR_FD_FORMAT_EXCEPTION	File handle format exception	-90005
ECR_WRITE_DATA_EMPTY	Write data is empty	-90007
ECR_WRITE_DATA_FAILED	Write data exception	-90008

5. Entity Class Definition

5.1 com.kozen.financial.aidl.emv.EmvCapk

Constant Name	Type	Value	Description
ALGO_IND_RSA	int	1	-
ALGO_IND_SM	int	4	-
AlgorithmInd	byte	-	Algorithm Flag.
CapkIndex	byte	-	Capk Index.
Checksum	byte[]	-	Checksum.
Exponent	byte[]	-	Exponent.
HASH_IND_NOT	int	0	-
HASH_IND_SHA1	int	1	-
HashInd	byte	-	HASH Algorithm Flag.
Module	byte[]	-	Module.
RID	byte[]	-	Application Registration Service Provider ID.

5.2 com.kozen.financial.aidl.emv.EmvAid

Constant Name	Type	Description
AcquirerIdentifier	byte[]	Acquirer Identifier. Tag: 9F01.Value Type : byte[]. Required Field.
AdditionalTerminalCapabilities	byte[]	Additional Terminal Capabilities. Tag: 9F40.Value Type : byte[].Required Field.
AID	byte[]	Application Id. Tag: 9F06. Value Type: byte[]. Required Field.
CombinationData	byte[]	Combination Data.

		public byte[] CombinationData
		Combination Data. Set Parameters for AID Will Overwrite the Default Parameters. Not Related to CombinationType.
		eg. aid.CombinationData = getKernel(getLimit(999999999999L,
		999999999999L,
		999999999999L,
		999999999999L),
		HexUtil.parseHex("9F660436004000"), null, null, null, null);
		The getKernel Function is Defined to Set Parameters for Different Card Organizations
		Can Set EMV Standard Tag.
		"DF10", "DF11", "DF12", "DF13", "DF14", "DF17" in this Function are Custom Tags.
		Just Pass in the Tag and Value that Need to be Set as Parameters.
		eg. 9F660436004000.
		private static byte[] getKernel(byte[] kernel, byte[] visa, byte[] unionpay, byte[] mastercard, byte[] discover, byte[] mir) {
		BerTlvBuilder tlvBuilder = new BerTlvBuilder();
		if (kernel != null) {
		tlvBuilder.addBytes(new BerTag("DF10"), kernel);
		}
		if (visa != null) {
		tlvBuilder.addBytes(new BerTag("DF11"), visa);
		}
		if (unionpay != null) {
		tlvBuilder.addBytes(new BerTag("DF12"), unionpay);
		}
		if (mastercard != null) {
		tlvBuilder.addBytes(new BerTag("DF13"), mastercard);
		}
		if (discover != null) {
		tlvBuilder.addBytes(new BerTag("DF14"), discover);
		}
		if (mir != null) {
		tlvBuilder.addBytes(new BerTag("DF17"), mir);
		}
		return tlvBuilder.buildArray();
		}
		Set CVM Parameters.
		"DF01", "DF02", "DF03", "DF04" in this Function are Custom Tags.
		Just Pass in the Tag and Value that Need to be Set as Parameters. eg.
		999999999999L,
		999999999999L,
		999999999999L,
		999999999999L.

		private static byte[] getLimit(long contactlessLimit, long contactlessCVMLimit, long contactlessFloorLimit, long contactlessDynamicLimit) {
		BerTlvBuilder tlvBuilder = new BerTlvBuilder();
		tlvBuilder.addBytes(new BerTag("DF01"),
		getAmount(contactlessLimit));
		tlvBuilder.addBytes(new BerTag("DF02"),
		getAmount(contactlessCVMLimit));
		tlvBuilder.addBytes(new BerTag("DF03"),
		getAmount(contactlessFloorLimit));
		tlvBuilder.addBytes(new BerTag("DF04"),
		getAmount(contactlessDynamicLimit));
		return tlvBuilder.buildArray();
		}
		private static byte[] getAmount(long value) {
		StringBuilder builder = new StringBuilder(12);
		builder.append(value);
		while (builder.length() < 12) {
		builder.insert(0, '0');
		}
		return HexUtil.parseHex(builder.toString());
		}

CombinationType	int	Combination Type. Not Usage. Value Type : int.
ContactlessCVMLimit	int	Contactless CVM Required Limit. Value Type : int. Optional Field.
ContactlessCVMLimitL	long	Contactless CVM Required Limit. Value Type : long. Optional Field.
ContactlessFloorLimit	int	Contactless Floor Limit. Value Type : int. Optional Field.
ContactlessFloorLimitL	long	Contactless Floor Limit. Value Type : long. Optional Field.
ContactlessTransLimit	int	Contactless Transaction Limit. Value Type : int. Optional Field.
ContactlessTransLimitL	long	Contactless Transaction Limit. Value Type : long. Optional Field.
dDOL	byte[]	Default dDOL. Tag: 97. Value Type : byte[]. Required Field.
DynamicTransLimit	int	Dynamic Trans Limit. Value Type : int. Optional Field.
DynamicTransLimitL	long	Dynamic Trans Limit. Value Type : long. Optional Field.
FloorLimit	int	Floor Limit. Value Type : int. Optional Field.
MaxTargetPercentage	int	The Maximum Target Percentage of Offset Random

		Selection. Value Type : int. Optional Field.
MerchantCategoryCode	byte[]	Merchant Category Code. Tag: 9F15. Value Type : byte[].Required Field.
SelectIndicator	boolean	Application Select Indicator: True: FULL_MATCH, False: PART_MATCH. Value Type : boolean. Optional Field.
TACDefault	byte[]	TAC-Default. Tag: DF8120.Value Type: byte[].Required Field.
TACDenial	byte[]	TAC-Denial. Tag : DF8121.Value Type : byte[].Required Field.
TACOnline	byte[]	TAC-Online. Tag : DF8122.Value Type : byte[].Required Field.
TargetPercentage	int	Target Percentage of Random Selection. Value Type : int. Optional Field.
tDOL	byte[]	Default tDOL. Tag : 9F49.Value Type : byte[].Required Field.
TerminalCapabilities	byte[]	Terminal Capability. Value Type : byte[]. Required Field.
TerminalCountryCode	byte[]	Terminal Country Code. Tag : 9F1A.Value Type : byte[].Required Field.
TerminalRiskManagementData	byte[]	Terminal Risk Management Data. Tag : 9F1D.Value Type : byte[].Required Field.
TerminalType	byte[]	Terminal Type. Value Type : byte[]. Required Field.
Threshold	int	Threshold of Bias Random Selection. Value Type : int. Optional Field.
TransCurrencyCode	byte[]	Trans Currency Code. Tag : 5F2A.Value Type : byte[]. Required Field.
TransCurrencyExp	byte[]	Trans Currency Exp. Tag : 5F36.Value Type : byte[]. Required Field.
TypeIndicator	boolean	Application Type Indicator: True: Contactless, False: Both for Contact and Contactless. Value Type : boolean. Optional Field.
Version	byte[]	Application Version. Tag : 9F09.Value Type : byte[]. Required Field.

