



Webservice 接入协议

技能平台开发配置

云平台

2018

版本	说明	作者	时间
1.0	初创	郭洪光	
1.1	参见 1.1 变更说明	郭洪光	
1.2.beta.a	参见 1.2.beta.a 变更说明	郭洪光	2019/3/29
1.2.beta.b	参见 1.2.beta.b 变更说明	郭洪光	2019/4/1
1.2	参见 1.2 变更说明	郭洪光	2019/5/23

变更说明

1.1 变更说明

1. 删除 UniOS 请求数据协议中 request 参数下的 intent.id 参数
2. UniOS 请求协议增加参数 request.intent.slots.isFocus 参数
3. UniOS 响应协议 response.target.content 参数修改成 response.target.text

1.2 Beta.a 变更说明

1. 增加终端能力接口上行协议
2. 增加控件协议下行指令协议
3. 增加控件上行事件协议定义
4. request 的 3 个阶段概念更换
原：
 1. requestService 准备请求所需资源
 2. probeService 处理请求数据
 3. normalEndService。结束请求，回收资源背景
 现：
 [标准 request 参数](#)
5. session 的概念更换
原：
 - 1) session 代表人机交互的整个会话
 - 2) session 的创建是在 asr 的 requestService 阶段
 现：
 [Session 参数](#)

1.3 1.2.beta.b 变更说明

1. 增加[指令与 nlu 协议对应关系](#)说明
2. 迁移 directives.provider 节点到 directives.data 下
3. 变更 directives.token/templateCode/page/style 对应 NLU 协议 data.token/templateCode/page/style 节点
4. 变更 directives.data 对应 NLU 协议的 data.data 节点

1.4 1.2 变更说明

5. 增加[网页控件](#)
6. 变更网页[控件类型](#)编码 URL 为 WEB

-
7. 修改所有控件 style 节点，将原对控件属性样式的 json 对象均转化 key: Value 格式，不再使用对象，key 的命名规则采用驼峰，并重新做了命名，具体见各控件 [style](#) 的说明
 8. 修改各控件属性 image.url 首图的对象方式更改为 imageUrl，不再使用对象格式
 9. 修改[指令-nlu 协议](#)关系章节中，去掉 service 领域的转换（技能平台对控件的指令均是使用技能平台自定义意图，并非使用 nlu 的领域）
 10. 在列表控件中，增加节点 directives.data.token

一、 使用范围

有一定研发能力的技能开发者，有想借助 UniOS 的语音识别和语义理解能力，打造个性化或复杂的对话场景

二、 名词解释

UniOS

云知声提供的云知声开放平台

技能开发者

入住 USK 开放平台，有打造个性化技能的开发者，要求实名认证

开发者技能 | 技能

开发者提供的云端技能服务

Webservice 地址

技能开发者云端技能的服务网址，UniOS 会将用户与技能的交互请求转发给此地址

NTP

网络时间协议

三、 接入协议

说明

UniOS 与第三方技能间发送请求，采用 http 或 https 协议（优先推荐），数据格式使用 json，并采用 sha1 签名机制防止数据篡改

1. 签名机制

要求 UniOs 与第三方技能双向请求均需要摘要签名。

签名所需摘要由以下 3 部分组成：

属性	说明
secretKey	归属于技能，系统自动生成（默认 32 字符），用户可修改，最大长度限定 32 字符 不同技能可有不同 secretKey
请求中 Json 数据	请求协议定义的 json 数据

签名步骤：

1. 将 secretKey，json 数据按此顺序排序拼接成形成字符串 digestStr 摘要
2. 使用 sha1 算法对摘要 digestStr 做加密签名，并付值给签名变量 signature
3. 把 signature 放在请求头中，key: signature

验签步骤

1. 通过请求 body 中获取 json 数据，结合已知（开发者设置）的 token，将二者按 secretKey，json 数据顺序生成字符串 digestStr 摘要
2. 通过 sha1 加密 digestStr 摘要，并付值给签名变量 signature_temp
3. 使用 signature_temp 与请求头中获取的 signature 进行比对，相同代表验签通过

