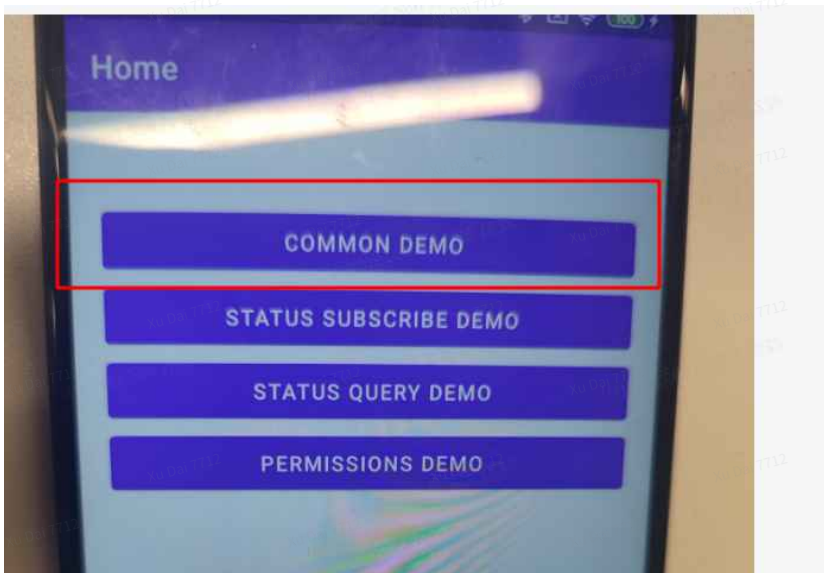
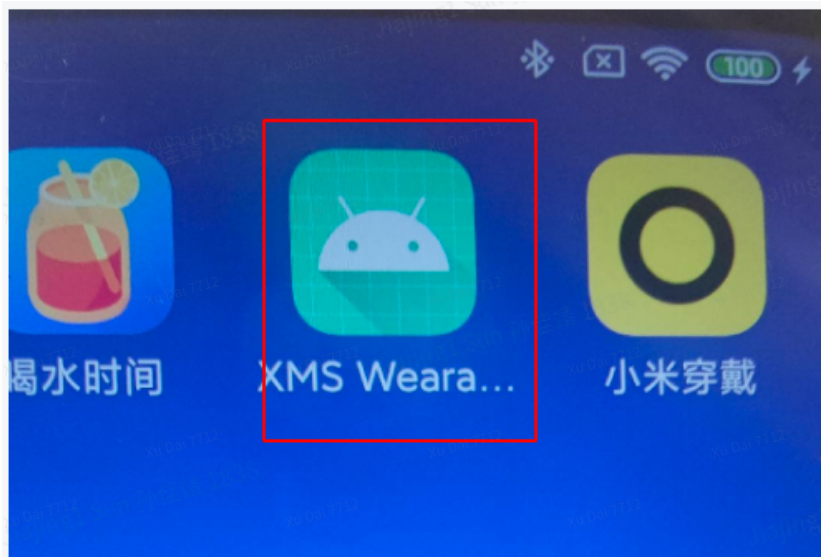


