

Alexa 开发配置简要说明文档

V1.0

2020/12/15

目录

1、 前期准备:	3
1.1、 注册亚马逊开发人员账号.....	3
1.2、 设备创建.....	3
1.3、 配置 Echo 音箱.....	4
1.3.1、 添加 Echo 音箱.....	5
1.3.2、 Skill 使能.....	5
2、 Alexa 设备端接口配置.....	6
3、 接口说明.....	7
3.1、 payload 数据回调接口.....	7
3.2、 directive 解码接口.....	8
3.2.1、 ble_alex_decodeDirective.....	8
4、 Skills 说明.....	8
4.1、 新增 skills 流程.....	8
4.2、 Skill 新增对话配置.....	9
4.3、 python code 编写.....	12
5、 参考文档.....	12

1、前期准备:

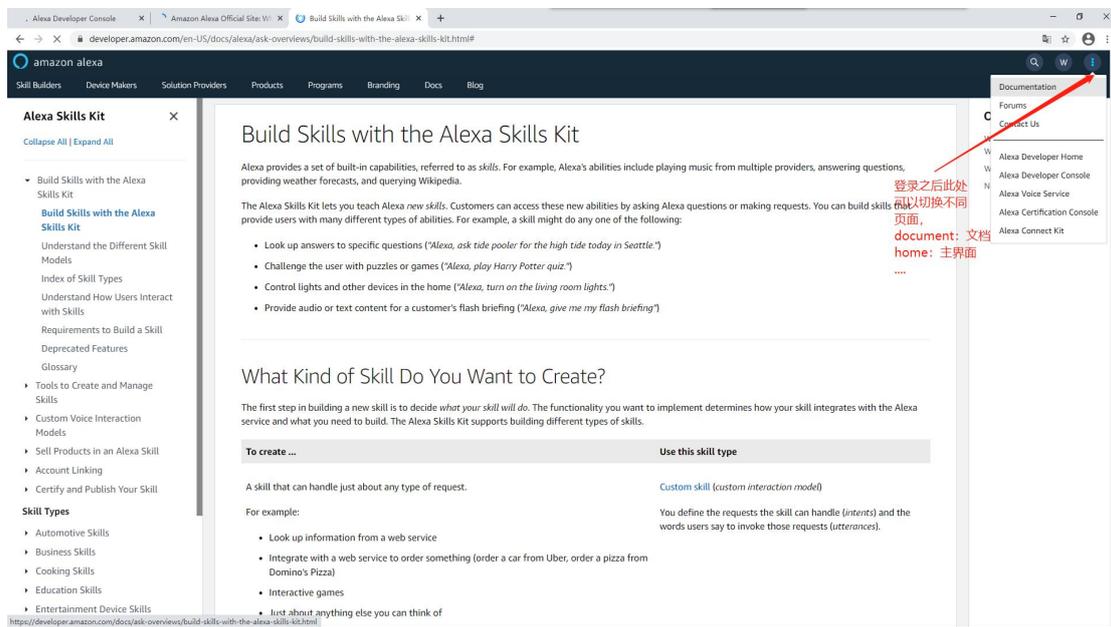
1.1、注册亚马逊开发人员账号

注意，Echo 需要在 Skills 所在的开发人员账号下，才能通过 Echo 语音控制实现 Skills 功能。

做 Alexa 平台开发，先下载 Google 浏览器，别的浏览器可能无法很好的与 Echo 通讯开发。
链接:

https://www.amazon.com/ap/register?clientContext=131-8058625-3501838&showRememberMe=true&openid.pape.max_auth_age=7200&openid.identity=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&marketPlaceId=ATVPDKIKX0DER&pageId=amzn_dante_us&openid.pape.preferred_auth_policies=Singlefactor&openid.return_to=https%3A%2F%2Fdeveloper.amazon.com%2Falexa%2Fconsole%2Fask&prevRID=PGMQEVMGHDS4734FEWP0&openid.assoc_handle=amzn_dante_us&openid.mode=checkid_setup&prepopulatedLoginId=&failedSignInCount=0&openid.claimed_id=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0

登录之后可以在右上角切换不同的页面信息，如下图所示:



