Device Pairing UI BizBundle

Version: 20241218

Contents

Device Pairing UI BizBundle is used to implement pairing guidance and smart device activation. It applies to multiple common types of smart devices, such as Wi-Fi devices, Zigbee devices, Bluetooth devices, and devices that support QR code scanning like GPRS & NB-smart devices. The UI BizBundle provides the service logic and UI encapsulation to guide pairing and activate smart devices over different protocols.

1. Device types

1.1. Wi-Fi device pairing

Wi-Fi smart devices can be paired and connected to cloud services in two modes: Wi-Fi Easy Connect (EZ) mode and access point (AP) or hotspot mode. IP cameras (IPCs) can be paired after QR code scanning.

Term	Description
Wi-Fi EZ mode	Also known as the EZ mode. This pairing mode is implemented through the following steps: • The app encapsulates the pairing data into the specified section of an IEEE 802.11 packet and sends the packet to the network. • The Wi-Fi module of a smart device runs in promiscuous mode and listens for and captures all packets over the network. • The Wi-Fi mode parses the packets that carry the pairing data from the app into the data format specified by the protocol.
AP mode	Also known as the hotspot mode. A mobile phone acts as a station (STA) and connects to the hotspot of a smart device. Then, both devices are paired to establish a socket connection between them and exchange data through the specified ports.
IPC pairing in QR code mode	An IPC scans the QR code on the app to get the pairing data.

1.2. Zigbee device pairing

Zigbee gateways and sub-devices can be paired.

Term	Description
Zigbee gateway	The device that integrates the coordinator with Wi-Fi features on a Zigbee network. The gateway is responsible to formulate the Zigbee network and store data.
Zigbee sub-device	A router or an end device on a Zigbee network. Each Zigbee sub-device is responsible to forward data, or respond to control commands.

1.3. Bluetooth device pairing

Tuya provides the following Bluetooth solutions.

Term	Description
Bluetooth Low Energy (LE)	A point-to-point connection is created between a Bluetooth or Bluetooth LE device and a mobile phone.
Bluetooth mesh	Enable many-to-many (m:m) device communications over a mesh network released by Bluetooth Special Interest Group (SIG). This solution is also known as Bluetooth LE mesh.
Tuya mesh	Tuya's proprietary technology that enables Bluetooth devices to communicate over a mesh network.
Combo devices	The devices that support Bluetooth and Wi-Fi combo can be paired over **either Bluetooth or Wi-Fi**.

1.4. QR code pairing

A smart device that supports this pairing mode is connected to Tuya's cloud services immediately after power on. The app can be used to scan the QR code on the device to implement cloud-based device activation and binding. The QR code must comply with Tuya's QR code rules. For more information, contact Tuya's account manager.

Term	Description
GPRS devices	Smart devices that apply GPRS technologies to connect to the network and access cloud services.
NB-smart devices	Smart devices that apply NB-IoT technologies.

1.5. Matter device pairing

If you want to use the Matter device pairing feature, see Prepare for Integration with Matter Device for initialization.

1.6. Pairing by auto discovery

Tuya provides efficient pairing features on top of Tuya's generic pairing technologies for smart devices.

2. Integrate with the UI BizBundle

1. Create a project.

Integrate Smart Life App SDK for Android into your project with Android Studio

and add the framework of the UI BizBundle to your project .

2. Configure the `build.gradle` file.

```
1  dependencies {
       api enforcedPlatform("com.thingclips.smart:thingsmart-BizBundlesBom:\
2  ${ #link("https://developer.tuya.com/en/docs/app-development/integration?id=Ka8
   qhzk13vkfq#title-6-Dependency%20sources")[biz_bom_version] }")
3      api "com.thingclips.smart:thingsmart-bizbundle-device_activator"
       \/\/ To implement the QR code scanning feature, depend on the BizBundle for
4
   QR code scanning.
5      api "com.thingclips.smart:thingsmart-bizbundle-qrcode_mlkit"
       api "com.thingclips.smart:thingsmart:\${ #link("https://developer.tuya.com/
6  en/docs/app-development/integration?id=Ka8qhzk13vkfq#title-6-Dependency%20
   sources")[sdk_version] }}"
7  }
```