interconnect测试

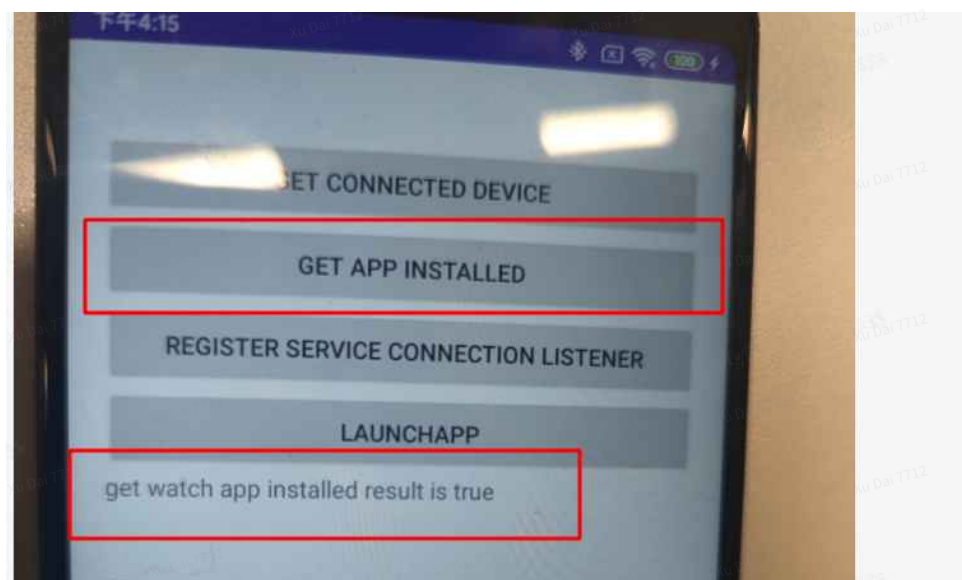
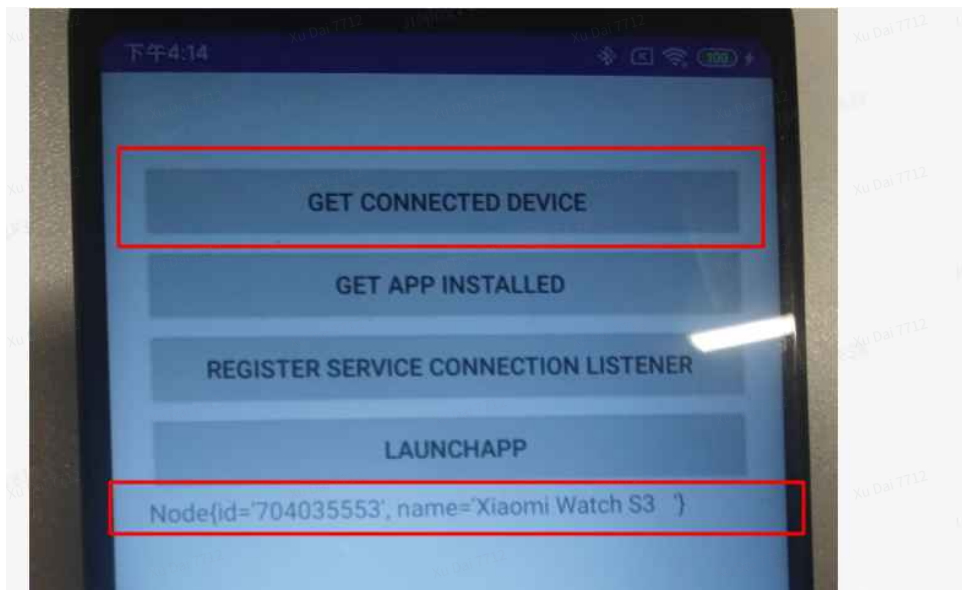
1. 通过小米运动健康安装快应用 com.xiaomi.xms.wearable.demo.debug.1.0.0.rpk 到手表上。
2. 手机端安装安卓应用 app-debug.apk