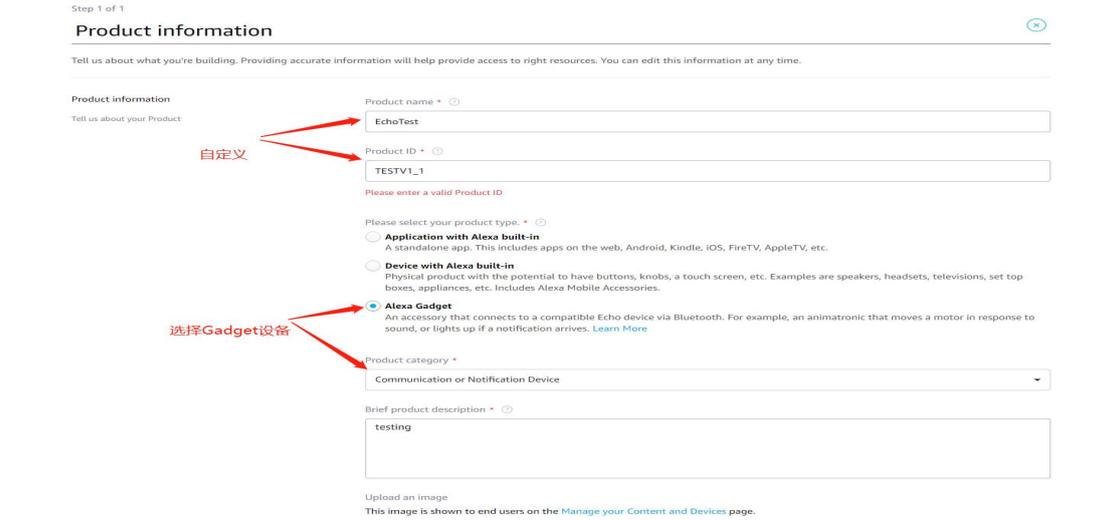
1.2、设备创建

创建设备的流程可以选择第一个界面 document 里面查看文档。

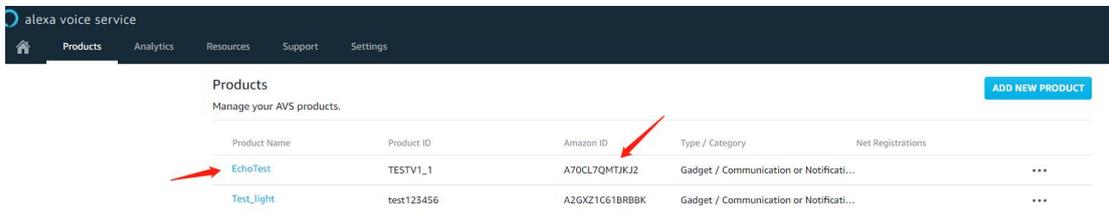
设备管理界面可以切换到 Alexa Voice Service 界面查看。



设备管理界面点击 Products-ADD NEW PRODUCT，创建一个新的产品。



后面的选择 NO 即可，点击 FINISH，即创建成功，如下：



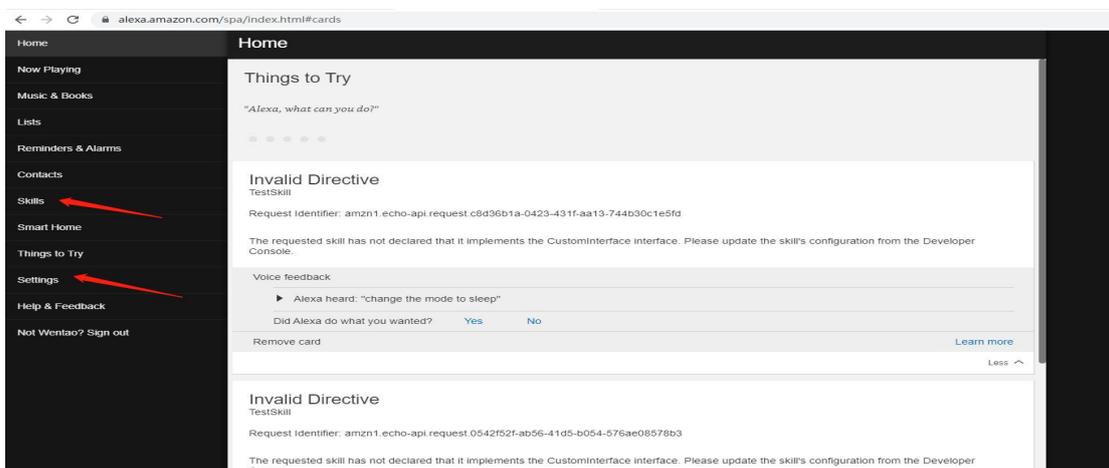
点击进去可以获取 Amazon ID 以及 Amazon Gadget Secret 值，此处的两个值与后面代码里面的 ALEXA_DEVICE_TYPE、ALEXA_DEVICE_SECRET 相对应，Skills 的开发与这两个 ID 是挂钩的，所以**很重要**。



1.3、配置 Echo 音箱

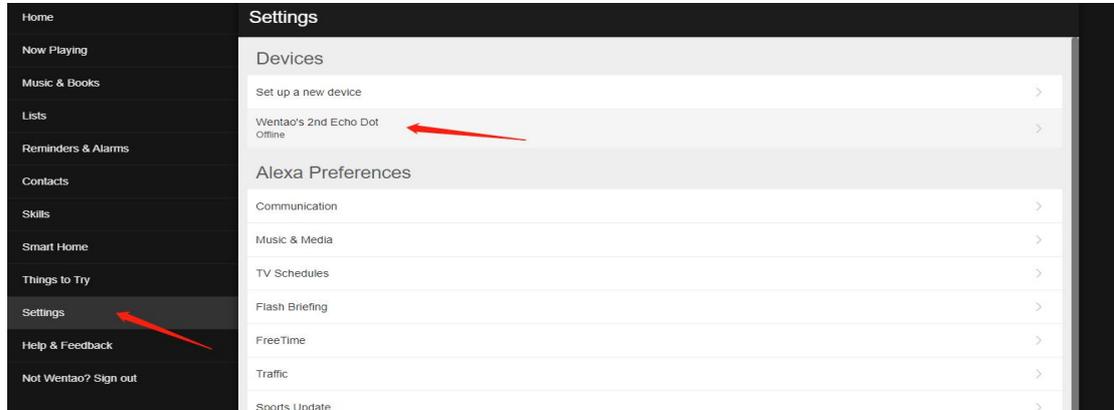
链接：<https://alexa.amazon.com>

创建账户，账户与 alexa 开发者账户一样就行。登录之后如下界面，常用的即为 skills 与 settings

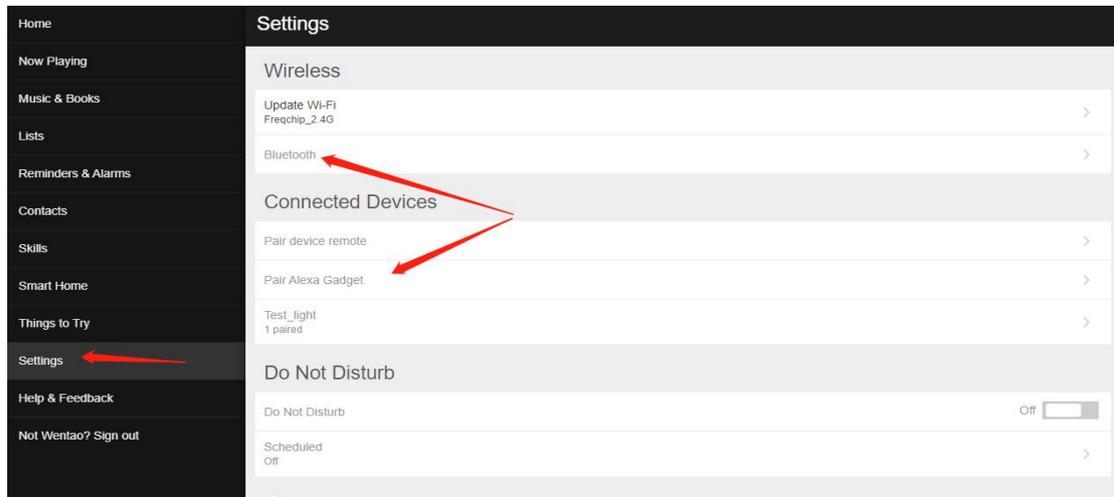


1.3.1、添加 Echo 音箱

在 Settings 界面，选择 set up a new device->Echo Dot，根据提示一步一步操作，开始 Echo 会共享出一个网络“Amazon-XXX”，电脑连接上共享网络之后即可与 Echo 音箱通讯，然后进行后面的配置，给 Echo 音箱配置一个新的 wifi 连接，配置完成之后可以看到界面上会显示有你的 Echo 音箱设备：



