EJEMPLO NFC.ANDROID

# NFCAndroidExample

NFC Card Reader Example Using Xamarin.Android

# Xamarin - Near Field Communication

This technology was officially accepted in 2003 by the International Standards Organization. NFC is based on standards. These standards are created by the NFC forum. NFC connects and exchange data based on existing Radio Frequency Identification (RFID) standards. NFC operates at a short distance because of security concerns. And also that is why NFC is widely chosen as the technology for contactless payments.

# There are three types of NFC modes:

# Card emulation:

Physical cards are being replaced by NFC-enabled devices. For instance, instead of a card for opening your hotel room, you could use your smartphone for opening the door.

# Read and write:

An enabled NFC device can exchange data by using tags and smart posters. A tag can hold a small amount of data that can be read by a device. An enabled NFC device can also write data on a tag (Basically they are Target and Initiator). Basically, smart posters can hold multiple tags. Smart posters are widely used for marketing/advertisement purposes.

# Peer to peer communication:

Two enabled NFC devices can communicate with each other. Think of two smartphones that are exchanging data.

# Xamarin and NFC

The Android.Nfc namespace provides functionality to work with NFC. Also there are different tag technologies. The NFC forum standardized five different tags, type 1, 2, 3, 4, 5. These tags are the most common ones and are being supported by most of the enabled NFC devices. There are also alternatives like MIFARE classic.

# NFC Data Exchange format

The NDEF format is standardized by the NFC forum. This message format can be used to read or write from a tag or when working with two active enabled NFC devices. The message contains multiple records:

# Header:

The header record contains important information about the message. One of those is the Type Name Format (TNF). This field indicates the type of data in the payload (the actually transferred data). These are the possible values of the TNF field:

0 -> Empty

1 -> Well-known (text, uri, etc)

2 -> Multipurpose Internet Mail Extension (MIME)

3 -> Absolute Uniform Resource Identifier (URI)

4 -> External

5- > Unknown

6 -> Unchanged (when data in the payload field is too large, the data is chunked in multiple records)

7 -> Reserved (not used currently)

Other information stored in the header record are MB (message begin), ME (message end), CF, SR, and IL.

# Type length:

The length of the payload type.

# Payload length:

The length of the payload field, again in the payload field the actual data is stored.

# ID length:

Length of the ID field.

# Record type:

The type of the payload data. This value corresponds to the TNF field in the header.

# Record ID:

ID of the record, mostly used for external applications to identify the message.

# Payload:

Contains the actual data stored in bytes.

# Android.Nfc

The namespace Android.Nfc contains all classes you need to work with NFC functionality.

# NfcAdapter:

This class represents the device's NFC adapter. This is the main starting point for developing with NFC. It contains all operations you need for discovering tags. This class also contains some constants which I'll explain later.

# NdefMessage:

This class represent the NFC Data Exchange format message what I explained in the previous chapter.

# NdefRecord:

As I explained in the previous section a message can have multiple records.

What I mention earlier is that there different tag types available. Xamarin supports a number of these type and can be found under the namespace Android.Nfc.Tech.

# Ndef :

Provides access to NDEF content and operations on a Tag.

# Note:

When you're building an App you need to explicit configure that your app wants to use the device's NFC adapter. You can do this by selecting the required permissions in the Android Manifest.

The NfcAdapter class contains three action constants we can use for discovering tags. Meaning our app will do something when the smartphone scan a tag.

# NfcAdapter.ActionNdefDiscovered:

A tag with an NDEF payload is discovered.

# NfcAdapter.ActionTagDiscovered:

A tag is discovered.

# NfcAdapter.ActionTechDiscovered:

A tag is discovered and activities are registered for the specific technologies on the tag.