准备工作

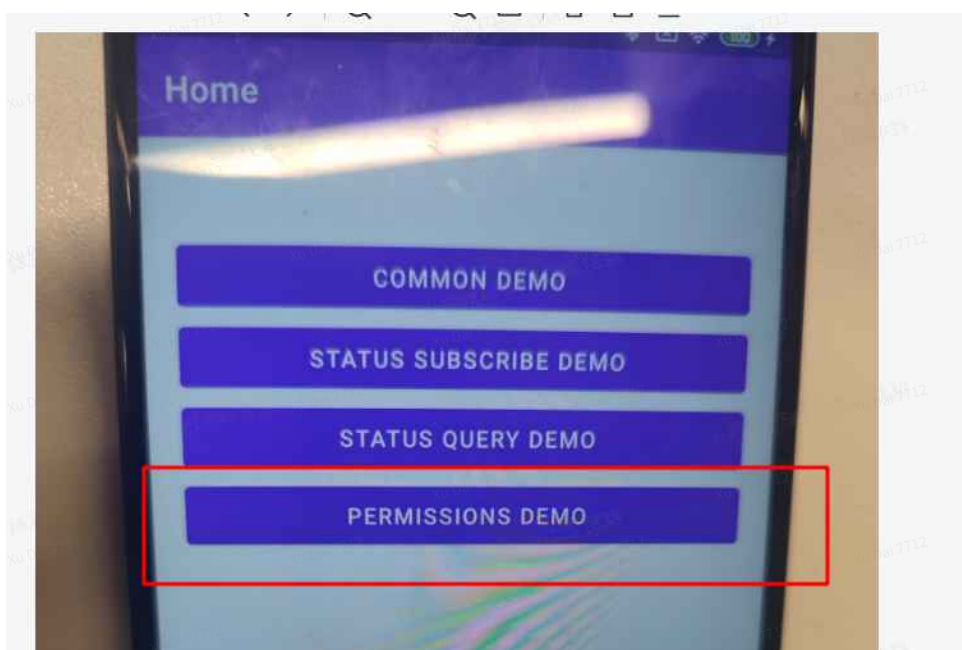
1. 打开安卓应用，进入后先点击COMMON DEMO



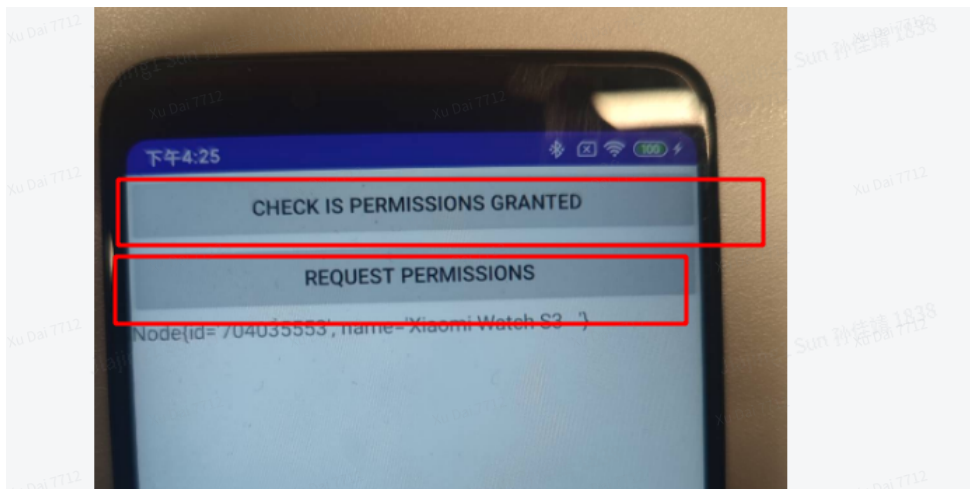
2. 进入后分别点击GET CONNECTED DEVICE 和 GET APP INSTALLED查看信息是否正确
(如果提示错误，先执行第3步，再回头执行第2步试试)
(正确情况：见下图)



