


DOCUMENTATION

- Overview
- Guides
- UI Guide
- Reference
- Samples
- Design & Quality

 Filter

- Enhance your apps with 5G
- Build client-server applications with gRPC
- Transferring data without draining the battery
- Reduce network battery drain
- Transfer data using Sync Adapters
- Bluetooth
- ▾ Bluetooth Low Energy

Overview

Find BLE devices

Connect to a GATT server

Transfer BLE data
- NFC
- Telecom
- Wi-Fi
- USB
- UWB
- VPN
- Session initiation protocol overview
- Open Mobile App development

Android Developers > Docs > Guides

War das hilfreich?  

Find BLE devices

 ▾

To find BLE devices, you use the `startScan()` method. This method takes a `ScanCallback` as a parameter. You must implement this callback, because that is how scan results are returned. Because scanning is battery-intensive, you should observe the following guidelines:

- As soon as you find the desired device, stop scanning.
- Never scan on a loop, and always set a time limit on your scan. A device that was previously available may have moved out of range, and continuing to scan drains the battery.

In the following example, the BLE app provides an activity (`DeviceScanActivity`) to scan for available Bluetooth LE devices and display them in a list to the user. The following snippet shows how to start and stop a scan:

KotlinJava

```
private BluetoothLeScanner bluetoothLeScanner = bluetoothAdapter.getBluetoothLeScanner();
private boolean scanning;
private Handler handler = new Handler();

// Stops scanning after 10 seconds.
private static final long SCAN_PERIOD = 10000;

private void scanLeDevice() {
    if (!scanning) {
        // Stops scanning after a predefined scan period.
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                scanning = false;
                bluetoothLeScanner.stopScan(leScanCallback);
            }
        }, SCAN_PERIOD);

        scanning = true;
        bluetoothLeScanner.startScan(leScanCallback);
    } else {
        scanning = false;
        bluetoothLeScanner.stopScan(leScanCallback);
    }
}
```

★ **Note:** The `BluetoothLeScanner` is only available from the `BluetoothAdapter` if Bluetooth is currently enabled on the device. If Bluetooth is not enabled, then `getBluetoothLeScanner()` returns null.

To scan for only specific types of peripherals, you can instead call `startScan(List<ScanFilter>, ScanSettings, ScanCallback)` , providing a list of `ScanFilter` objects that restrict the devices that the scan looks for and a `ScanSettings` object that specifies parameters about the scan.

The following code sample is an implementation of `ScanCallback` , which is the interface used to deliver BLE scan results. When results are found, they are added to a list adapter in the `DeviceScanActivity` to display to the user.

KotlinJava

```
private LeDeviceListAdapter leDeviceListAdapter = new LeDeviceListAdapter();

// Device scan callback.
private ScanCallback leScanCallback =
    new ScanCallback() {
        @Override
        public void onScanResult(int callbackType, ScanResult result) {
            super.onScanResult(callbackType, result);
            leDeviceListAdapter.addDevice(result.getDevice());
            leDeviceListAdapter.notifyDataSetChanged();
        }
    };
};
```


★ **Note:** You can only scan for Bluetooth LE devices or scan for classic Bluetooth devices, as described in [Bluetooth overview](#). You cannot scan for both Bluetooth LE and classic devices at the same time.

War das hilfreich?




Content and code samples on this page are subject to the licenses described in the [Content License](#). Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates.

Last updated 2021-10-27 UTC.



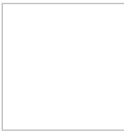
Twitter

Follow @AndroidDev on Twitter



YouTube

Check out Android Developers on YouTube



LinkedIn

Connect with the Android Developers community on LinkedIn

MORE ANDROID	DISCOVER	ANDROID DEVICES	RELEASES	DOCUMENTATI... AND DOWNLOADS	SUPPORT
Android	Gaming	Large screens	Android 11		Report platform bug
Android for Enterprise	Machine Learning	Wear OS	Android 10	Android Studio guide	Report documentation bug
Security	Privacy	Android TV	Pie	Developers guides	Google Play support
Source	5G	Android for cars	Oreo	API reference	Join research studies
News		Android Things	Nougat	Download Studio	
Blog		Chrome OS devices	Marshmallow	Android NDK	
Podcasts			Lollipop		
			KitKat		

[Google Developers](#)

- Android
- Chrome
- Firebase
- Google Cloud Platform
- All products