5.3 com.kozen.financial.aidl.emv.EmvRevocationIPK

Constant Name	Type	Description
CapkIndex	byte	Capk Index Value Type : byte[]. Required Field.
RID	byte[]	RID

		Tag : 9F06.Value Type : byte[]. Required Field.
SerialNo	byte[]	SerialNumber Value Type : byte[]. Required Field.

5.4 com.kozen.financial.aidl.emv.EmvExceptionFile

Constant Name	Type	Description
PAN	byte[]	Primary Account Number. (PAN). Value Type : byte[]. Required Field.
SerialNo	byte[]	Serial Number. Value Type : byte[]. Required Field.

5.5 ConstantCardReader

Constant Name	Type	Value
ATR	String	"cardAtr"
ATS	String	"cardAs"
CARD_ATTRIBUTE	String	"cardAttribute"
CARD_CATEGORY	String	"cardCategory"
CARD_CHANNEL	String	"cardChannel"
CARD_SERIAL_NUM	String	"cardSerialNum"
CARD_TYPE	String	"cardType"
EXPIRED_DATE	String	"cardExpDate"
ID_FOR_MANUFACTURER	String	"IDm"
PAN	String	"cardPan"
PARAMETER_FOR_MANUFACTURER	String	"PMm"
REQUEST_DATA	String	"RequestData"
SERVICE_CODE	String	"cardServiceCode"
TIMEOUT	String	"cardTimeout"
TRACK1	String	"cardTrack1"
TRACK2	String	"cardTrack2"
TRACK3	String	"cardTrack3"

5.6 ConstantCardReader.CardType

Constant Name	Type	Value
ALL	int	0
CONTACT	int	2
CONTACTLESS	int	4
FELICA	int	32
MAGNETIC	int	1
NFC_TAG	int	64
PSAM	int	16
MIFARE_CLASSIC	int	128
MIFARE_DESFIRE	int	512

MIFARE_PLUS	int	256
MIFARE_ULTRALIGHT	int	8

5.7 ConstantEmv.POIEmvCoreManager

Constant Name	Type	Value
CMD_AMOUNT_CONFIG	int	1
CMD_CARD_READ_SUCCESS	int	64
CMD_GAC1	int	18
CMD_GAC2	int	19
CMD_GPO_BEFORE	int	49
CMD_GPO_FILTER	int	16
CMD_ISSUER_REFERRAL	int	2
CMD_READ_RECORD	int	17
CMD_READ_RECORD_FILTER	int	17
CMD_SELECT_AFTER	int	48
CMD_SELECT_APPLICATION	int	16
CMD_SELECT_KERNEL	int	32
CMD_TRY_OTHER_APPLICATION	int	0
CMD_APDU_CMD_LOG	int	65
CMD_APDU_RESP_LOG	int	66
CMD_KERNEL_LOG	int	67
DEVICE_CONTACT	int	1
DEVICE_CONTACTLESS	int	2
DEVICE_MAGSTRIPE	int	4
DEVICE_MIFARE_CLASSIC	int	8
DEVICE_MIFARE_DESFIRE	int	64
DEVICE_MIFARE_PLUS	int	32
DEVICE_MIFARE_ULTRALIGHT	int	16
DEVICE_VICC	int	128
DEVICE_FELICA	int	10
EMV_ADMINISTRATIVE	int	7
EMV_BALANCE_ENQUIRY	int	13
EMV_BALANCE_UPDATE	int	14
EMV_CARD_AMEX	int	5
EMV_CARD_DISCOVER	int	4
EMV_CARD_EFTPOS	int	11
EMV_CARD_INTERAC	int	10
EMV_CARD_JCB	int	6
EMV_CARD_MASTERCARD	int	3
EMV_CARD_MIR	int	7
EMV_CARD_NOT	int	0
EMV_CARD_PURE	int	9
EMV_CARD_RUPAY	int	8

EMV_CARD_UNIONPAY	int	2
EMV_CARD_VISA	int	1
EMV_CARD_MONYX	int	12
EMV_CASH	int	3
EMV_CASHBACK	int	4
EMV_DEPOSIT	int	10
EMV_DISBURSEMENT	int	8
EMV_GOODS	int	1
EMV_INQUIRY	int	11
EMV_MONEY_ADD	int	12
EMV_PAYMENT	int	6
EMV_REFUND	int	9
EMV_SERVICE	int	2
EMV_SERVICE_CREATION	int	16
EMV_TRANSFER	int	5
EMV_VOID	int	15
GET_LIB_VERSION	int	0
GET_VERSION_AMEX	int	6
GET_VERSION_APPLE	int	16
GET_VERSION_CL1	int	14
GET_VERSION_DISCOVER	int	5
GET_VERSION_EFTPOS	int	12
GET_VERSION_EMV	int	1
GET_VERSION_INTERAC	int	11
GET_VERSION_JCB	int	7
GET_VERSION_L1	int	13
GET_VERSION_MASTERCARD	int	4
GET_VERSION_MIR	int	8
GET_VERSION_PURE	int	10
GET_VERSION_RUPAY	int	9
GET_VERSION_UNIONPAY	int	3
GET_VERSION_VISA	int	2
PIN_ENCIPHER_PIN	int	2
PIN_ONLINE_PIN	int	1
PIN_PLAIN_PIN	int	0

5.8 ConstantEmv.POIEmvCoreManager.AppleTerminalConstraints

Constant Name	Type	Value
CAPABILITY	String	"capability"
CAPABILITY_DUAL_MODE	int	1
CAPABILITY_PAYMENT_ONLY	int	3
CAPABILITY_SINGLE_MODE	int	0
CAPABILITY_VAS_ONLY	int	2