2. 数据协议

UniOS 请求数据协议

UniOS 向技能发送的数据协议，请求协议中包含如下四个节点：

属性	说明
version	数据协议版本
context	终端信息，含软硬件参数，用户信息等
session	用户与技能的会话，一次会话可有多次包
request	用户与技能的单次请求 包含：标准请求，见： 标准 request 参数 ；事件请求，见： 事件 request 参数

session 参数

session 代表技能会话，是从语义命中技能开始到技能结束的完整会话。

节点	必填	说明
new	Y	是否是新的会话 session，true：是，false：否 命中技能创建新会话
sessionId	Y	会话 Id
attributes	N	用户与技能同一会话期间的交互数据清单 1、在请求 normalEndService 类型时发给第三方技能，格式见数据协议章节中实例

context 参数

说明：

1、说明中带有字样“asr 协议头字段”参数，要求 tr 调用 nlu-manager 服务室带上此参数。

节点	必填	说明
user	N	用户信息
user.userId	N	在有用户认证场景，可传入 userId asr 协议头字段 userId
device	Y	设备信息
device.udid	Y	设备唯一标识 asr 协议头字段 udid
device.info	N	客户端信息 asr 协议头字段 clientinfo
device.imei	N	设备 IMEI asr 协议头字段 imei
device.interfaces	N	设备支持接口
device.interfaces.voiceInput	N	语音输入支持，默认支持 1:支持 0:不支持

device.interfaces.voiceOutput	N	语音输出支持，默认支持 1:支持 0:不支持
device.interfaces.audio	N	音频接口参数，默认支持 代表终端支持语音拾取和音视频
device.interfaces.audio.formats	N	支持音频格式，多个以逗号间隔，默认支持格式清单，见： 支持音频格式/编码/采样率 asr 协议头字段 audioFormat
device.interfaces.audio.codecs	N	支持音频编码，多个以逗号间隔，默认支持编码清单，见： 支持音频格式/编码/采样率 asr 协议头字段 audioCodec
device.interfaces.audio.bitrates	N	支持音频采样率，默认 16k，支持清单，见： 支持音频格式/编码/采样率 asr 协议头字段 sampleRate
device.interfaces.vedio	N	视频接口参数，默认不支持
device.interfaces.audio.formats	N	支持视频格式，多个以逗号间隔，支持清单见： 支持视频格式/编码/分辨率
device.interfaces.vedio.codecs	N	支持视频编码，多个以逗号间隔，支持清单，见： 支持视频格式/编码/分辨率 trafficParamater 中
device.interfaces.vedio.bitrates	N	支持视频分辨率，支持清单，见： 支持视频格式/编码/分辨率
device.interfaces.vedio.width	N	支持视频宽度，单位：px
device.interfaces.vedio.height	N	支持视频高度，单位：px
skill	Y	技能
skill.skillId	Y	技能 id

标准 request 参数

节点	必填	说明
type	Y	请求类型，请求分 3 个阶段，取值： 1. start 进入技能时 2. process 交互过程 3. end 。技能结束，回收资源
requestId	Y	单次请求标识 Asr 协议中 sessionRequestID
timestamp	Y	此次请求时间戳,单位 ms
info	N	请求的信息
info.type	N	请求信息类型，取值 1、 TEXT 文本格式

		2、JSON json 格式
info.recongize	N	语音识别结果
intent	N	USK 语义到的意图 目前支持 1 个
intent.name	Y	意图标识
intent.isConfirm	Y	意图是否已确认 1:是 0:否 所有槽位已确认，且选项“是否需要确认意图”未被勾选
intent.slots	Y	意图中所有槽位
intent.slots.name	Y	槽位名称，唯一标识
intent.slots.isConfirm	Y	是否已确认槽位 1:是。0:否（默认）
intent.slots.isFocus	Y	是否是焦点槽位，即：本次语句命中槽位 1:是 0:否（默认）