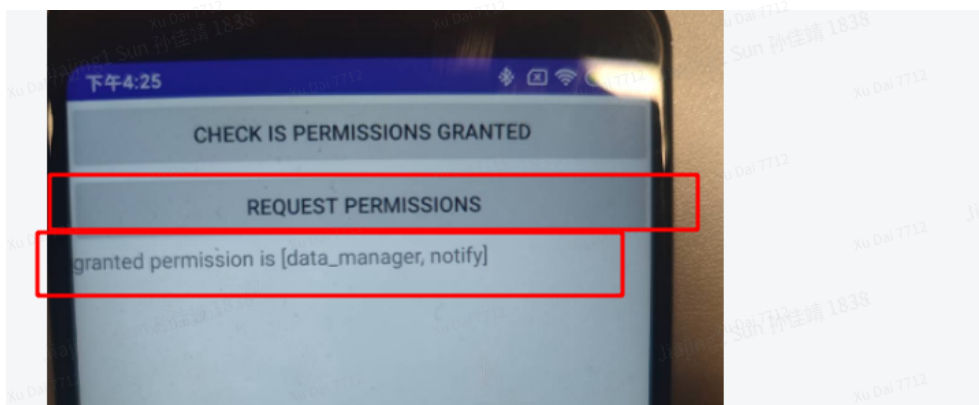
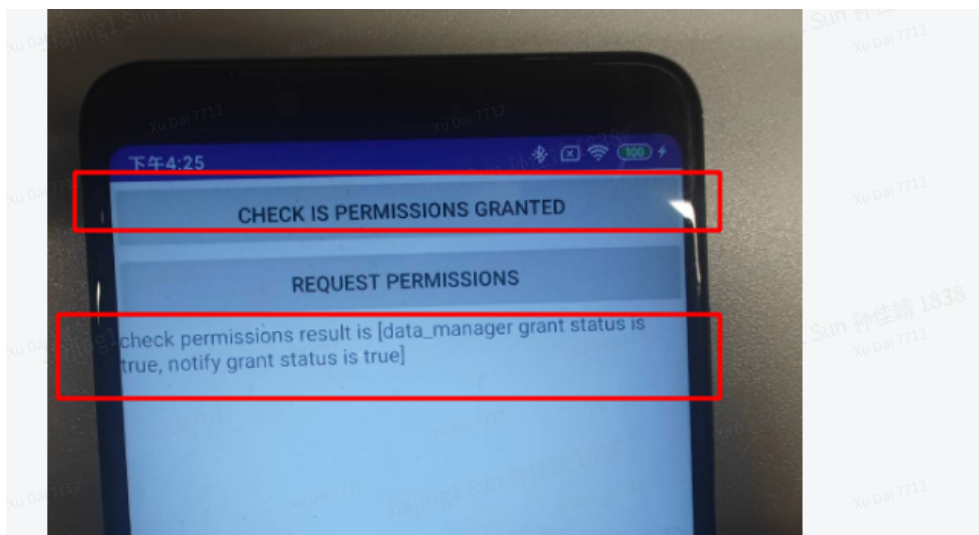
3. 退出到上一页，然后点击PERMISSIONS DEMO，进行权限申请



