Nut Android SDK Document

Revision：

Time Version mark

|  |  |  |
| --- | --- | --- |
| 2016/12/29 | 1.0 | Init |
| 2020/08/19 | 1.1 | Support Beacon |

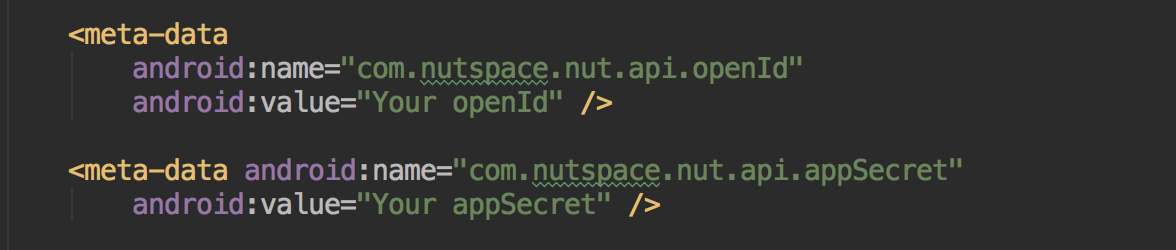
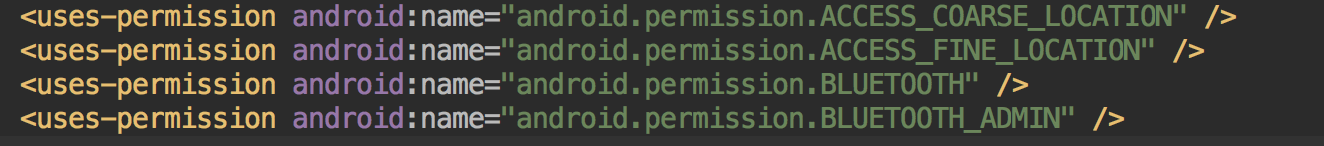
#### Development guide

1. Development environment

Recommend Android Studio 2.0 or up；Android OS 4.3 or up。

1. Configuration manifest and input openId and appSecret

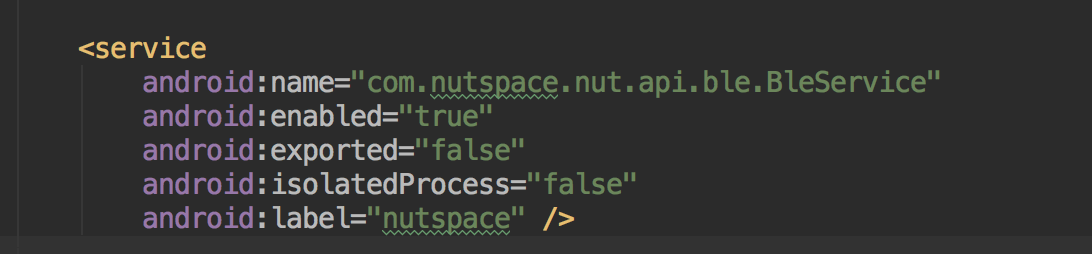
1、Manifest configuration includes the addition of permissions, the following permissions are indispensable, fill openId and appSecret, the code example is as follows:



Permission description：

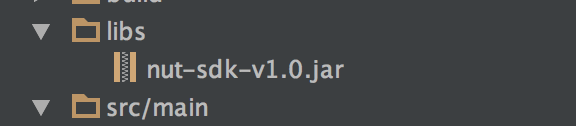
|  |  |
| --- | --- |
| Permission | Use |
| ACCESS\_COARSE\_LOCATION | Scan Nut Device |
| ACCESS\_FINE\_LOCATION | Scan Nut Device |
| BLUETOOTH | BLE |
| BLUETOOTH\_ADMIN | BLE |

2、Configuration BleService

In the manifest definition BleService, the code example is as follows:

3、Import and use the SDK

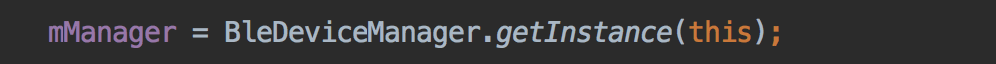
3.1、Import the SDK jar package into the project, for example:



3.2、Initialize BleManager in the component (Activity or Service) and bind BleService.

The code example is as follows:

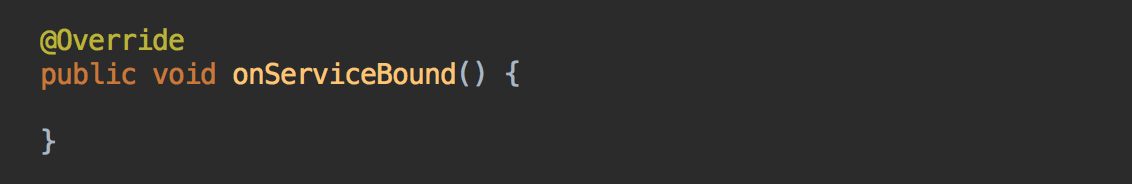
Initialization BleManager：



Bind BleService：



Note: The bind bind () and unbind () methods must appear in pairs

After the binding is successful callback method:

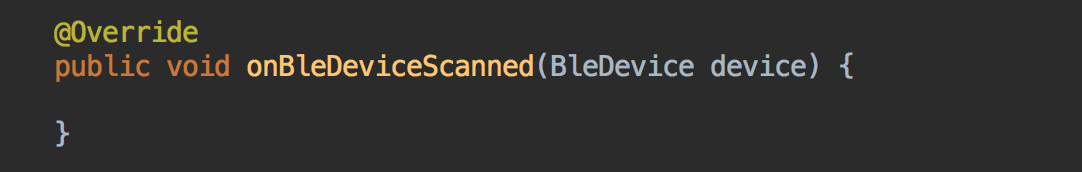
After the binding is successful, you can use the SDK by calling the object mManager.

3.3、Function used to scan as an example:

Register the scan callback interface ScanResultCallback、EventCallback、EventCallbac,

the code example is as follows: 

Note: You can call the removeScanResultCallback () method to remove a listening event when you do not need to listen for a scan event

Scan to device after the callback method: 

And returns the object of class BleDevice that contains the device information.

4. Beacon configuration function (Device need customized firmware to support this API)

//Set the UUID information of Beacon, the string format of UUID needs to adopt standard format, for example: "10102233-4455-6677-8899-AABBCCDDEEFF"

void setBeaconUUID(BleDevice device, String beaconUUID)

//Setting UUID information is complete, result=true means success, false means failure

void onBeaconUUID(BleDevice device, String uuid, boolean result)

//Set the Major and Minor information of Beacon

void setBeaconMajorMinor(BleDevice device, int major, int minor)

//Set Major and Minor information completed, result=true means success, false means failure

void onBeaconMajorMinor(BleDevice device, int major, int minor, boolean result)

Note: Each time you update the Beacon information, you need to call setBeaconUUID first, and then call setBeaconMajorMinor after completion. Each time you update the Beacon information, you need to update the UUID and MajorMinor. Updating only one of them will cause the Beacon information to be abnormal.

5、Interface introduction：

|  |  |
| --- | --- |
| Interface | Features |
| ScanResultCallback | Monitor the results of the scanning device |
| EventCallback | Monitor rssi value changes, call device, device key count, battery changes |
| ConnectStateChangedCallback | Listen to device connection status |
| BeaconResultCallback | Listen to device set beacon configuration |

The code snippet in the document is available in the SDK Demo