DATA	String	"data"
PROTOCOL	String	"protocol"
PROTOCOL_FULL_VAS	int	1
PROTOCOL_URL_ONLY	int	0
TAG_APPLE_SET_DELIMITER	String	"DF01"
TAG_APPLE_SET_FILTER	String	"9F2B"
TAG_APPLE_SET_MERCHANT_ID	String	"9F25"
TAG_APPLE_SET_MERCHANT_URL	String	"9F29"

5.9 ConstantEmv.POIEmvCoreManager.EmvCardInfoConstraints

Constant Name	Type	Value
ATR	String	"atr"
CARD	String	"card"
DATA	String	"data"
OUT_AMOUNT	String	"amount"
OUT_AMOUNT_OTHER	String	"amountOther"
OUT_CONFIRM	String	"confirm"
OUT_TLV	String	"tlv"
OUT_TVR	String	"tvr"
TRACK1	String	"track1"
TRACK2	String	"track2"
TRACK3	String	"track3"

5.10 ConstantEmv.POIEmvCoreManager.EmvDrlConstraints

Constant Name	Type	Value
CONFIG	String	"Config"
TAG_DRL_SET_CVM_REQUIRED_LIMIT	String	"DF24"
TAG_DRL_SET_DELIMITER	String	"DF01"
TAG_DRL_SET_ENTRY_POINT	String	"DF30"
TAG_DRL_SET_FLOOR_LIMIT	String	"DF25"
TAG_DRL_SET_PROGRAM_ID	String	"9F5A"
TAG_DRL_SET_STATUS_ZERO_AMOUNT	String	"DF32"
TAG_DRL_SET_TRANSACTION_LIMIT	String	"DF23"
TYPE_AMEX	int	2
TYPE_VISA	int	1

5.11 ConstantEmv.POIEmvCoreManager.EmvOnlineConstraints

Constant Name	Type	Value
APPLE_DATA	String	"appleData"
APPLE_MERCHANT	String	"appleMerchant"
APPLE_RESULT	String	"appleResult"
EMV_DATA	String	"emvData"

EMV_ONLINE_APPROVE	int	0
EMV_ONLINE_DENIAL	int	2
EMV_ONLINE_FAIL	int	1
EMV_ONLINE_REFER_TO_CARD_ISSUER	int	3
ENCRYPT_DATA	String	"encryptData"
ENCRYPT_RESULT	String	"encryptResult"
OUT_AUTH_CODE	String	"outAuthCode"
OUT_AUTH_DATA	String	"outAuthData"
OUT_AUTH_RESP_CODE	String	"outAuthRespCode"
OUT_ISSUER_SCRIPT	String	"outIssuerScript"
OUT_SPECIAL_AUTH_RESP_CODE	String	"outSpecialAuthRespCode"

5.12 ConstantEmv.POIEmvCoreManager.EmvPinConstraints

Constant Name	Type	Value
OUT_PIN_BLOCK	String	"outPinBlock"
OUT_PIN_TRY_COUNTER	String	"outPinTryCounter"
OUT_PIN_VERIFY_RESULT	String	"outPinVerifyResult"
PIN_BLOCK_FORMAT	String	"pinBlockFormat"
PIN_BYPASS	String	"pinBypass"
PIN_BYPASS_ALL_TYPE	String	"pinBypassAllType"
PIN_CARD	String	"pinCard"
PIN_CARD_RANDOM	String	"pinCardRandom"
PIN_COUNTER	String	"pinCounter"
PIN_DUKPT_KEY_LENGTH	String	"pinDukptKeyLength"
PIN_ENCRYPT	String	"pinEncrypt"
PIN_EXPONENT	String	"pinExponent"
PIN_IS_ORDER	String	"isOrder "
PIN_ISO_FMT0	int	0
PIN_ISO_FMT1	int	1
PIN_ISO_FMT1_SM4	int	5
PIN_ISO_FMT2	int	2
PIN_ISO_FMT2_SM4	int	6
PIN_ISO_FMT3	int	3
PIN_ISO_FMT3_SM4	int	7
PIN_ISO_FMT4	int	4
PIN_KEY_ALGORITHM	String	"pinKeyAlgorithm"
PIN_KEY_ALGORITHM_AES	int	16
PIN_KEY_ALGORITHM_TDES	int	0
PIN_KEY_INDEX	String	"pinKeyId"
PIN_KEY_MODE	String	"pinKeyMode"
PIN_KEY_MODE_DUKPT	int	2
PIN_KEY_MODE_TPK	int	1
PIN_LENGTH_LIMIT	String	"lengthLimit"

PIN_MODULE	String	"pinModule"
PIN_TIMEOUT	String	"pinTimeout"
PIN_TYPE	String	"pinType"
SKIP_PIN_VERIFY	int	100
VERIFY_CANCELED	int	4
VERIFY_ERROR	int	3
VERIFY_MANUALLY_CANCELED	int	5
VERIFY_NO_PASSWORD	int	1
VERIFY_PIN_BLOCK	int	2
VERIFY_SUCCESS	int	0
VERIFY_TIMEOUT	int	6

5.13 ConstantEmv.POIEmvCoreManager.EmvResultConstraints

Constant Name	Type	Value
APPLE_DATA	String	"appleData"
APPLE_MERCHANT	String	"appleMerchant"
APPLE_RESULT	String	"appleResult"
CVM	String	"cvm"
CVM_CONFIRMATION_CODE_VERIFIED	int	2
CVM_NO_CVM	int	0
CVM_SIGNATURE	int	1
EMV_DATA	String	"emvData"
ENCRYPT_DATA	String	"encryptData"
ENCRYPT_RESULT	String	"encryptResult"
SCRIPT_RESULT	String	"scriptResult"
SECOND_TAP_CANCEL	int	3
SECOND_TAP_FAIL	int	1
SECOND_TAP_RESULT	String	"secondTapResult"
SECOND_TAP_SUCCESS	int	0
SECOND_TAP_TIMEOUT	int	2

5.14 ConstantEmv.POIEmvCoreManager.EmvServiceConstraints

Constant Name	Type	Value
CONFIG	String	"Config"
TAG_PRMACQ_SET_DELIMITER	String	"DF02"
TAG_PRMACQ_SET_INDEX	String	"DF30"
TAG_PRMACQ_SET_KCV	String	"DF32"
TAG_PRMACQ_SET_KEY	String	"DF31"
TAG_SERVICE_SET_DATA	String	"DF19"
TAG_SERVICE_SET_DELIMITER	String	"DF01"
TAG_SERVICE_SET_ID	String	"DF16"
TAG_SERVICE_SET_MANAGEMENT	String	"DF18"

