**1、Scan for Bluetooth use SWDeviceScanManager class**

**1.1、**Use SWDeviceScanManager class to scan BLE Bluetooth devices, get SWDeviceScanManager instance of an object.

scanManager = **new** SWDeviceScanManager(**this**)

**1.2、**the callback set after scanning Bluetooth scanManager.setScanCallBack(**new** DeviceScanCallBack｛

@Override

**public** **void** onLeScan(BluetoothDevice device, **int** rssi, **byte**[] scanRecord) {

**final** BluetoothDevice device2 = device;

runOnUiThread(**new** Runnable() {

@Override

**public** **void** run() {

mLeDeviceListAdapter.addDevice(device2);

mLeDeviceListAdapter.notifyDataSetChanged();

}

});

}

｝);

**1.3、**start scanning bluetooth devices，scanManager.startScan(); stop scanning using scanManager.stopScan();

**2、connecting Bluetooth or mobile devices and Bluetooth communication using SWDevice class**

**2.1、**Get SWDevice instance of an object，swDevice = **new** SWDevice([**Context**](eclipse-javadoc:â=MyTag/D:/Android%20Project/MyTag/libs/MyTagSDK.jar%3ccom.comime.swdevice(SWDevice.classâSWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;âContext) context, [**String**](eclipse-javadoc:â=MyTag/D:/Android%20Project/MyTag/libs/MyTagSDK.jar%3ccom.comime.swdevice(SWDevice.classâSWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;âString) tag,[**BluetoothDevice**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82BluetoothDevice) device, [**DeviceListener**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82DeviceListener)deviceListener);**for example**: swDevice = **new** SWDevice(DeviceScanActivity.**this**, device.getAddress(),

device, DeviceScanActivity.**this**);

Or using swDevice = **new** SWDevice([**Context**](eclipse-javadoc:â=MyTag/D:/Android%20Project/MyTag/libs/MyTagSDK.jar%3ccom.comime.swdevice(SWDevice.classâSWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;âContext) context, [**String**](eclipse-javadoc:â=MyTag/D:/Android%20Project/MyTag/libs/MyTagSDK.jar%3ccom.comime.swdevice(SWDevice.classâSWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;âString) tag);

**Then** swDevice.setBluetoothDevice([**BluetoothDevice**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82BluetoothDevice) bluetoothDevice);

swDevice.setDeviceListener([**DeviceListener**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82DeviceListener)deviceListener);

**2.2、** Connect bluetooth use swDevice.connectGatt();To disconnect and reconnect use swDevice.connect();

**2.3** By implementing [DeviceListener](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82DeviceListener) interface，you can get some of the callback methods Bluetooth communications operations，such as :

(1)public void onConnected(String tag, BluetoothDevice device)，Bluetooth device successfully connected calls this method;

(2)public void onDisconnected (String tag, BluetoothDevice device)，Bluetotoh devices connect fail or disconnect will call this method;

(3)after use swDevice.readRssi()method; will execute public void onGetRssi(String tag, int rssi, BluetoothDevice device)，We can get rssi;

(4)public void onGetValue(String tag, byte[] value, BluetoothDevice device),Bluetooth will call the callback method when sending data to the APP，we can get the corresponding data through value;

(5)public void onWriteSuccess(String tag, byte[] value, BluetoothDevice device)，APP write data to bluetotoh devices, if write sucessfully, it will call the callback method, we can see what data is written successfully through value;

**2.4、**Use swDevice.startRing() to send an alarm command to bluetooth device；Use swDevice.stopRing() to send a command to stop the alarm；Use swDevice.closeBluetoothDevice() can turn off the Bluetooth device;

**3. If still have what not understand, you can view our example(MyTagexample)**