4. 进入后依次点击按钮CHECK IS PERMISSION GRANTED 和 REQUEST PERMISSION即可

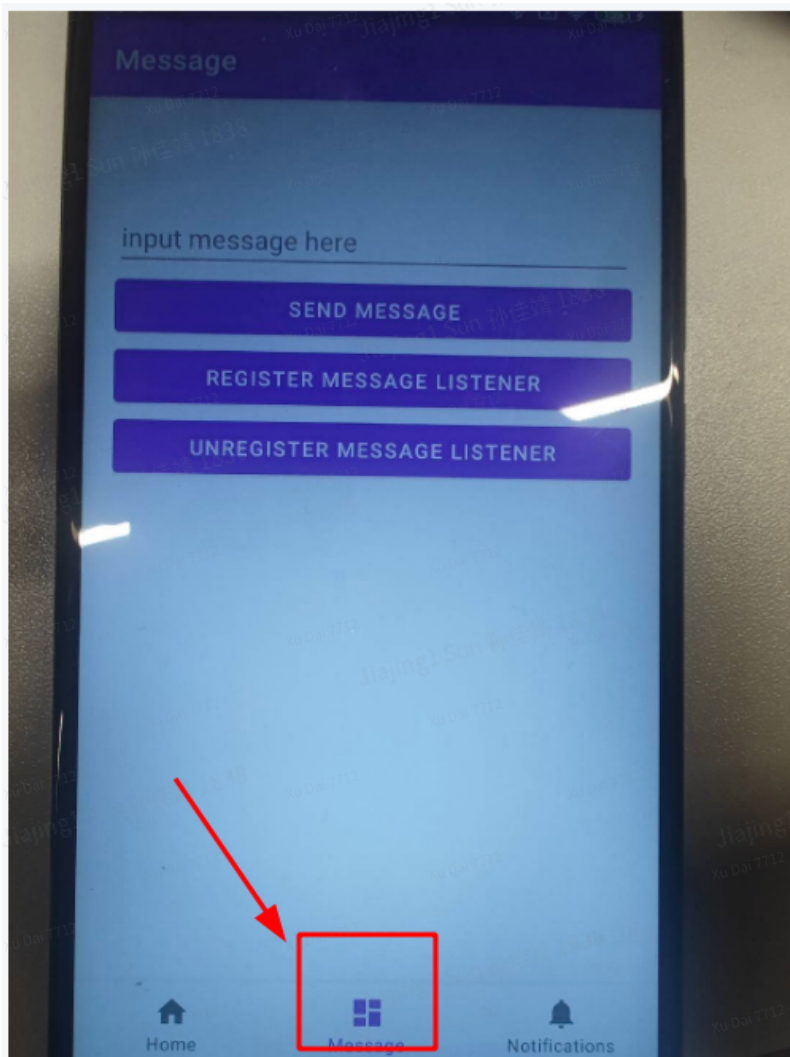


正确提示如下:

