Table of Contents

SDK API

initialize

Initialize the SDK.

```
void initialize(Context context, DeviceListener listener)
```

Parameters

- `context` : application context.
- `listener` : A callback listener that receives callback data from the device.

> Refer to DeviceListener Class

> If this function is not called, the SDK will not work properly.
> It should be called as soon as possible after running the application.

run

Run the SDK service.

```
int run()
```

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

> The SDK was developed for use with a single application.
> It requires caution because it is not used concurrently in other applications.

getVersions

Get the SDK/Firmware versions.

```
int getVersions(Versions versions)
```

Parameters

- `versions` : version information(SDK / Firmware / Fingerprint library / Card library)

> Refer to Version Class

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

getDeviceInfo

Get the device information.

```
int getDeviceInfo(Device device)
```

Parameters

- `device` : device information

> Refer to Device Class

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

setAutoStartApplication

Automatically launch the application when Android starts.

```
int setAutoStartApplication(boolean enable)
```

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

setCardType

Set the RF card type.

```
int setCardType(int cardType)
```

Parameters

- `cardType` : RF Card type

No.	Description
1	HIGH_FREQUENCY
2	HIGH_FREQUENCY_ICLASS
3	LOW_FREQUENCY
4	LOW_FREQUENCY_PROX

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

> It is set up automatically on the device. Used only when card type change is required.

connectFtpServer

connect to the FTP server.

```
int connectFtpServer(FirmwareOption option)
```

Parameters

- `option`: Firmware file option.

> Refer to FirmwareOption Class

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

> To use the firmware upgrade function, you need to build an FTP server.
> The API acts as an FTP client.
>
> To use the API, Android applications require the following permissions:
>
> ```
> <uses-permission android:name="android.permission.INTERNET"/>
> <uses-permission
> android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
> ```

disconnectFtpServer

Disconnects to the FTP server.

```
int disconnectFtpServer()
```

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

> To use the firmware upgrade function, you need to build an FTP server.
> The API acts as an FTP client.
>
> To use the API, Android applications require the following permissions:
>
> ```
> <uses-permission android:name="android.permission.INTERNET"/>
> ```

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

getFtpFileList

Get a list of file names for the FTP server.

```
int getFtpFileList(ArrayList<String> fileList)
```

Parameters

- `fileList`: List of file names

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

To use the firmware upgrade function, you need to build an FTP server.
The API acts as an FTP client.

To use the API, Android applications require the following permissions:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

upgradeFirmware

Copy the upgrade file to the specified firmware file.

```
int upgradeFirmware(FirmwareOption option)
```

Parameters

- `option`: Firmware File Option

> Refer to FirmwareOption Class

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

> To use the firmware upgrade function, you need to build an FTP server.
> The API acts as an FTP client.
>
> To use the API, Android applications require the following permissions:
>
> ```
> <uses-permission android:name="android.permission.INTERNET"/>
> <uses-permission
> android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
> ```

rebootDevice

Reboot the device.

```
int rebootDevice()
```

Return Code

Returns “SUCCESS” if successfully launched; returns the corresponding error code if an error occurs.

> The firmware will be upgraded at device reboot.