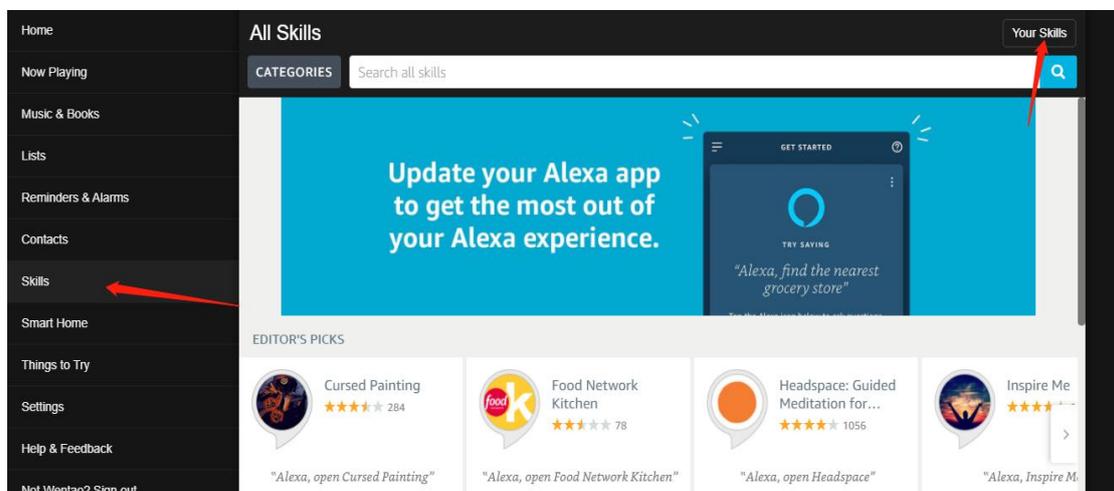
点击进去，选择 Bluetooth 即可添加新的蓝牙设备：



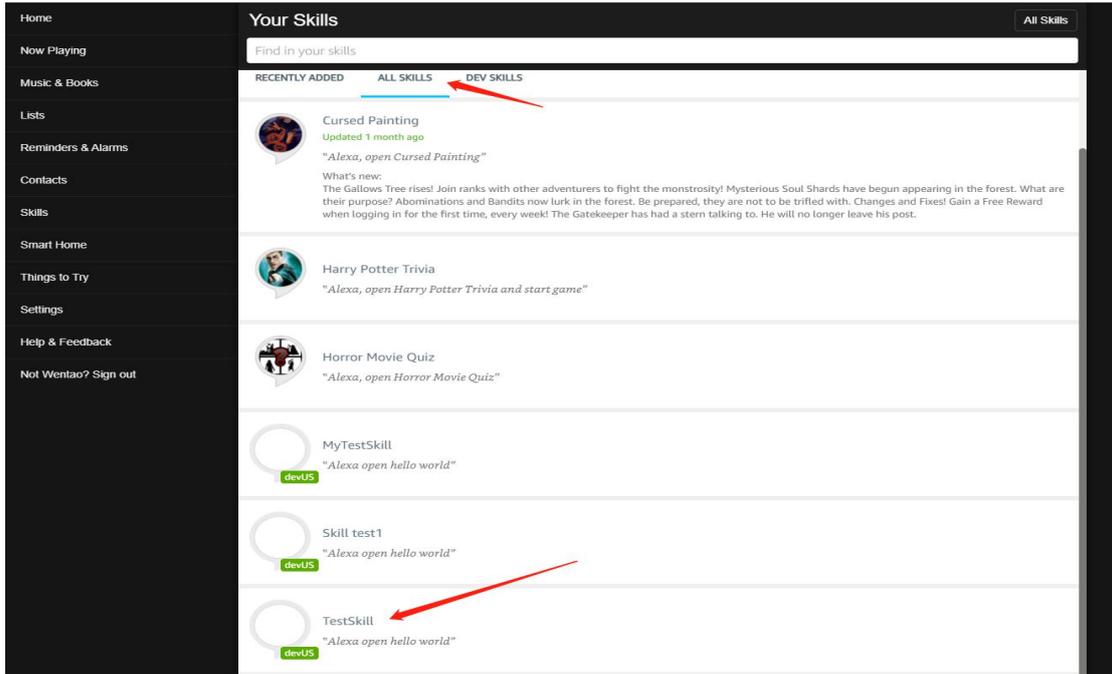
蓝牙设备添加成功之后，语音唤醒 Alexa，在设备 uart 即可看到 log 信息。

1.3.2、Skill 使能

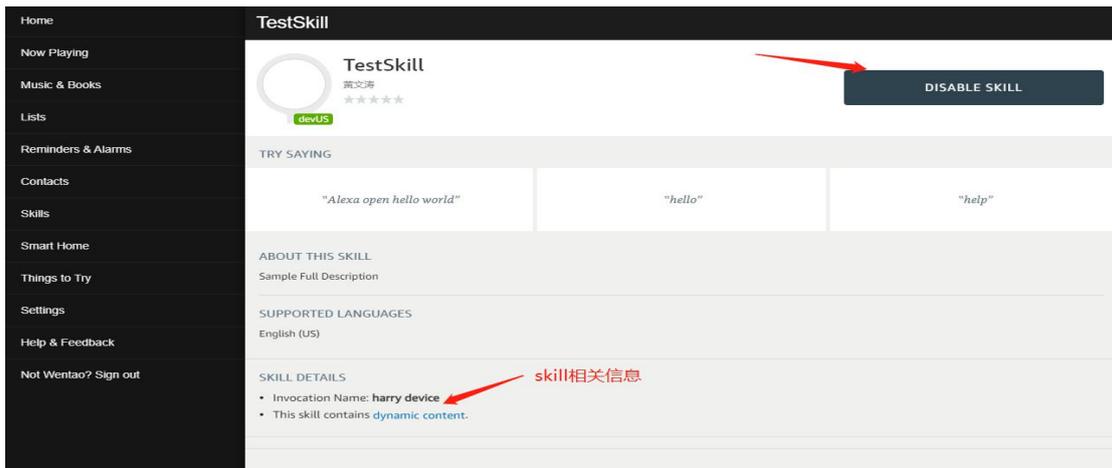
在 skills 界面点击右上角“your skills”，进入用户创建的 skills 界面



