API Documentation

(short form description)

Product / Project:	**TELID®300.nfc Library API for Android**
Customer / Project Code:	**-**
Product:	**-**
Product Code:	**-**
Document Revision:	**1.1**
Date:	**2019-07-12**

API - Definition for Android® devices

This document describes the java class library for Android® devices supporting NFC.

Driver name :	de.microsensys.telid300interface
Tested devices:	Samsung Galaxy S6, Huawei Mate 9
Supported host interface:	Smartphone NFC interface
Revision:	----
Package:	de.microsensys.telid300interface
Min Android SDK:	19 (Android 4.4 = KitKat)
Class:	TELIDLoggerHandler

API Documentation

(short form description)

1. Short history

This chapter includes a short history of modifications.

Date	Reason	Modification	Release Date / Version	FileName
2019-03	- base document creation	Preliminary definition for TELID39x.l.nfc	-	
2019-07	- update for Release 1.1	Check after final implementation of TELID391.l.nfc	2019-07 / 1.1	

API Documentation

(short form description)

2. Content

3. Constructors and initialization functions

public TELIDLoggerHandler(TelidCommunicationCallback _callback)

Constructor used to initialize a new instance of the class. The parameter should be an instance of the "TelidLoggerCallback" interface which will receive updates of the opening process with defined functions.

public boolean checkStartupIntent(Intent _intent)

(Helpful only if NDEF is defined in Intent-Filter for starting the App) Function passes the Intent that started the App. The contents of the Intent will be checked. In case the NDEF-Message of a TELID is contained, will return "true" and automatically start reading the contents. Otherwise this function returns "false" (see SampleCode).

public void enableTelidSearch(NfcAdapter _adapter, Activity _activity)

Function enables the search for a TELID using NFC interface. It is recommended to call this function in "onResume" or "onPostResume" functions. Parameters "_adapter" and "_activity" are needed to enable the search:

- _adapter → Instance of "NfcAdapter". Can be obtained with "NfcAdapter.getDefaultAdapter(Context)" from an Activity.
- _activity → Instance of Activity that starts the search.

(For further details, please check SampleCode)

public void disableTelidSearch(NfcAdapter _adapter, Activity _activity)

Function disables the search. It is recommended to call this function in "onPause" function. Parameters "_adapter" and "_activity" are needed to enable the search:

- _adapter → Instance of "NfcAdapter". Can be obtained with "NfcAdapter.getDefaultAdapter(Context)" from an Activity.
- _activity → Instance of Activity that starts the search.

(For further details, please check SampleCode)

public void startReadProtocol(TELIDInformation _telid, TELIDStateInformation _telidStatus)

Function starts the read protocol process. This function may be called after "TELIDLoggerCallback.telidStatusRead()" has been called and all the needed information is available. Parameters "_telid" and "_telidStatus" are needed to enable the search:

- _telid → Instance of "TELIDInformation". Obtained from TELIDLoggerCallback.
- _telidStatus → Instance of "TELIDStateInformation". Obtained from TELIDLoggerCallback.

public void startProgram(TELIDInformation _telid, TELIDStateInformation _telidStatus, TELIDProgramParameters _programParams)

Function starts the read protocol process. This function may be called after "TELIDLoggerCallback.telidStatusRead()" has been called and all the needed information is available. Parameters "_telid" and "_telidStatus" are needed to enable the search:

- _telid → Instance of "TELIDInformation". Obtained from TELIDLoggerCallback.
- _telidStatus → Instance of "TELIDStateInformation". Obtained from TELIDLoggerCallback.
- _programParams → Instance of "TELIDProgramParameters". (See class description below)

4. Helper classes and interfaces

TELIDLoggerCallback

This interface represents the functions that will be called while the TELID parameters are being read:

- void telidLost(): Called when the NFC communication with TELID is lost
- void telidFound(TELIDInformation): Called when a TELID is first found
- void telidRemarksRead(TELIDInformation, String): Called when the Remarks are read
- void telidStatusRead(TELIDInformation, TELIDStateInformation): Called when the TELID status is read
- void teildReadingMeasurements(int _numberRead, int _totalNumber): Called while the logged measurements are being read. This helps calculate the read progress
- void telidReadCompleted(TELIDInformation, TELIDDataProtocol[]): Called when the read process has finished

TELIDInformation

This class contains information of the TELID data logger:

- ID number → String getTelidSerialNumber()
- Type information → String getTelidTypeString()
- Physical data → int[] getTelidPhysicalData()

TELIDStateInformation

This class contains state information of the TELID data logger:

- Battery state → int GetBatteryState() (not implemented, returns always 0)
- Logging mode → boolean getStatus_isLogging()
- Start time → Calendar getProgrammedStartTime()
- Measurement interval → int getIntervalInSeconds()
- Number of measurements → getNumberLoggedMeasurements()
- Max Number of measurements → getMaxnumberMeasurements()
- Stop time → Calendar getExpectedStopTime()
- Limit min → float getLimitMin()
- Limit max → float getLimitMax()

TELIDDataProtocol

This class contains the measurement protocol of the TELID data logger:

- Date Time → Calendar getTimestamp() // String getTimestampString()
- Type →
- SensorValues → TELIDDataProtocol.SensorValue[] getSensorValues()
- Battery state → int getBatteryState() (not implemented)
- State → int getState()

TELIDProgramParameters

This class is used to program the TELID with new parameters:

- TELIDProgramParameters Constructor. This constructor is used to start a new measurement. Has following parameters:
  - Calendar _start → Time of start of logging
  - Calendar _stop → Time of end of logging
  - int _intervalSeconds → Time interval between measurements, in seconds
  - int _limitMin → Minimum limit
  - int _limitMax → Maximum limit
  - double _shockThreshold → Shock event threshold
  - int _selectedSizes → Selected physical sizes (Mask)
  - String _remarks → Remarks to write into memory
- TELIDProgramParameters getStopParametersInstance() → Use this function to obtain parameters to stop current measurement

LibraryVersion

This class contains Version information of the Library:

- String getVersionNumber() returns a String representing the current Library version

Physical data identifiers

For "TELIDInformation.getTelidPhysicalData()" and "TELIDDataProtocol.SensorValue.getType()", an int value is returned to represent the physical data type. This int can have the following values:

- Temperature → 1
- Shock → 2
- Humidity → 3
- Pressure → 4
- UV Index → 10
- Light → 11

5. Activity recommended implementation

Variables:

NfcAdapterm, TELIDLoggerHandler

onCreate(Bundle savedInstanceState)

Here initialize NfcAdapter and TELIDLoggerHandler variables. For NfcAdapter use "NfcAdapter.getDefaultAdapter(Context)"

onResume()

Check if NFC is enabled (can also be done in "onCreate") and enable search using "TELIDLoggerHandler.enableTelidSearch(NfcAdapter, Activity)"

onPause()

Disable search using "TELIDLoggerHandler.disableTelidSearch(NfcAdapter, Activity)"

This document is subject to change without notice.