TAG_SERVICE_SET_PRIORITY	String	"DF17"
TAG_SERVICE_SET_PRMACQ	String	"DF21"
TAG_SERVICE_SET_PRMISS	String	"DF20"

5.15 ConstantEmv.POIEmvCoreManager.EmvTerminalConstraints

Constant Name	Type	Value
BYPASS_PIN_ENTRY	String	"BypassPINEntry"
CARD_HOLDER_CONFIRM	String	"CardHolderConfirm"
CONFIG	String	"Config"
DEFAULT_DDOL	String	"DefaultDDOL"
DEFAULT_TDOL	String	"DefaultTDOL"
EXCEPTION_FILE	String	"ExceptionFile"
FLOOR_LIMIT_CHECKING	String	"FloorLimitChecking"
FORCED_ACCEPT	String	"ForcedAccept"
FORCED_ONLINE	String	"ForcedOnline"
GET_DATA_FOR_PIN_COUNTER	String	"GetDataForPINCounter"
IFD_SERIAL_NUMBER	String	"IfdSerialNumber"
ISSUER_REFERRAL	String	"IssuerReferral"
LANGUAGE_SELECT	String	"LanguageSelect"
MERCHANT_CATEGORY_CODE	String	"MerchantCategoryCode"
MERCHANT_ID	String	"MerchantId"
MERCHANT_NAME	String	"MerchantName"
PSE	String	"Pse"
RANDOM_TRANSACTION_SELECTION	String	"RandomTransactionSelection"
REVOCATION_ISSUER_PUBLIC_KEY	String	"RevocationIssuerPublicKey"
SETTINGS_AMEX	int	24
SETTINGS_DISCOVER	int	23
SETTINGS_EFTPOS	int	30
SETTINGS_INTERAC	int	29
SETTINGS_JCB	int	25
SETTINGS_MASTERCARD	int	22
SETTINGS_MIR	int	26
SETTINGS_PURE	int	28
SETTINGS_RUPAY	int	27
SETTINGS_UNIONPAY	int	21
SETTINGS_VISA	int	20
SUBSEQUENT_BYPASS_PIN_ENTRY	String	"SubsequentBypassPINEntry"
TAG_AMEX_SET_ENTRY_POINT	String	"DF30"
TAG_AMEX_SET_KERNEL_CONFIG	String	"DF1B"
TAG_AMEX_SET_QUALIFIERS	String	"9F6E"
TAG_AMEX_SET_STATUS_ZERO_AMOUNT	String	"DF32"
TAG_CARD_DATA_INPUT_CAPABILITY	String	"DF8117"

TAG_DISCOVER_SET_ENTRY_POINT	String	"DF30"
TAG_DISCOVER_SET_QUALIFIERS	String	"9F66"
TAG_DISCOVER_SET_STATUS_ZERO_AMOUNT	String	"DF32"
TAG_EFTPOS_SET_ENTRY_POINT	String	"DF30"
TAG_EFTPOS_SET_KERNEL_CONFIG	String	"DF1B"
TAG_EFTPOS_SET_QUALIFIERS	String	"9F66"
TAG_EFTPOS_SET_STATUS	String	"DF31"
TAG_EFTPOS_SET_ZERO_AMOUNT	String	"DF32"
TAG_JCB_SET_ENTRY_POINT	String	"DF30"
TAG_JCB_SET_KERNEL_CONFIG	String	"DF1B"
TAG_JCB_SET_QUALIFIERS	String	"9F53"
TAG_JCB_SET_STATUS	String	"DF31"
TAG_JCB_SET_ZERO_AMOUNT	String	"DF32"
TAG_MASTERCARD_SET_CVM_CAPABILITIES	String	"DF8118"
TAG_MASTERCARD_SET_DEFAULT_UDOL	String	"DF811A"
TAG_MASTERCARD_SET_KERNEL_CONFIG	String	"DF811B"
TAG_MASTERCARD_SET_KERNEL_ID	String	"DF810C"
TAG_MASTERCARD_SET_MAGSTRIPE_APP_VERSION	String	"9F6D"
TAG_MASTERCARD_SET_MAGSTRIPE_CVM_CAPABILITIES	String	"DF811E"
TAG_MASTERCARD_SET_MAGSTRIPE_NO_CVM_CAPABILITIES	String	"DF812C"
TAG_MASTERCARD_SET_MOBILE_SUPPORT_INDICATOR	String	"9F7E"
TAG_MASTERCARD_SET_NO_CVM_CAPABILITIES	String	"DF8119"
TAG_MASTERCARD_SET_RRP_ACCURACY_THRESHOLD	String	"DF8136"
TAG_MASTERCARD_SET_RRP_CAPDU_EXPECTED	String	"DF8134"
TAG_MASTERCARD_SET_RRP_MAX_GRACE	String	"DF8133"
TAG_MASTERCARD_SET_RRP_MIN_GRACE	String	"DF8132"
TAG_MASTERCARD_SET_RRP_MISMATCH_THRESHOLD	String	"DF8137"
TAG_MASTERCARD_SET_RRP_RAPDU_EXPECTED	String	"DF8135"
TAG_MIR_SET_ENTRY_POINT	String	"DF30"
TAG_MIR_SET_QUALIFIERS	String	"9F66"
TAG_MIR_SET_STATUS_ZERO_AMOUNT	String	"DF32"
TAG_PURE_SET_ENTRY_POINT	String	"DF30"
TAG_PURE_SET_KERNEL_CONFIG	String	"DF1B"
TAG_PURE_SET_QUALIFIERS	String	"C7"
TAG_PURE_SET_STATUS	String	"DF31"
TAG_PURE_SET_ZERO_AMOUNT	String	"DF32"
TAG_SECURITY_CAPABILITY	String	"DF811F"
TAG_UNIONPAY_SET_ENTRY_POINT	String	"DF30"
TAG_UNIONPAY_SET_QUALIFIERS	String	"9F66"
TAG_UNIONPAY_SET_STATUS_ZERO_AMOUNT	String	"DF32"
TAG_VISA_SET_ENTRY_POINT	String	"DF30"