点击 all skills，即可看到自己创建的 skills



点击进入 skill 界面，并 enable skill。



2、Alexa 设备端接口配置

在使用 alexa 接口前，需要设定 alexa_config.h 文件，文件内说明如下：

```

//fw version
#define FIRMWARE_VERSION (0x00000001)
//The Gadget's name, the length better less than 26
#define SCAN_RSP_NAME "test Gadget"
//mac addresss
#define LOCAL_MAC_ADDR {0xBD, 0xAD, 0xD0, 0xF0, 0x90, 0x21}

// Alexa GATT Profile Service UUID
#define ALEXA_SVC_UUID (0xFE03)
#define ALEXA_TX_UUID {0x76, 0x30, 0xF8, 0xDD, 0x90, 0xA3, 0x61, 0xAC, 0xA7, 0x43, 0x05, 0x30, 0x77, 0xB1, 0x4E, 0xF0}
#define ALEXA_RX_UUID {0x0B, 0x42, 0x82, 0x1F, 0x64, 0x72, 0x2F, 0x8A, 0xB4, 0x4B, 0x79, 0x18, 0x5B, 0xA0, 0xEE, 0x2B}
    
```

其中 SCAN_RSP_NAME 为搜索配对时的名称，按需要修改
 LOCAL_MAC_ADDR 为本设备的 mac 地址，代码已在 user_custom 接口中用 chip_unique_id 做了设置，以保证每个设备不一样，也可以自行设置
 其他 Service UUID 项为固定项，请勿修改

```

//for device infomation
#define ALEXA_SERIAL_NUMBER "A2GXZ1C61BRBBKX005"
#define ALEXA_FRIENDLY_NAME "Gadget-Test"
#define ALEXA_DEVICE_TYPE "A2GXZ1C61BRBBK"
#define ALEXA_DEVICE_SECRET "7C46D2BF27761E7D00D535E1E2158B3E282AC83A63621181C2703B4B6DD585CB"

```

Device information 项需要按照实际情况进行修改。

ALEXA_SERIAL_NUMBER 该项为本机设备的设备串号，为用户自定义项，但**需要确保每台设备唯一**

ALEXA_FRIENDLY_NAME 为设备昵称，可自行设定。

ALEXA_DEVICE_TYPE 为开发人员注册的亚马逊设备 ID

ALEXA_DEVICE_SECRET 为开发人员注册的亚马逊设备 ID 对应的秘钥

以上信息十分重要，请确保无误。

```

//for discover response
#define ALEXA_DISCOVER_RESPONSE_ENDPOINTID ALEXA_SERIAL_NUMBER
#define ALEXA_DISCOVER_RESPONSE_FRIENDLYNAME ALEXA_FRIENDLY_NAME
#define ALEXA_DISCOVER_RESPONSE_DEVICETYPE ALEXA_DEVICE_TYPE
#define ALEXA_DISCOVER_RESPONSE_RADIOADDRESS "B0ADD0F08010"
#define ALEXA_DISCOVER_RESPONSE_FIRMWARE "01"
#define ALEXA_DISCOVER_RESPONSE_ENCRYPTIONTYPE "1"
#define ALEXA_DISCOVER_RESPONSE_MODELNAME "TestGedget"
#define ALEXA_DISCOVER_RESPONSE_DESCRIPTION "TestGedget1"
#define ALEXA_DISCOVER_RESPONSE_MANUFACTURERNAME "tf"
#define ALEXA_DISCOVER_RESPONSE_CAPABILITIES {
    /*type interface version supported-type-name(options)*/\
    {"AlexaInterface", "Notifications", "1.0", ""},\
    {"AlexaInterface", "Alexa.Gadget.StateListener", "1.0", "alarms,timers,reminders,wakeword,timeinfo"},\
    {"AlexaInterface", "Custom.Interface", "1.0", ""},\
}

```

Discover response 较为重要的部分

Capabilities

该部分为向 alexa 报告当前设备 support 的能力类型，demo 项目中存在基础系统 notify 部分，和 custom interface 部分。

注意：如果没有在此部分声明支持的接口，则设备无法接收到对应的 event。

3、接口说明

3.1、payload 数据回调接口

3.1.1、接口 ble_alex_registerSetIndicatorCallBack

注册 notify 数据回调，此消息为系统内置消息。可用来通知用户有新的消息等，用户收到提醒之后可以做相应的操作，比如询问 Echo 有些什么消息通知。

详细数据说明以及触发说明，请参考：

<https://developer.amazon.com/en-US/docs/alexa/alexa-gadgets-toolkit/notifications-interface.html>

3.1.2、ble_alex_registerDiscoverCallBack

注册 discover 数据回调，此消息为系统内置消息。Echo 可以通过该消息获取当前设备的制造商或者版本信息，或者支持哪些小工具功能。

详细说明请参考：

<https://developer.amazon.com/en-US/docs/alexa/alexa-gadgets-toolkit/alexa-discovery-interface.html>

3.1.3、ble_alex_registerStateUpdateCallBack

注册 state 数据回调，此消息为系统内置消息。设置或者清除计时、警报等功能。有个测试版本是来自 Echo 设备的时间消息，没 30 分钟下发一次。

详细说明请参考：

<https://developer.amazon.com/en-US/docs/alexa/alexa-gadgets-toolkit/alexa-gadget-statelistener-interface.html>

3.1.4、ble_alex_registerSpeechMarksCallBack

注册 speechMarks 数据回调。语音标记功能，将语音与视觉同步的功能。

详细说明请参考：

<https://developer.amazon.com/en-US/docs/alexa/alexa-gadgets-toolkit/alexa-gadget-speechdata-interface.html>

3.1.5、ble_alexa_registerCustomRequestCallBack

注册用户自定义指令回调。

Sdk 包含了一套自定义指令，其结构为：

```
namespace='Custom.Interface',name='request'
```

payload 为自定义 json 结构(与 Skill 配置相关，后续说明)

