

设备控制用例

设备控制前置工作

AI需要获取当前设备列表并存储到短/长期记忆如会话上下文，粒度至少细到可以区分房间、设备名称、设备类型

(进一步的) 获取当前设备物模型关键属性状态,以支持AI根据当前设备的开关、属性判断当前设备状态
示例，可以返回用户房间内的所有设备列表，可以带出当前状态

代码块

```
1  [
2    {
3      "user_id": 10020,
4      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D",
5      "name": "演示调光灯",
6      "room": "卧室",
7      "device_type": "57D56F4D-3302-41F7-AB34-5365AA180E81",
8      "attribute": "{\n\"brightness\":30,\n\"hue\":170,\n\"isOn\":true,\n\"saturation\":100,\n\"type\":\n\"彩色灯\"}",
9      "tsl_source": "HomeKit"
10   },
11   {
12     "user_id": 10020,
13     "sn": "7EF787F0-DED1-58AD-9876-27C2CB27E237",
14     "name": "演示窗帘",
15     "room": "卧室",
16     "device_type": "2FB9EE1F-1C21-4D0B-9383-9B65F64DBF0E",
17     "attribute": "{\n\"currentPosition\":0,\n\"isOn\":false,\n\"state\":\n\"已停止\",
18     \"targetPosition\":0,\n\"type\":\n\"智能窗帘\"}",
19     "tsl_source": "HomeKit"
20   }
21 ]
```

用例

ON/OFF设备

1. 智能五路调光灯

关灯场景

用户语音交互：Rei，关上灯

结果：AI找到并关上用户短期记忆位置的灯

说明：

前置LLM输出指令给服务端，下面两种做法都可以，目前是第一种做法

第一种做法，下发完整预期的值。字段从设备列表接口中获取，这种做法可以由AI简单的对设备属性进行替换，AI负担可能小一些

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", //智能灯的SN
3      "isOn": false,
4      "brightness": 100,
5      "hue": 50, //调光值, 比如暖黄光
6      "saturation": 70,
7      "actionType": "DEVICE_CONTROL", // 设备控制Action
8      "type": "彩色灯"
9  }
```

另一种做法，只下发预期改变的属性字段，这种需要AI提取需要控制的属性，然后组装出Action

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", // 智能灯的SN
3      "actionType": "DEVICE_CONTROL", // 设备控制Action
4      "action": {
5          "isOn": false // 关灯
6      }
7  }
```

阅读场景

用户语音交互：Rei，我想看看书，帮我换个舒服的光线

结果：AI找到用户常待房间的灯，设置灯光亮度80%，饱和度70%，Hue调光50%

- 色调 (Hue): 50°
- 饱和度 (Saturation): 70%
- 亮度 (Brightness): 80%

说明：

前置LLM输出指令给服务端

下发完整预期的值。字段从设备列表接口中获取

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", //智能灯的SN
3      "brightness":80,
4      "hue":50,      //调光值
5      "isOn":true,
6      "saturation":70,
7      "actionType": "DEVICE_CONTROL", // 设备控制Action
8      "deviceType":"彩色灯"
9  }
```

派对场景

用户语音交互：Rei，气氛热闹点

结果：灯闪烁100次，紫色

- 色调 (Hue): 270°
- 饱和度 (Saturation): 100%
- 亮度 (Brightness): 80%

说明

前置LLM输出指令给服务端

下发完整预期的值。字段从设备列表接口中获取

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", //智能灯的SN
3      "brightness":80%, // 亮度80%
4      "saturation":100,
5      "hue": 270,
6      "interval": 200, // 间隔200ms
7      "actionType": "LIGHT_BLINKING", // 灯光闪烁Action, 用于支持场景化的控制,减少AI重
      复控制时间和Token成本
8      "times": "100", // 闪烁次数100次
9      "deviceType":"彩色灯"
10 }
```

非onoff设备

1. 窗帘

睡眠场景

用户语音交互：Rei, 窗外太亮了

结果：拉上一定比例的窗帘，50%半开

前置LLM输出指令给服务端

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", //智能灯的SN
3      "targetPosition":50,
4      "actionType": "DEVICE_CONTROL", // 设备控制Action
5      "deviceType":"智能窗帘"
6  }
```

传感器

1. 人体存在传感器

询问是否有人场景

用户语音交互：Rei，客卫有人吗

结果：返回有/无人

说明：可调用设备查询接口，通过同步接口查询，也可以通过设备列表接口一次性读取

示例回复

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", 人体存在传感器的SN
3      "room": "客卫",
4      "isOccupied": false,
5  }
```

复合场景

睡眠场景

用户语音交互：Rei, 我要睡觉了

结果：关灯并且关窗帘

前置LLM输出指令给服务端，下发两条指令，可以是数组，也可以分开

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", // 智能灯的SN
3      "isOn": false,
4      "brightness": 100,
5      "hue": 50, // 调光值, 比如暖黄光
6      "saturation": 70,
7      "actionType": "DEVICE_CONTROL", // 设备控制Action
8      "type": "彩色灯"
9  }
```

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", // 智能灯的SN
3      "targetPosition": 100, // 拉上窗帘
4      "deviceType": "智能窗帘"
5  }
```

起床场景

用户语音交互：Rei, 我起床了

结果：开灯并且开窗帘

前置LLM输出指令给服务端，下发两条指令，可以是数组，也可以分开

代码块

```
1  {
2      "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", // 智能灯的SN
3      "isOn": true,
4      "brightness": 100,
5      "hue": 50, // 调光值, 比如暖黄光
6      "saturation": 70,
7      "actionType": "DEVICE_CONTROL", // 设备控制Action
8      "type": "彩色灯"
9  }
```

```
1  {  
2    "sn": "8CD5C5AD-CB2C-5ECB-9C73-9EF3AEF69A1D", //智能灯的SN  
3    "targetPosition":0, // 拉开窗帘  
4    "deviceType":"智能窗帘"  
5  }
```