Committed : 18/02/2020
Comment : 2 files in “services” package are modified (GATTBroadcastReceiver.java & ServiceBLE.java)

We are using a **BroadcastReceiver** to get broadcast intents. The **onReceive** method is called when the BroadcastReceiver is receiving an Intent broadcast. (L’idée est exactement pareille que le listener du bouton que tu m’avais demandé mais ce dernier est pour le broadcast.)

1. In GATTBroadcastReceiver.java: I added some codes (highlighted in yellow) to read the data.

```java
@Override
public void onReceive(Context context, Intent intent) {
    final String action = intent.getAction();

    if (Service_GATT.ACTION_GATT_CONNECTED.equals(action)) {
        connected = true;
    } else if (Service_GATT.ACTION_GATT_DISCONNECTED.equals(action)) {
        connected = false;
        Utils.toast(activity.getApplicationContext(), "Disconnected From Device");
        activity.finish();
    } else if (Service_GATT.ACTION_GATT_SERVICES_DISCOVERED.equals(action)) {
        ServiceBLE.updateServices();
    } else if (Service_GATT.ACTION_DATA_AVAILABLE.equals(action)) {
        String uuid = intent.getStringExtra(Service_GATT.EXTRA_UUID);
        String data = intent.getStringExtra(Service_GATT.EXTRA_DATA);

        //ADD YOUR CODES HERE (IT'S LIKE A LISTENER)
        //OR
        //YOU WRITE YOUR CODE IN -> ServiceBLE.updateCharacteristic() AND CALL IT HERE

        ServiceBLE.updateCharacteristic(uuid, data);
    }
    return;
}
```

2. In ServiceBLE.java: I added the updateCharacteristic() method, you can add your code here if you choose not to do it in the onReceive method (In GATTBroadcastReceiver.java). This method can be removed if you choose to write your code directly in OnReceive, but it is preferable to implement this method.

```java
public static void updateCharacteristic(String uuid, String data) {
  //YOUR CODE HERE (Use the data which is User Id to get their business card from databases etc.)
 //listAdapter.notifyDataSetChanged(); (This method is useful if you are using a list adapter for showing something)
//In ServiceBLE.java, the attribute "ble" is your activity, you can use it to make changes on the UI eg: ble.getCard();
}
```

Commits: https://github.com/LeeJoeIng/PACT_BLE/commits?author=LeeJoeIng

Git : https://github.com/LeeJoeIng/PACT_BLE

To do: Test and add features

Known issue: Deprecated API

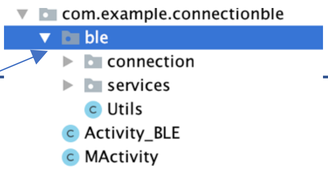
Solution for some problems: Clean project. Rebuild project. Delete app.iml file if necessary

1. In manifest file: Declare the Bluetooth.

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

<uses-feature
    android:name="android.hardware.bluetooth_le"
    android:required="false" />
```

2. Import the “connect” package into the project.



3. Update the app/src/main/res/values/strings.xml
4. At main activity: Add two constants, a button (to initiate pairing onClick), 2 private objects (initialize in OnCreate function).

```
public static final int REQUEST_ENABLE_BT = 1;
public static final int BLE_SERVICES = 2;
private ConnectionBLE connectionBLE;
private BLEStateUpdateReceiver bleStateUpdateReceiver;
private Button button;

(Inside OnCreate function)
connectionBLE = new ConnectionBLE(this,button);
bleStateUpdateReceiver = new BLEStateUpdateReceiver(this);
button = findViewById(R.id.button);
```

5. At main activity in OnCreate function: Add `PackageManager.hasSystemFeature()` to determine BLE availability.

```
if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH_LE)) {
    Toast.makeText(getApplicationContext(), R.string.ble_not_supported, Toast.LENGTH_SHORT).show();
    finish();
}
```

6. At main activity: Initiate pairing (onClick)

```
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        connectionBLE.pairing();
    }
});
```

7. At main activity: override these 6 methods:

```
@Override
protected void onStart() {
    super.onStart();
    registerReceiver(bleStateUpdateReceiver, new IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED));
}
@Override
protected void onResume() {
    super.onResume();
}
@Override
protected void onPause() {
    super.onPause();
    connectionBLE.stopScan();
}
@Override
protected void onStop() {
    super.onStop();
    unregisterReceiver(bleStateUpdateReceiver);
    connectionBLE.stopScan();
}
@Override
public void onDestroy() {
    super.onDestroy();
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    // Check which request we're responding to
    if (requestCode == REQUEST_ENABLE_BT) {
        // Make sure the request was successful
        if (resultCode == RESULT_OK) {
        } else if (resultCode == RESULT_CANCELED) {
            Utils.toast(getApplicationContext(), getString(R.string.info_ble));
        }
    } else if (requestCode == BLE_SERVICES) {
        Utils.toast(getApplicationContext(), getString(R.string.pairing));
    }
}
```

8.Create or refactor an activity (interface) called `Activity_BLE` that allows user to get ID from hardware (a button or something else).

9.At Activity_BLE: Add these attributes.

```java
public static final String EXTRA_ADDRESS = "com.example.ble.Activity_Services.ADDRESS";
public static final String EXTRA_NAME = "com.example.connectionble.Activity_BLE.NAME";
private ServiceBLE serviceBLE;
private GATTBroadcastReceiver gattUpdateReceiver;
private Button button2;
private Intent ble_Service_Intent;
```

10. At Activity_BLE in OnCreate function: Add these.

```java
button2 = findViewById(R.id.button2);

Intent intent = getIntent();
String name = intent.getStringExtra(Activity_BLE.EXTRA_NAME);
String address = intent.getStringExtra(Activity_BLE.EXTRA_ADDRESS);
serviceBLE = new ServiceBLE(this, name, address);

button2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        serviceBLE.write();
        //serviceBLE.infoHardware();
        //todo add second button to read
    }
});
```

11. Add these two elements in manifest file

```xml
<receiver
    android:name=".ble.services.GATTBroadcastReceiver"
    android:enabled="true"
    android:exported="true"></receiver>

<service
    android:name=".ble.services.Service_GATT"
    android:enabled="true"
    android:exported="true" />
```

12. At Activity_BLE: Override these functions.

```
@Override
protected void onStart() {
    super.onStart();

    gattUpdateReceiver = new GATTBroadcastReceiver(this);
    registerReceiver(gattUpdateReceiver, Utils.makeGattUpdateIntentFilter());

    ble_Service_Intent = new Intent(this, Service_GATT.class);
    bindService(ble_Service_Intent, serviceBLE.getBle_ServiceConnection(), Context.BIND_AUTO_CREATE);
    startService(ble_Service_Intent);
}

@Override
protected void onResume() {
    super.onResume();
}

@Override
protected void onPause() {
    super.onPause();
}

@Override
protected void onStop() {
    super.onStop();

    unregisterReceiver(gattUpdateReceiver);
    unbindService(serviceBLE.getBle_ServiceConnection());
    ble_Service_Intent = null;
}
```