

Table of Contents

- SDK API** 1
- initialize** 1
 - Parameters 1
- run** 1
 - Return Code 1
- getVersion** 2
 - Parameters 2
 - Return Code 2
- setCardType** 2
 - Parameters 2
 - Return Code 2
- connectFtpServer** 3
 - Parameters 3
 - Return Code 3
- disconnectFtpServer** 3
 - Return Code 3
- getFirmwareFileList** 4
 - Parameters 4
 - Return Code 4
- upgradeFirmware** 4
 - Parameters 5
 - Return Code 5
- rebootDevice** 5
 - Return Code 5

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.

getVersion

Get the SDK/Firmware version.

```
int getVersion(Version version)
```

Parameters

- version : SDK/Firmware version

[Refer to Version Class](#)

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 : Parameters

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.

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" />
```

getFirmwareFileList

Get a list of file names for the FTP server.

```
int getFirmwareFileList(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.

From:

<http://kb.supremainc.com/svpsdk/> - **SVP Android SDK**

Permanent link:

http://kb.supremainc.com/svpsdk/doku.php?id=en:sdk_api&rev=1545001609

Last update: **2018/12/17 08:06**