TAG_VISA_SET_KERNEL_CONFIG	String	"DF1B"
TAG_VISA_SET_QUALIFIERS	String	"9F66"
TAG_VISA_SET_STATUS_ZERO_AMOUNT	String	"DF32"
TERMINAL_CAPABILITY	String	"TerminalCapability"
TERMINAL_COUNTRY_CODE	String	"TerminalCountryCode"
TERMINAL_ENTRY_MODE	String	"TerminalEntryMode"
TERMINAL_EX_CAPABILITY	String	"TerminalExCapability"
TERMINAL_ID	String	"TerminalId"
TERMINAL_TYPE	String	"TerminalType"
TRANS_CURRENCY_CODE	String	"TransCurrencyCode"
TRANS_CURRENCY_EXP	String	"TransCurrencyExp"
TRANS_REFER_CURRENCY_CODE	String	"TransReferCurrencyCode"
TRANS_REFER_CURRENCY_EXP	String	"TransReferCurrencyExp"
TYPE_AMEX	int	7
TYPE_CONFIG	int	2
TYPE_DISCOVER	int	6
TYPE_INTERAC	int	10
TYPE_MASTERCARD	int	5
TYPE_MIR	int	8
TYPE_RUPAY	int	9
TYPE_TERMINAL	int	1
TYPE_UNIONPAY	int	4
TYPE_VISA	int	3
UNABLE_TO_GO_ONLINE	String	"UnableToGoOnline"
VELOCITY_CHECKING	String	"VelocityChecking"

5.16 ConstantEmv.POIEmvCoreManager.EmvTransDataConstraints

Constant Name	Type	Value
ACCOUNT_MASK_HEAD	String	"accountMaskHead"
ACCOUNT_MASK_TAIL	String	"accountMaskTail"
ACCOUNT_TYPE	String	"accountType"
AMOUNT_CONFIG	String	"amountConfig"
APPLE_VAS	String	"appleVas"
CL_SPECIAL_TYPE	String	"clSpecialType"
CT_SPECIAL_TYPE	String	"ctSpecialType"
DISABLE_APDU_LOG	String	"disableApduLog"
DOUBLE_BCD	String	"doubleBCD"
ENCRYPT_AES_IV	String	"encryptAesIv"
ENCRYPT_BASE64	String	"encryptBase64"
ENCRYPT_CONTACT	String	"encryptContact"
ENCRYPT_CONTACTLESS	String	"encryptContactless"
ENCRYPT_EMV_DATA	String	"encryptEmvData"
ENCRYPT_KEY_INDEX	String	"encryptKeyIndex"

ENCRYPT_KEY_MODE	String	"encryptKeyMode"
ENCRYPT_KEY_MODE_TRANS_ARMOR	int	1
ENCRYPT_MAGSTRIPE	String	"encryptMagstripe"
ENCRYPT_MODE	String	"encryptMode"
ENCRYPT_MODE_CBC	int	2
ENCRYPT_MODE_ECB	int	1
ENCRYPT_OPEN_CONTACT	int	1
ENCRYPT_OPEN_CONTACTLESS	int	2
ENCRYPT_OPEN_MAGSTRIPE	int	4
ENCRYPT_PADDING	String	"encryptPadding"
ENCRYPT_SHA1	String	"encryptSHA1"
ENCRYPT_TRACK_USE_BCD	String	"encryptTrackUseBCD"
ENCRYPT_TRACK2_EXPIRATION_DATE	String	"encryptTrack2ExpirationData"
ENCRYPT_TYPE	String	"encryptType"
ENCRYPT_TYPE_DUKPT_AES_DECRYPT	int	17
ENCRYPT_TYPE_DUKPT_AES_ENCRYPT	int	16
ENCRYPT_TYPE_DUKPT_AES_ENCRYPT_DECRYPT	int	18
ENCRYPT_TYPE_DUKPT_DATA_REQUEST	int	3
ENCRYPT_TYPE_DUKPT_DATA_RESPONSE	int	4
ENCRYPT_TYPE_DUKPT_MAC	int	2
ENCRYPT_TYPE_DUKPT_PIN	int	5
ENCRYPT_TYPE_RSA	int	6
ENCRYPT_TYPE_TDK	int	1
ENCRYPT_TYPE_TTK	int	7
ENCRYPT_VECTOR	String	"encryptVector"
ENCRYPT_WORK_KEY_TYPE	String	"encryptWorkKeyType"
ENCRYPT_WORK_KEY_TYPE_2TDEA	int	0
ENCRYPT_WORK_KEY_TYPE_3TDEA	int	1
ENCRYPT_WORK_KEY_TYPE_AES_128	int	2
ENCRYPT_WORK_KEY_TYPE_AES_192	int	3
ENCRYPT_WORK_KEY_TYPE_AES_256	int	4
GOOGLE_SMART_TAP	String	"googleSmartTap"
OPEN_ENCRYPT	String	"openEncrypt"
PAN_MASK_LEFT_KEEP_COUNT	String	"panMaskLeftKeepCount"
PAN_MASK_RIGHT_KEEP_COUNT	String	"panMaskRightKeepCount"
RSA_TRANS_ARMOR_KEY_ID	String	"rsaTransArmorKeyId"
RSA_TRANS_ARMOR_POS_ID	String	"rsaTransArmorPosId"
SPECIAL_CONTACT	String	"specialContact"
SPECIAL_CONTACT_TIME	String	"specialContactTime"
SPECIAL_MAGSTRIPE	String	"specialMagstripe"
SPECIAL_MAGSTRIPE_TIME	String	"specialMagstripeTime"
SPECIAL_START_MODE	String	"specialStartMode"
SPECIAL_TYPE	String	"specialType"
START_A	int	0