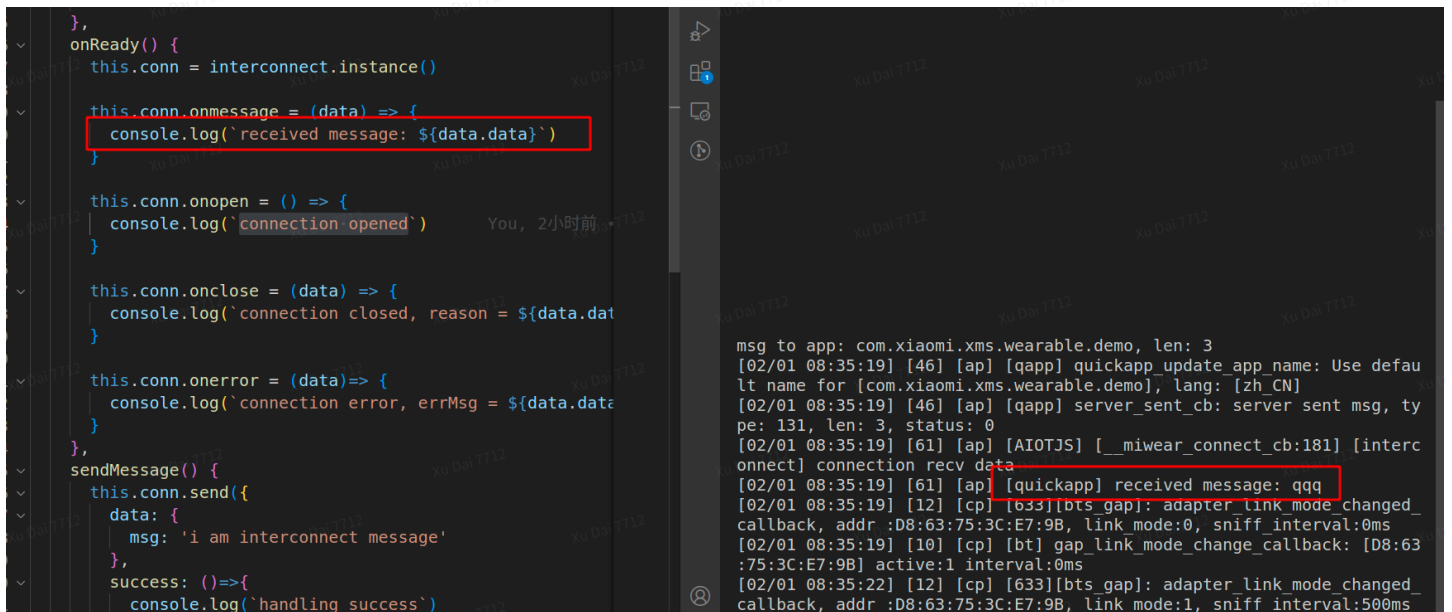


onmessage

1. 点击下方message按钮

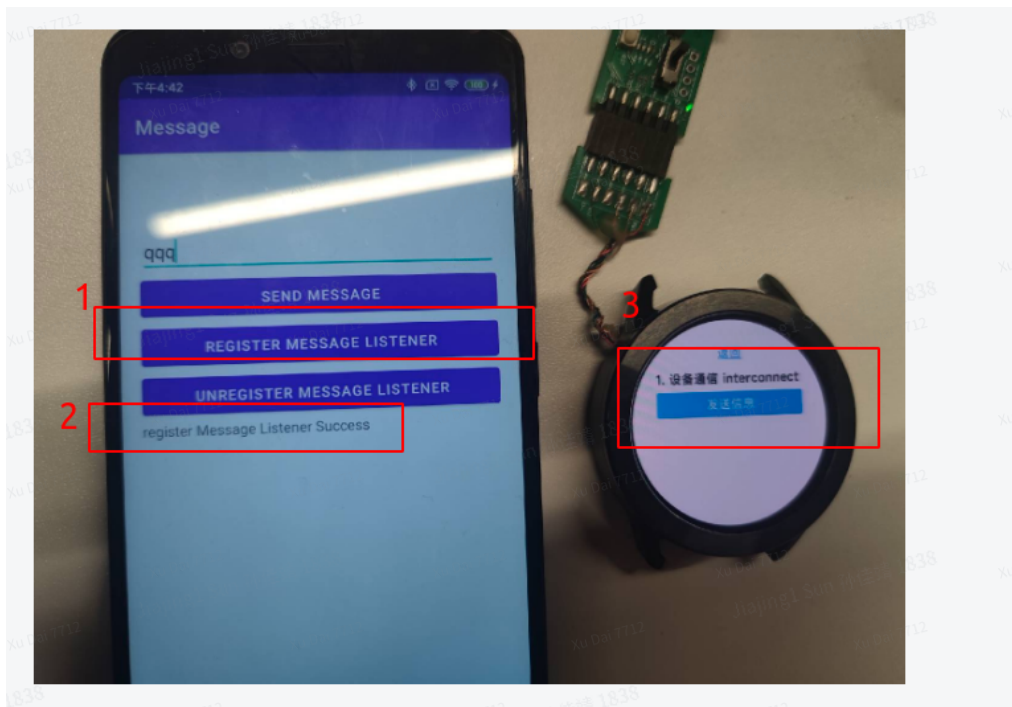


2. 在输入框内输入内容，然后点击SEND MESSAGE，然后看手表测是否可以正常收到log



send

1. 点击REGISTER MESSAGE LISTENER，下方提示成功
2. 点击手表上的发送信息



结果验证

可以看log，有对应的length信息，查看是否跟你传递的信息length一致

```
[02/01 08:46:22] [46] [ap] [qapp] quickapp_update_app_name: Use default name for [com.xiaomi.xms.wearable.demo], lang: [zh_CN]
[02/01 08:46:22] [61] [ap] [AIOTJS] [__miwear_send:302] [interconnect] apk_status send data, len: 22
[02/01 08:46:22] [61] [ap] [AIOTJS] [__send_cb:233] [interconnect] send callback status:0
[02/01 08:46:22] [61] [ap] [quickapp] handling success
```