3.2、directive 解码接口

3.2.1、ble_alexa_decodeDirective

说明：解析 Alexa 发送给蓝牙设备的指令，并返回指令类型。

解析过程中如果存在指令回调，则执行回调之后返回。如果需要获取指令内部数据，请使用 payload 数据回调接口获取。

4、Skills 说明

亚马逊针对设备 skills 主要包含 2 种类型，1.custom skill。2.smart home skill。

两者主要的区别在如下

Custom skills:

- 有较高的灵活性，可以自由配置对话内容以及交互过程。
- 需要有一定的 python/node.js 等相关的基础，需要自行编写 lambda_function 代码。
- 亚马逊提供免费托管服务，无需搭建服务器即可使用。

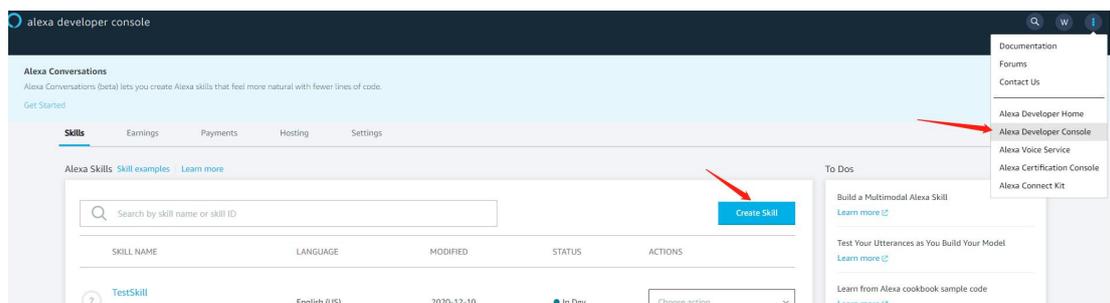
Smart home skills:

- 亚马逊内置的一套针对智能家居相关的交互接口，包含了几乎市面上常见的设备类型的控制交互逻辑。
- 无需编写 lambda_function 代码，配置逻辑较为简单。
- 需要自行搭建 auth2.0 服务器，需要一定的服务端对接能力。
- 纯蓝牙设备 support 有待对接完善*

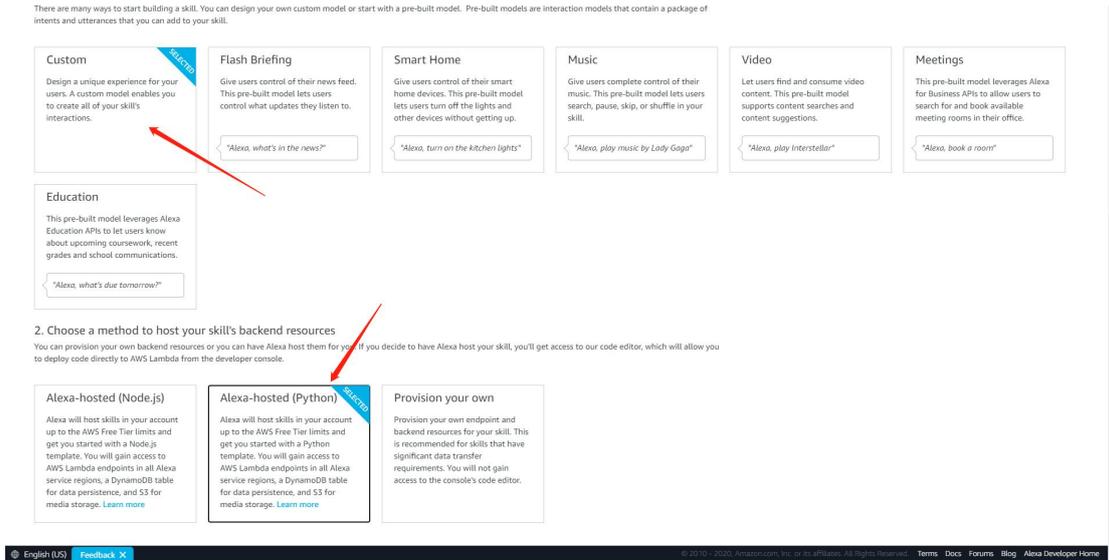
本文档仅针对 custom skills 进行说明，暂不讨论 smart home skills 的情况。

4.1、新增 skills 流程

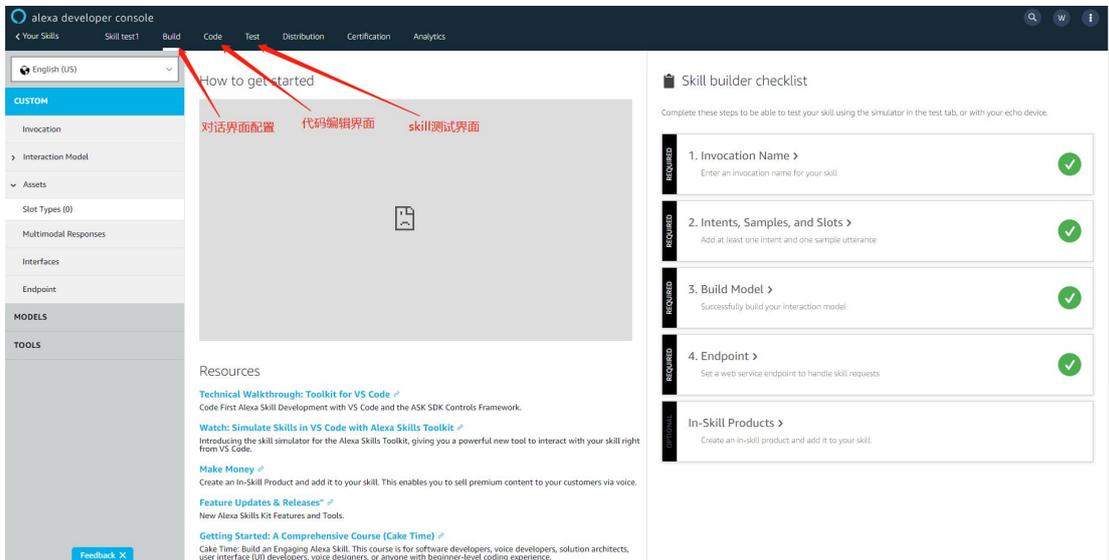
- 登录亚马逊开发者后台
- 点击 alexa developer console，进入 <https://developer.amazon.com/alexa/console/ask>
- 点击 create skill