START_B	int	1
START_C	int	2
START_D	int	3
TARNS_COUNTER	String	"tarnsCounter"
TRANS_AMOUNT	String	"transAmount"
TRANS_AMOUNT_OTHER	String	"transAmountOther"
TRANS_DATE	String	"transDate"
TRANS_FALLBACK	String	"transFallback"
TRANS_MODE	String	"transMode"
TRANS_TIME	String	"transTime"
TRANS_TIMEOUT	String	"transTimeout"
TRANS_TYPE	String	"transType"
USE_57_SPECIAL_FILL	String	"use57SpecialFill"
USE_ABECS	String	"useABECS"
USE_APDU_LOG	String	"apduLog"
USE_CARD_READ_SUCCESS	String	"useCardReadSuccess"
USE_CT_RUPAY	String	"useCTRupay"
USE_CTLS_PIN_BYPASS	String	"useCtlsPinBypass"
USE_DELAY_PIN	String	"useDelayPIN"
USE_DSPREAD_ENCRYPT_MAGSTRIPE_CONFIGURE	String	"useDspreadEncryptMagstripeConfigure"
USE_ENCRYPT_AMEX_TRACK	String	"useEncryptAmexTrack"
USE_ENCRYPT_KSN	String	"useEncryptKsn"
USE_ENCRYPT_MAGSTRIPE_CONFIGURE	String	"useEncryptMagstripeConfigure"
USE_FASTBOY	String	"useFastboy"
USE_FILTER	String	"useFilter"
USE_FORCED_AID_SELECTION	String	"useForcedAIDSelection"
USE_FORCED_ICC_AID_SELECTION	String	"useForcedIccAIDSelection"
USE_FORCED_RETURN_OF_CARD	String	"useForcedReturnOfCard"
USE_GAC1_FILTER	String	"useGac1Filter"
USE_GAC2_FILTER	String	"useGac2Filter"
USE_GPO_BEFORE_FILTER	String	"useGpoBeforeFilter"
USE_KERNEL_LOG	String	"kernelLog"
USE_LOG	String	"log"
USE_MAGSTRIPE_FILTER	String	"useMagstripeFilter"
USE_MCCS	String	"useMCCS"
USE_MOSAMBEE_ENCRYPT_MAGSTRIPE_CONFIGURE	String	"useMosambeeEncryptMagstripeConfigure"
USE_PAN_MASK_KEEP_MODE	String	"usePanMaskKeepMode"
USE_PPSE_FAIL_SEND_AIDS_OPTION	String	"usePPSEFailSendAidsOption"
USE_SELECT_AFTER_FILTER	String	"useSelectAfterFilter"
USE_SELECT_KERNEL	String	"useSelectKernel"
USE_SPECIAL_AID_SELECTION	String	"useSpecialAIDSelection"
USE_TERMINATE_ON_CARD_READ_ERROR	String	"terminateOnCardReadError"
USE_TSYS	String	"useTSYS"
USE_USA_VISA	String	"useUSAVisa"

5.17 ConstantPrinter

Constant Name	Type	Value
STATUS_IDLE	int	0
STATUS_NO_PAPER	int	3
STATUS_OVERHEAT	int	2
STATUS_PRINTING	int	1
STATUS_USABLE	int	4

5.18 ConstantPrinter.Align

Enum Constant	Type	Description
CENTER	ENUM	Center alignment
LEFT	ENUM	Left alignment
RIGHT	ENUM	Right alignment

5.19 ConstantPrinter.BarcodeFormat

Enum Constant	TYPE	Description
CODABAR	ENUM	CODABAR 1D format.
CODE_128	ENUM	Code 128 1D format.
CODE_39	ENUM	Code 39 1D format.
CODE_93	ENUM	Code 93 1D format.
DATA_MATRIX	ENUM	Data Matrix 2D barcode format.
EAN_8	ENUM	EAN-8 1D format.
QR_CODE	ENUM	QR Code 2D barcode format.
UPC_E	ENUM	UPC-E 1D format.

5.20 ConstantPrinter.PrintFailurePolicy

Enum Constant	TYPE	Description
ABORT_ALL	ENUM	Aborts all pending print jobs in the queue
IGNORE_AND_CONTINUE	ENUM	Ignores current error and continues next print

5.21 ConstantPrinter.FontSize

Constant Name	Type	Value
LARGE	float	36.0f
NORMAL	float	24.0f
SMALL	float	16.0f

5.22 ConstantSecurity

Constant Name	Type	Value

AUTHENTICATION_BOTH	int	2
AUTHENTICATION_GENERATION	int	0
AUTHENTICATION_VERIFICATION	int	1
DUKPT_KEY_SELECT_DATA_REQUEST	int	1
DUKPT_KEY_SELECT_DATA_RESPONSE	int	2
DUKPT_KEY_SELECT_MAC_REQUEST_OR_RESPONSE	int	0
DUKPT_KEY_SELECT_PIN_ENCRYPTION	int	3
DUKPT_MAC_MODE_CBC	int	2
DUKPT_MAC_MODE_ECB	int	0
DUKPT_MODE_AES_MODE	int	128
ENCRYPTION_ALGORITHM_AES	int	16
ENCRYPTION_ALGORITHM_SM4	int	32
ENCRYPTION_ALGORITHM_TDES	int	0
ENCRYPTION_MECHANISM_DUKPT	int	2
ENCRYPTION_MECHANISM_MK_SK	int	1
KCV_MODE_CHK_0	int	1
KCV_MODE_CHK_EVEN	int	3
KCV_MODE_CHK_ODD	int	2
KCV_MODE_NO_VERIFY	int	0
KEY_ALG_TYPE_2TDEA	int	0
KEY_ALG_TYPE_3TDEA	int	16
KEY_ALG_TYPE_AES_128	int	32
KEY_ALG_TYPE_AES_192	int	48
KEY_ALG_TYPE_AES_256	int	64
KSN_AUTO_INCREASING_BY_DUKPT_TDES_MAC_BOTH_KEY	int	0
KSN_NOT_AUTO_INCREASING_BY_DUKPT_TDES_MAC_BOTH_KEY	int	20
KSN_NOT_AUTO_INCREASING_BY_DUKPT_TDES_MAC_RSP_KEY	int	40
MAC_ALGORITHM_ANSI_X9_19	int	2
MAC_ALGORITHM_ANSI_X9_9	int	3
MAC_ALGORITHM_CBC	int	0
MAC_ALGORITHM_CMAC	int	7
MAC_ALGORITHM_XOR_ECB_MAC	int	1
MAC_MODE_ANSI_X9_19	int	2
MAC_MODE_ANSI_X9_9	int	3
MAC_MODE_CBC	int	0
MAC_MODE_XOR_ECB_MAC	int	1
NOT_SELF_INCREASING	int	0
OPERATION_DIRECTION_DECRYPT	int	0
OPERATION_DIRECTION_ENCRYPT	int	1
OPERATION_MODE_CBC	int	2
OPERATION_MODE_ECB	int	0
PED_CALC_DES_MODE_CBC_DEC	int	2
PED_CALC_DES_MODE_CBC_ENC	int	3
PED_CALC_DES_MODE_ECB_DEC	int	0

