

Package name: android.bluetooth.*, 主要相关类介绍如下:

- * BluetoothAdapter: 本地蓝牙设备的适配类, 所有的蓝牙操作都要通过该类完成;
- * BluetoothDevice: 蓝牙设备类, 代表了蓝牙通讯过程中的远端设备;
- * BluetoothSocket: 蓝牙通讯套接字, 代表了与远端设备的连接点, 使用socket本地程序可以通过inputstream和outputstream与远端程序进行通讯;
- * BluetoothServerSocket: 服务器通讯套接字, 与TCP ServerSocket类似;
- * BluetoothClass: 用于描述远端设备的类型, 特点等信息, 通过getBluetoothClass()方法获取代表远端设备属性的BluetoothClass对象。

使用蓝牙必须获取的权限:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

建立蓝牙连接:

通过BluetoothAdapter.getDefaultAdapter()方法获取BluetoothAdapter对象。

判断当前蓝牙是否启动, 如果没有启动提示用户手动启动:

```
if (!mBluetoothAdapter.isEnabled()) {  
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);  
}
```

在Activity的onActivityResult()方法中, 对用户的设定结果进行处理。

搜寻远端蓝牙设备

首先获取已配对的远端设备：

```
Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
```

然后通过BluetoothAdapter.startDiscovery()方法启动蓝牙设备的搜寻。这是个异步方法，调用的时候立刻就会返回。为了获得搜寻的结果，必须在用户自己的Activity中注册一个BroadcastReceiver，代码如下：

```
// Create a BroadcastReceiver for ACTION_FOUND
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        // When discovery finds a device
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            // Get the BluetoothDevice object from the Intent
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            // Add the name and address to an array adapter to show in a ListView
            mArrayAdapter.add(device.getName() + "\n" + device.getAddress());
        }
    }
};

// Register the BroadcastReceiver
IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
registerReceiver(mReceiver, filter); // Don't forget to unregister during onDestroy
```

设置本地设备可以被发现

只有将本地设备设置为可被发现，远端的蓝牙设备才能够找到并和本地设备建立连接。通过下面的代码发送

Intent对象，让用户手动启动可发现设置。

```
Intent discoverableIntent = new  
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);  
discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300);  
startActivity(discoverableIntent);
```

在本地设备可发现的过程中，可以通过注册BroadcastReceiver监听可发现状态的改变。

连接蓝牙设备

要想让两台设备上自己的应用程序进行交互，必须在程序中同时实现服务器端和客户端机制。