- 选择 custom、python，然后右上角 create skill。创建完成之后 continue。

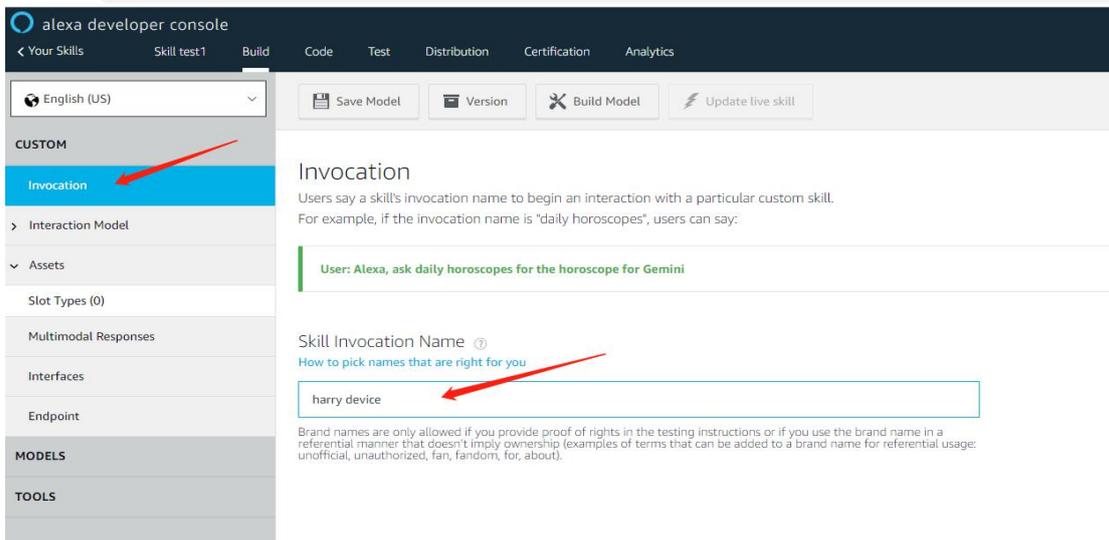


以下即成功创建了 Skill

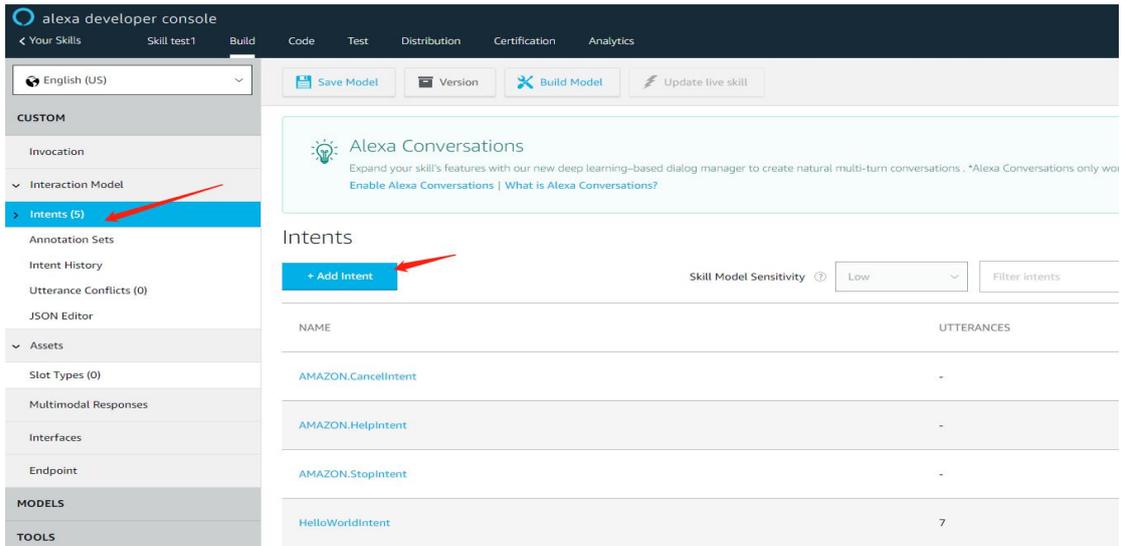


4.2、Skill 新增对话配置

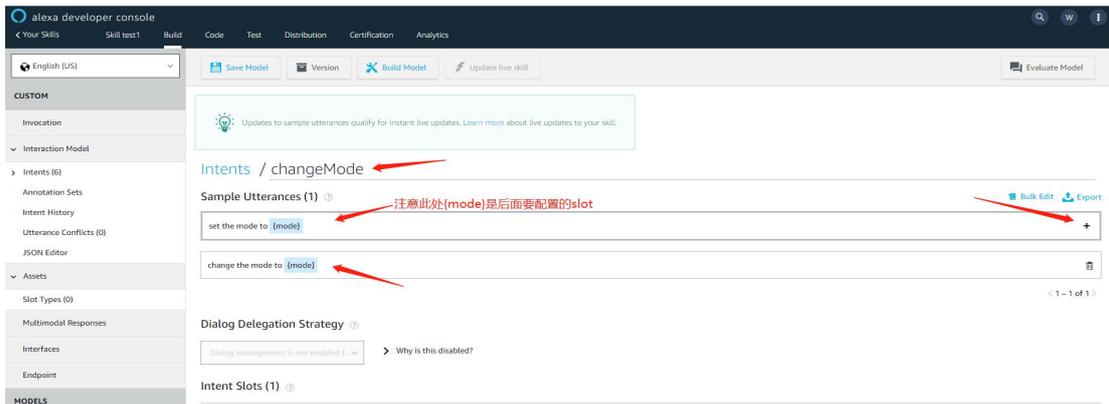
首先先配置 skill 触发词，比如 harry device



新增一条 intent，如下：



设置 intent 的名称，以及 intent 中的触发语义：



此时，即可通过如下语义触发 skill