PED_CALC_DES_MODE_ECB_ENC	int	1
PED_CALC_DUKPT_MODE_DEC	int	0
PED_CALC_DUKPT_MODE_ENC	int	1
PED_CALC_RSA_MODE_NO_PADDING	int	0
PED_CALC_RSA_MODE_OAEP_PADDING	int	2
PED_CALC_RSA_MODE_PKCS1_PADDING	int	1
PED_PROTECT_KEY_TYPE_DUKPT	int	1
PED_PROTECT_KEY_TYPE_MKSK	int	0
PED_PROTECT_KEY_TYPE_RSA	int	2
PED_PROTECT_TYPE_DEC	int	1
PED_PROTECT_TYPE_TR31	int	0
PED_PROTECT_WRITE_TYPE_DUKPT	int	2
PED_PROTECT_WRITE_TYPE_TLK	int	0
PED_PROTECT_WRITE_TYPE_TMK	int	1
PED_RSA	int	10
PED_TAK	int	4
PED_TDK	int	5
PED_TEK	int	6
PED_TIK	int	7
PED_TLK	int	1
PED_TMK	int	2
PED_TPK	int	3
PED_TTK	int	9
PINBLOCK_DUKPT_FMT_ISO9564_0	int	32
PINBLOCK_DUKPT_FMT_ISO9564_0_KSN_INC	int	0
PINBLOCK_DUKPT_FMT_ISO9564_1	int	33
PINBLOCK_DUKPT_FMT_ISO9564_1_KSN_INC	int	1
PINBLOCK_DUKPT_FMT_ISO9564_2	int	34
PINBLOCK_DUKPT_FMT_ISO9564_2_KSN_INC	int	2
PINBLOCK_DUKPT_FMT_ISO9564_4	int	36
PINBLOCK_DUKPT_FMT_ISO9564_4_KSN_INC	int	4
PINBLOCK_TPK_FMT_ISO9564_0	int	0
PINBLOCK_TPK_FMT_ISO9564_1	int	1
PINBLOCK_TPK_FMT_ISO9564_3	int	2
PINBLOCK_TPK_FMT_ISO9564_4	int	4
SELF_INCREASING	int	64
USE_BOTH_WAYS_KEY	int	2
USE_DATA_DECRYPT_KEY	int	1
USE_DATA_ENCRYPT_KEY	int	0
WRITE_DUKPT_WITH_TMK_ALG_TYPE_AES	int	17
WRITE_DUKPT_WITH_TMK_ALG_TYPE_TDES	int	16
DUKPT_PADDING_MODE_9797_1_M1	int	1
DUKPT_PADDING_MODE_9797_1_M2	int	2
DUKPT_PADDING_MODE_NONE	int	0

DUKPT_PADDING_MODE_PKCS7	int	3

5.23 ConstantScanner

Constant Name	Type
ALL_BARCODES	ConstantScanner.BarcodeFormat[]
ONE_DIMENSIONAL_BARCODES	ConstantScanner.BarcodeFormat[]
TWO_DIMENSIONAL_BARCODES	ConstantScanner.BarcodeFormat[]

5.24 ConstantScanner.BarcodeFormat

Enum Constant	Type	Description
CODABAR	1D	Used in blood banks and libraries, supports digits and some special characters
CODABLOCKF	1D	Stacked barcode format used in logistics and industrial applications
CODE11	1D	Primarily for telecom equipment, supports digits and hyphens
CODE128	1D	High-density linear barcode supporting ASCII, widely used in logistics and retail
CODE39	1D	General-purpose alphanumeric barcode for industrial applications
CODE93	1D	Improved version of Code 39 with higher density character support
EAN13	1D	International standard retail barcode (13-digit)
EAN8	1D	Compact version of EAN-13 for small products (8-digit)
GS1_128	1D	Supply chain barcode based on Code 128 standard
GS1_DATABAR	1D	Compact barcode for small retail items like produce
HK25	1D	Hong Kong variant of Interleaved 2 of 5 barcode
IATA25	1D	Air cargo specific barcode format
INDUSTRIAL25	1D	Industrial variant of 25 barcodes
ITF25	1D	High-density numeric barcode for carton labeling
MATRIX25	1D	Variant of 25 barcode format
MSI	1D	Inventory management barcode (digits only)
TELEPEN	1D	UK barcode standard supporting full ASCII
UPCA	1D	North American retail product barcode (12-digit)
UPCE	1D	Compressed version of UPC-A for small packages
USPS4ST	1D	US Postal Service tracking barcode
AZTEC	2D	Compact matrix barcode suitable for small spaces
DATAMATRIX	2D	2D barcode for product identification and marking
DOTCODE	2D	Dot-based barcode for high-speed industrial printing
GRIDMATRIX	2D	Chinese-developed 2D barcode for Chinese characters
GS1_DATAMATRIX	2D	GS1-compliant version of Data Matrix
HANXIN	2D	Chinese national standard 2D barcode
MAXICODE	2D	Fixed-size matrix barcode used in logistics
MICROPDF	2D	Compact PDF417 variant for ID documents

PDF417	2D	Stacked 2D barcode for transportation and ID cards
QRCODE	2D	Popular matrix barcode for general-purpose use

5.25 ConstantScanner.ScannerCameraType

Enum Constant	TYPE	Description
CAMERA_FRONT	ENUM	Front Camera
CAMERA_REAR	ENUM	Rear Camera
SCANNER	ENUM	E-Series Scanner

5.26 ConstantScanner.BarcodeSupport

Enum Constant	TYPE	Description
ALL_SUPPORT	ENUM	This barcode type is supported by both cameras and scanner
CAMERA_SUPPORT	ENUM	This barcode type is only supported by front/rear cameras
SCANNER_SUPPORT	ENUM	This barcode type is only supported by scanner

5.27 com.kozen.financial.pinpad.PinViewEnum

Enum Constant	TYPE	Description
BUTTON_BACKSPACE	ENUM	Backspace key
BUTTON_ENTER	ENUM	Enter key
BUTTON_ESC	ENUM	Escape key
BUTTON0	ENUM	Number key 0
BUTTON1	ENUM	Number key 1
BUTTON2	ENUM	Number key 2
BUTTON3	ENUM	Number key 3
BUTTON4	ENUM	Number key 4
BUTTON5	ENUM	Number key 5
BUTTON6	ENUM	Number key 6
BUTTON7	ENUM	Number key 7
BUTTON8	ENUM	Number key 8
BUTTON9	ENUM	Number key 9

5.28 ConstantPrinter.GrayPercent

Enum Constant	TYPE	Description
GRAY_100	ENUM	100% gray level (fully opaque gray).
GRAY_70	ENUM	70% gray level.
GRAY_80	ENUM	80% gray level.
GRAY_90	ENUM	90% gray level

5.29 ConstantPrinter.GlobalFontSize

Enum Constant	TYPE	Description
LARGE	ENUM	Large size.
NORMAL	ENUM	Normal size.
SMALL	ENUM	Small size.