onopen

因为testcase interconnect 的 onReady生命周期里面注册了onopen回调，所以打开这个页面观察log即可验证（安卓应用也要确保开启状态）

```
</template>
<script>
import router from '@system.router'
import interconnect from '@system.interconnect'
export default {
  private: {
    conn: null
  },
  onReady() {
    this.conn = interconnect.instance()

    this.conn.onmessage = (data) => {
      console.log('received message: ${data.data}')
    }

    this.conn.onopen = () => {
      console.log('connection opened')
    }

    this.conn.onclose = (data) => {

```

```
[02/01 08:52:09] [40] [ap] [qapp] got_phone_app_status: app: [com.xiaomi.xms.wearable.demo]-[connected]
[02/01 08:52:09] [46] [ap] [qapp] server_sent_cb: server sent message: 2, len: 8, status: 0
[02/01 08:52:09] [67] [ap] [AIOTJS] [__miwear_connect_cb:137] [interconnect] msg type:5
[02/01 08:52:09] [67] [ap] [AIOTJS] [__miwear_connect_cb:155] [interconnect] connected!
[02/01 08:52:09] [67] [ap] [quickapp] connection opened
[02/01 08:52:09] [10] [cp] [I] MASS DL: suk mass pb_handle
[02/01 08:52:09] [10] [cp] [I] MASS DL: cmd is 1, data type is 6
[02/01 08:52:09] [10] [cp] [I] MASS DL: ##### MassControl_Op_P
[02/01 08:52:09] [10] [cp] [bt] mass_finish_handle: ## mass_finish_handle, --b
[02/01 08:52:09] [10] [cp] [bt] mass_finish_handle: ## mass_index, filename is /data/mass/tmp/e6fda7391216a73596ff821c00fb4b78, size = 47
[02/01 08:52:09] [10] [cp] [bt] mass_finish_handle: ## mass_finish_handle, errcode is [pause]
[02/01 08:52:09] [10] [cp] [bt] miwear_mass_control_handle: finish result is pause
[02/01 08:52:09] [10] [cp] [bt] mass_data_info_init: ## mass_data_info_init, mass index is 0
[02/01 08:52:11] [12] [cp] [633][bts_gap]: adapter link mode change callback, addr: 08:63:75:3C:E7:9B, link mode: 1, sniff interval: 5
```

onclose

手机端退出应用，看log即可验证

```

this.conn.onclose = (data) => {
  console.log('connection closed, reason = ${data.data}')
}

this.conn.onerror = (data) => {
  console.log('connection error, errMsg = ${data.data}')
},
sendMessage() {
  this.conn.send({
    data: {
      [02/01 08:55:22] [46] [ap] [qapp] got_phone_app_status: app: [com.xiaomi.xms.wearable.demo]-[disconnected]
      [02/01 08:55:22] [46] [ap] [qapp] server_sent_cb: server sent msg, type: 2, len: 8, status: 0
      [02/01 08:55:22] [89] [ap] [AIOTJS] [__miwear_connect_cb:137] [interconnect] msg type:6
      [02/01 08:55:22] [89] [ap] [AIOTJS] [__miwear_connect_cb:163] [interconnect]-disconnected!
      [02/01 08:55:22] [89] [ap] [quickapp] connection closed, reason = connection closed, code = undefined
      [02/01 08:55:22] [12] [cp] [633][bts_gap]: adapter_link_mode_changed_callback, addr :D8:63:75:3C:E7:9B, link mode:0, sniff_interval:0ms
      [02/01 08:55:22] [10] [cp] [bt] gap_link_mode_change_callback: [D8:63:75:3C:E7:9B] active:1 interval:0ms
    }
  })
}
```