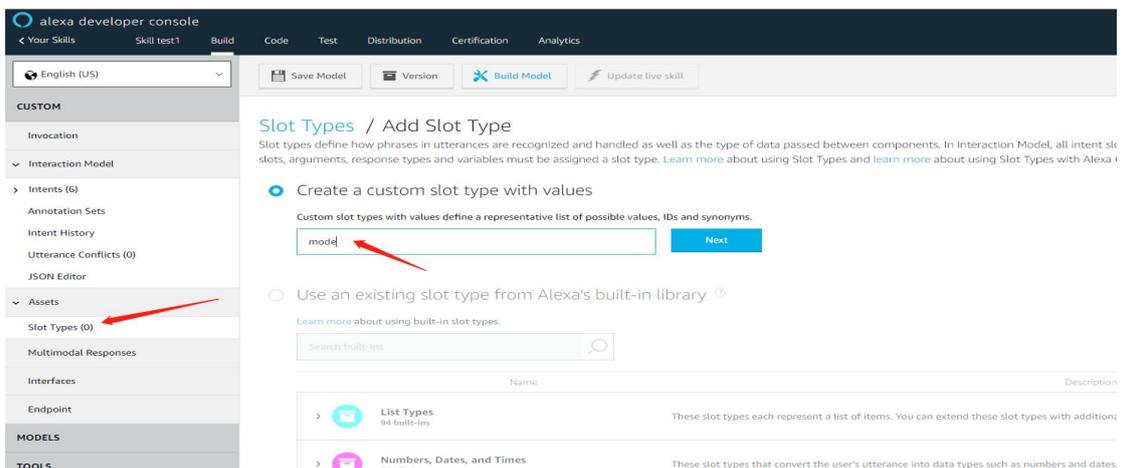
--》 Alexa

--》 open harry device

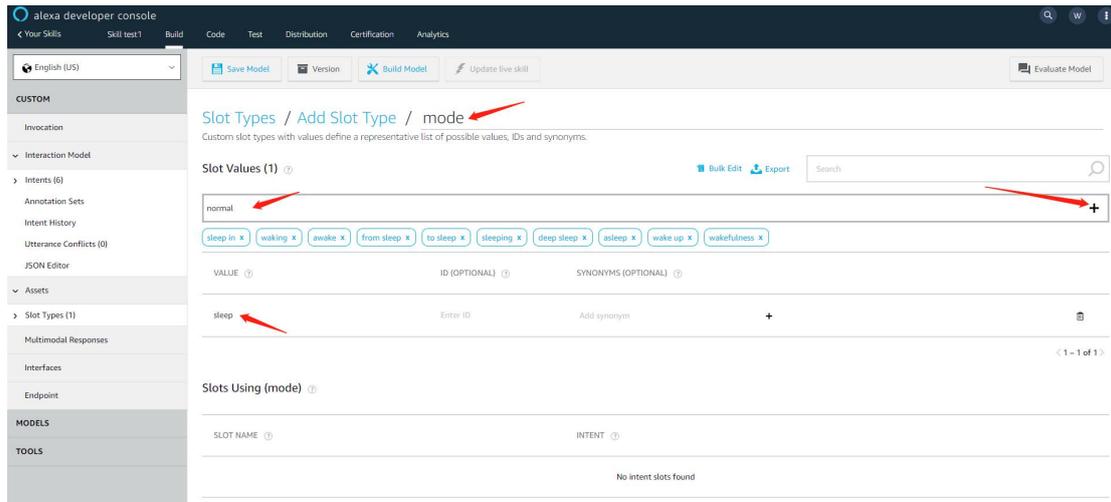
触发语义后，即可通过如下语义触发对应的 intent

--》 change the mode to sleep----sleep 即为{mode}，也就是后面的 slot

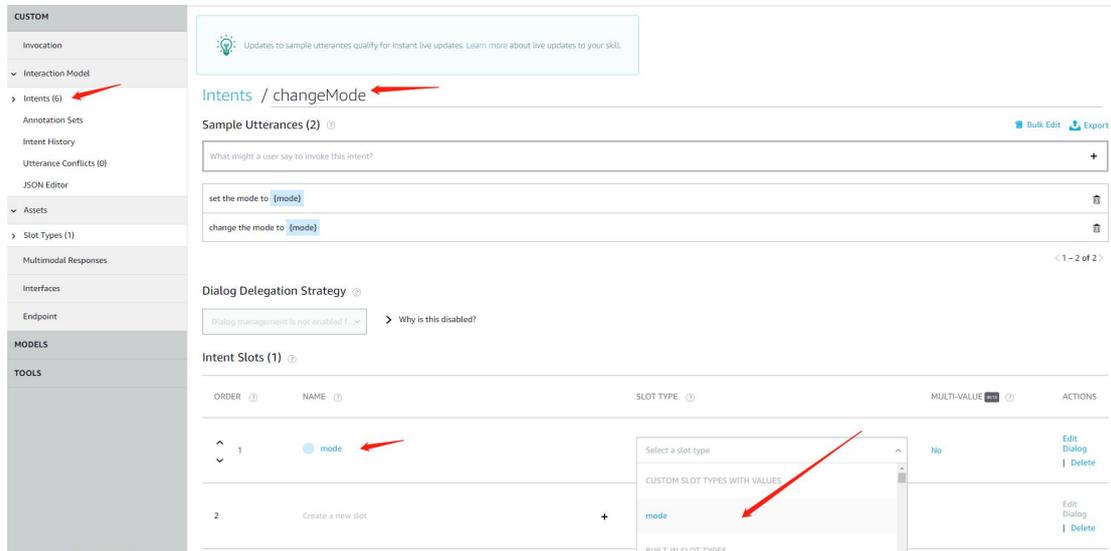
上述过程已经搭建好基础的 intent 交互模型，接下来需要配置 intent 中使用到的 slot，即数据值，注意此处的 slot 名字与上面 intent 使用的 slot 名字一致，此处以{mode}为例：