5.30 ConstantPrinter.LineSpaceMultiplier

Enum Constant	TYPE	Description
MULTIPLIER_05	ENUM	Multiplier value of 0.5.
MULTIPLIER_10	ENUM	Multiplier value of 1.0.
MULTIPLIER_15	ENUM	Multiplier value of 1.5.
MULTIPLIER_20	ENUM	Multiplier value of 2.0.

5.31 ConstantEcr.ConnectType

Enum Constant	TYPE	Description
BT	ENUM	bluetooth
HOST	ENUM	localhost

5.32 ConstantEcr.ConnectState

Enum Constant	TYPE	Description
CONNECT_TIMEOUT	ENUM	Connection timed out.
CONNECTED	ENUM	Connected.
CONNECTION_ERROR	ENUM	Connection error occurred.
DISCONNECTED	ENUM	Disconnected.
IDLE	ENUM	Initial state.
READ_ERROR	ENUM	Read error occurred.
SERVER_CREATE_ERROR	ENUM	Server creation error occurred.
SERVER_LISTENING	ENUM	Server is listening.
WRITE_ERROR	ENUM	Write error occurred.

5.33 ConstantGeneral.IndicatorType

Enum Constant	TYPE	Description
PINPAD_PHYSICAL	int	Physical pinpad

5.34 ConstantCardReader.DetectMifareMode

Constant Name	TYPE	Value
CARDREADER_DETECT_MODE_M	int	77
CARDREADER_DETECT_MODE_N	int	78
CARDREADER_DETECT_MODE_X	int	88

5.35 ConstantCardReader.MifareBlockOperateType

Constant Name	TYPE	Value
OPERATE_BLK_TYPE_ADD	int	43
OPERATE_BLK_TYPE_SUBSTRACT	int	45
OPERATE_BLK_TYPE_TRANSFER	int	62

5.36 ConstantCardReader.MifareCardAuthType

Constant Name	TYPE	Value
AUTH_KEY_TYPE_AES	int	83
AUTH_KEY_TYPE_CLASSIC_A	int	65
AUTH_KEY_TYPE_CLASSIC_B	int	66
AUTH_KEY_TYPE_TDES	int	84

5.37 ConstantCardReader.DetectCardType

Constant Name	TYPE	Value
CONTACT	int	2
CONTACTLESS	int	4
FELICA	int	32
MAGNETIC	int	1
MIFARE_CLASSIC	int	128
MIFARE_DESFIRE	int	512
MIFARE_PLUS	int	256
MIFARE_ULTRALIGHT	int	8
PSAM	int	16

5.38 ConstantCardReader.DetectContactlessMode

Constant Name	TYPE	Value
CARD_READER_DETECT_MODE_A	String	"A"
CARD_READER_DETECT_MODE_B	String	"B"
CARD_READER_DETECT_MODE_EMV	String	"1"
CARD_READER_DETECT_MODE_ISO14443	String	"0"

5.39 ConstantCardReader.DetectFelicaRequestCode

Constant Name	TYPE	Value
COMMUNICATION_PERFORMANCE_REQUEST	String	"02"
NO_REQUEST	String	"00"
SYSTEM_CODE_REQUEST	String	"01"

5.40 ConstantPinpad.PinpadType

Enum Constant	TYPE	Description
VIRTUAL	ENUM	Virtual PINPAD
PHYSICAL	ENUM	Physical PINPAD

5.41 ConstantPrinter.PaperWidth

Enum Constant	TYPE	Description
PAPER_WIDTH_2INCHES	ENUM	2-inch paper
PAPER_WIDTH_3INCHES	ENUM	3-inch paper

5.42 ConstantSecurity.RSA_BUNDLE_KEY

Constant Name	TYPE	Value
BUNDLE_RSA_EXPONENT	String	"rsaExponent"
BUNDLE_RSA_KEY	String	"rsaKey"
BUNDLE_RSA_MODULE	String	"rsaModule"

5.43 com.kozen.financial.aidl.cardreader.CustomByteArray

Constant Name	Type	Description
buffer	byte[]	Data content
len	int	Data byte length

5.44 com.kozen.financial.aidl.printer.BarcodePrintLine

Constant Name	Type	Description
alignment	ConstantPrinter.Align	Horizontal alignment.
barcodeFormat	ConstantPrinter.BarcodeFormat	Barcode type.
content	String	Barcode content.
height	int	Barcode height.
paperWidth	ConstantPrinter.PaperWidth	Paper width.
width	int	Barcode width.

5.45 com.kozen.financial.aidl.printer.BitmapPrintLine

Constant Name	Type	Description
alignment	ConstantPrinter.Align	Horizontal alignment.
bitmap	android.graphics.Bitmap	Bitmap object to print.
paperWidth	ConstantPrinter.PaperWidth	Paper width.

5.46 com.kozen.financial.aidl.printer.TextPrintLine

Constant Name	Type	Description
alignment	ConstantPrinter.Align	Horizontal alignment.

bold	boolean	Whether bold display is needed.
content	String	Text content to print.
paperWidth	ConstantPrinter.PaperWidth	Paper width.
size	float	Font size of the printed text.

5.47 com.kozen.financial.aidl.printer.BitmapPrintLine

Constant Name	Type	Description
alignment	ConstantPrinter.Align	Horizontal alignment.
bitmap	android.graphics.Bitmap	Bitmap object to print.
paperWidth	ConstantPrinter.PaperWidth	Paper width.

- 6. Access permission

Permission Name	Related Module	Tooltips
android.permission.SUPER_PERMISSIONS_PRINTER	PrinterManager	Control and use built-in printer
android.permission.SUPER_PERMISSIONS_PINPAD	PinpadManager	Access encrypted pinpad for input
android.permission.SUPER_PERMISSIONS_SCANNER	ScannerManager	Scan barcodes and QR codes
android.permission.SUPER_PERMISSIONS_CARD_READER	CardReaderManager	Access magstripe, IC, and contactless cards
android.permission.SUPER_PERMISSIONS_EMV	EmvManager	Perform EMV card transactions
android.permission.SUPER_PERMISSIONS_GENERAL	GeneralManager	Access general device functions (buzzer, LED)
android.permission.SUPER_PERMISSIONS_SECURITY	SecurityManager	Use security module (key management, encryption)
android.permission.SUPER_PERMISSIONS_ECR	EcrManager	Interact with external ECR system