3. Obfuscate the code.

```
1  #fastJson
2  -keep class com.alibaba.fastjson.**{*;}
3  -dontwarn com.alibaba.fastjson.**
4
5  #rx
6  -dontwarn rx.**
7  -keep class rx.** {*;}
8  -keep class io.reactivex.**{*;}
9  -dontwarn io.reactivex.**
10 -keep class rx.**{ *; }
11 -keep class rx.android.**{*;}
12
13 #fresco
14 -keep class com.facebook.drawee.backends.pipeline.Fresco
15 -keep @com.facebook.common.internal.DoNotStrip class *
16 -keepclassmembers class * {
17 @com.facebook.common.internal.DoNotStrip *;
18 }
19
20 #tuya
21 -keep class com.thingclips.**{*;}
22 -dontwarn com.thingclips.**
```

3. Configure features

Configuration items	Field name	Description	Default value
Hotspot name settings	`<string name="ap_mode_ssid">SmartLife</string>`	The prefix of a device hotspot for pairing.	`SmartLife`
Support for Bluetooth devices	`<bool name="is_need_ble_support">true</bool>`	Specifies whether Bluetooth devices can be paired.	`true`
Support for Bluetooth mesh devices	`<bool name="is_need_blemesh_support">true</bool>`	Specifies whether Bluetooth mesh devices can be paired.	`true`
Support for QR code scanning	`<bool name="is_scan_support">true</bool>`	Specifies whether the app can be used to scan the QR code on a device and start pairing.	`true`
Whether the Wi-Fi EZ mode is supported by automatic discovery	`<bool name="is_config_activator_ez">false</bool>`	Specifies whether smart devices that support the EZ mode can be automatically discovered.	`false`

3.1. Configure auto discovery

Create a JSON file named `activator_auto_search_capacity.json` in the `assets` directory. Add the following code to this file.

```
1  {
2      //Specifies whether to support searching for sub-devices.
3      "searchTypeZigbee": false,
```

```
 4        //Specifies whether to support searching for Wi-Fi EZ devices.
 5      "searchTypeEZ": true,
 6        //Specifies whether to support searching for Pegasus devices.
 7      "searchTypePegasus": false,
 8        //Specifies whether to support searching for Bluetooth devices.
 9      "searchTypeBle": false,
10        //Specifies whether to support searching for wired devices.
11      "searchTypeWired": false,
12        //Specifies whether to support searching for Matter devices.
13      "searchTypeMatter": false
14    }
```

4. Make API requests

4.1. Open the pairing page

```
1   ThingDeviceActivatorManager.INSTANCE.startDeviceActiveAction(this);
2
3   ThingDeviceActivatorManager.INSTANCE.addListener(new
    IThingDeviceActiveListener() {
4
5           @Override
6           public void onDevicesAdd(List<String> list) {
7
8           }
9
10          @Override
11          public void onRoomDataUpdate() {
12
13          }
14
15          @Override
16          public void onOpenDevicePanel(String s) {
17
18          }
19
20          @Override
21          public void onExitConfigBiz() {
22
23          }
24      });
```

Request parameters

Parameter	Description
activity	The pairing activity.
homeId	The home ID. For more information, see Home Information Management .

Response parameters

Parameter	Description
devIds	The success callback. A list of paired device IDs is returned.
updateRoomData	Indicates whether room device details are updated.
onOpenDevicePanel	Indicates whether a certain panel can be opened. You can implement this feature as needed.

Parameter	Description
onExitConfigBiz	Indicates whether to exit the pairing service if pairing is not performed. You can implement this feature as needed.

4.2. Open the QR code scanning page

```
1   ScanManager.INSTANCE.openScan(this);
```