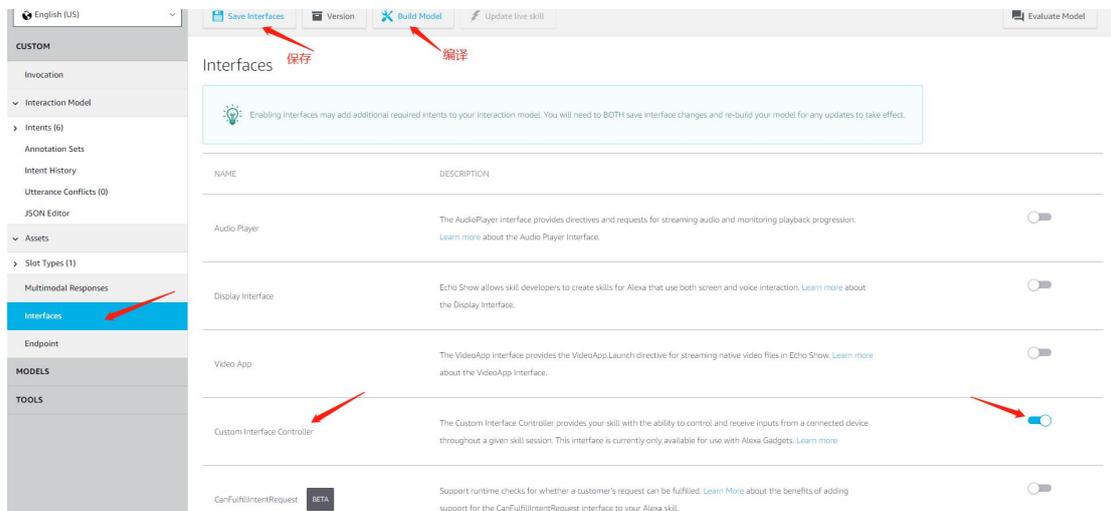
Slot 创建后，添加 slot 中的具体内容：



Slot 创建完成之后注意在上面 intent 界面将 intent 与 slot 关联起来：



上述工作都完成之后，最后在 interfaces 中使能 custom interface：



4.3、python code 编写

上述步骤已经做好基础部分，接下来需要编写对应的 `intent lambda_function` 函数。即可完成触发，对应的 `lambda` 接口请参考 `sdk` 中的 `test.py`，将其内容 `copy` 到 `code` 中。本例中使用的 `lambda intent` 为：

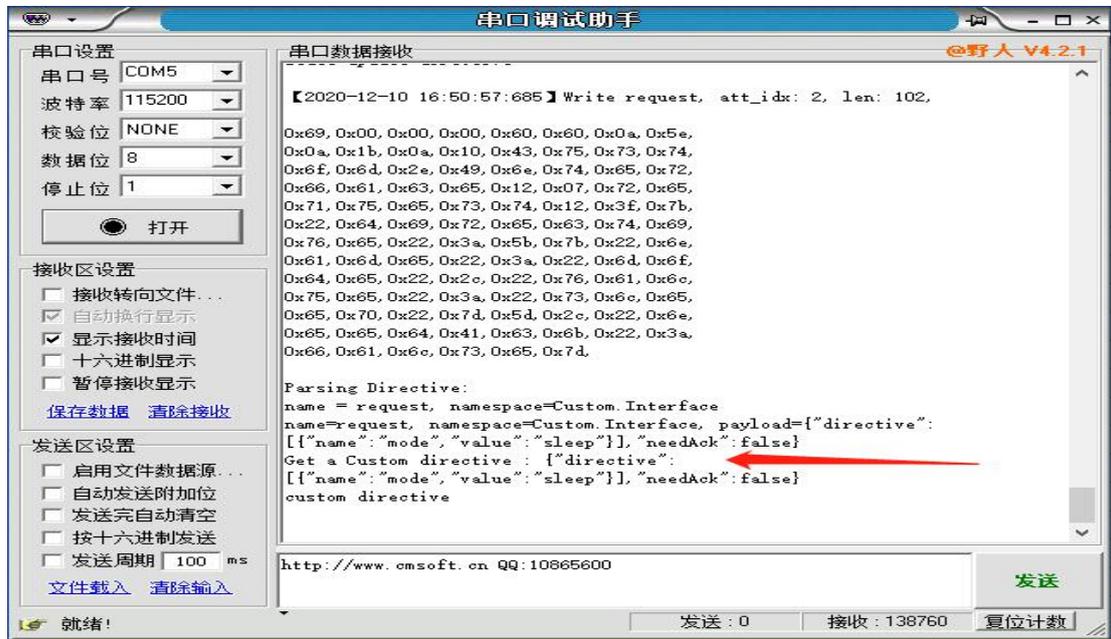
```

44 logger.debug("No connected gadget endpoints available.")
45 return (response_builder
46         .speak("No gadgets found. Please try again after connecting your gadget.")
47         .set_should_end_session(True)
48         .response)
49
50 endpoint_id = endpoints[0].endpoint_id
51
52 # Store endpoint ID for using it to send custom directives later.
53 logger.debug("Received endpoints. Storing Endpoint ID: %s", endpoint_id)
54 session_attr = handler_input.attributes.manager.session_attributes
55 session_attr["endpointId"] = endpoint_id
56
57 return (response_builder
58         .speak("Hi! This is Harry device, what can I do for you?")
59         .set_should_end_session(False)
60         .response)
61
62 @skill_builder.request_handler(can_handle_func_is_intent_name("changeMode"))
63 def change_mode_event_handler(handler_input: HandlerInput):
64     request = handler_input.request_envelope.request
65     session_attr = handler_input.attributes.manager.session_attributes
66     response_builder = handler_input.response_builder
67     endpoint_id = session_attr["endpointId"]
68     slot = get_slot_value(handler_input, "mode")
69
70     return (response_builder
71             .speak("Now change the Harry device mode to {}".str(slot))
72             .add_directive(build_custom_directive(endpoint_id, [{"name": "mode", "value": str(slot)}], False))
73             .set_should_end_session(True)
74             .response)
75
76 @skill_builder.request_handler(can_handle_func_is_intent_name("changeSpeed"))
77 def change_speed_event_handler(handler_input: HandlerInput):
78     request = handler_input.request_envelope.request
79     session_attr = handler_input.attributes.manager.session_attributes
80     response_builder = handler_input.response_builder
81     endpoint_id = session_attr["endpointId"]
82
83     slot = get_slot_value(handler_input, "speed")
    
```

代码变异完成之后点击 `Save` 保存，之后可以在 `build` 界面点击 `build mode` 编译看是否有错。

完成上述步骤后，通过如下交互模型，即可获取到类似如下数据，解析对应的 `payload` 即可

- open harry device
 - change the mode to sleep
- Device debug:



注意：调试用机，需要首先与设备配对。且当前调试用的 `alexa` 必须要与设备 `skills` 在同一个用户账户下，否则无法找到或触发对应 `skills` (`skills` 正式发布后不受此限制，需要用户 `add skills`)。

5、参考文档

Skills python 相关 Api 接口说明:

<https://alexa-skills-kit-python-sdk.readthedocs.io/en/latest/api/core.html?highlight=slot#handler-input>