请求类型，请求分 3 个阶段，取值：

1. start 准备请求所需资源
2. process 交互阶段
3. end。结束请求，回收资源

每个阶段请求数据协议有所不同。

start 请求

- 要求 nlu-manager 维护当前技能会话 session 期间对应的技能 context.skill，和终端信息和支持能力接口 context.device，从而打造【技能状态管理中心】，即：技能会话状态机

```
{
  "version": "1.0",
  "context": {
    "user": {
      "userId": "11"
    },
    "device": {
      "udid": "9090909fssdsdfsdf",
      "info": "xxxxxx",
      "imei": "imei",
      "interfaces": {
        "voiceInput": 1,
        "voiceOutput": 1,
        "audio": {
          "formats": [
            "audio/x-wav",
            "pcm"
          ]
        }
      }
    }
  }
}
```

```
    ],
    "codecs": [
        "opus"
    ],
    "bitrates": "16k"
},
"vedio": {
    "formats": [
        "H.264",
        "H.265"
    ],
    "codecs": [
        "AAC",
        "MP3"
    ],
    "bitrates": "360P",
    "width": 1280,
    "height": 768
}
},
"skill": {
    "skillId": "20181101233422"
}
},
"session": {
    "new": true,
    "sessionId": "sessionid",
    "attributies": [

    ]
},
"request": {
    "type": "start",
    "requestId": "requestId",
    "timestamp": "timestamp",
    "info": {
        "type": "TEXT",
        "recongize": "今天天气怎么样"
    },
    "intent": {
        "name": "查气温",
        "isConfirm": 0,
        "slots": [
```



```
{  
    {  
        "name": "时间",  
        "value": "今天",  
        "sort": 2,  
        "isConfirm": 1,  
        "isFocus": 1  
    }  
}  
]
```

process 请求

```
{
  "version": "1.0",
  "context": {
    "user": {
      "userId": "11"
    },
    "skill": {
      "skillId": "20181101233422"
    }
  },
  "session": {
    "new": false,
    "sessionId": "sessionid"
  },
  "request": {
    "type": "process",
    "requestId": "requestId",
    "timestamp": "timestamp",
    "info": {
      "type": "TEXT",
      "recongize": "北京"
    },
    "intent": {
      "name": "查气温",
      "isConfirm": 1,
      "slots": [
        {
          "name": "时间",
          "value": "今天",
          "isConfirm": 1,

```

```
        "isFocus":0
      },
      {
        "name":"地点",
        "value":"北京",
        "isConfirm":1,
        "isFocus":1
      }
    ]
  }
}
```

end 请求

```
{
  "version":"1.0",
  "context":{
    "user":{
      "userId":"11"
    },
    "device":{
      "udid":"9090909fssdsdfsdf",
      "info":"xxxxxx",
      "imei":"imei"
    },
    "skill":{
      "skillId":"20181101233422"
    }
  },
  "session":{
    "new":false,
    "sessionId":"sessionid",
    "attributies":[
      {
        "user":"你好"
      },
      {
        "skill":"你也好"
      },
      {
        "user":"今天天气怎么样"
      }
    ]
  }
}
```

```
        "skill": "问哪个城市"
    },
    {
        "user": "北京"
    },
    {
        "skill": "北京今天天气晴，温度 4-20 度"
    }
]
},
"request": {
    "type": "end",
    "requestId": "requestId",
    "timestamp": "timestamp",
    "reason": "退出原因",
    "error": {
        "type": "异常类型",
        "desc": "异常描述"
    }
}
}
```

事件 request 参数

- 语义意图或手动触控或程序等方式控制控件的行为变化称为事件
- 云端渲染，事件将由控件模版发起上行请求，对终端非透明
- 终端渲染，事件由终端应用自行发起上行请求（对应参数 eventType=事件类型(新增)），具体见事件上行接口定义

1. AudioPlayer 事件

AudioPlayer.Played

播放事件

节点	必填	说明
type	Y	AudioPlayer.Played
requestId	Y	单次请求标识 Asr 协议中 sessionRequestId
timestamp	Y	此次请求时间戳,单位 ms
token	Y	音视频 token
widgetToken	N	音视频控件 token
offsetInMilliseconds	Y	事件上行时播放的时长

```
{
  "version": "1.0",
  "context": {
    "user": {
      "userId": "11"
    }
  },
  "session": {
    "new": false,
    "sessionId": "sessionid"
  },
  "request": {
    "type": "AudioPlayer.Played",
    "requestId": "requestId",
    "timestamp": "timestamp",
    "token": "音频唯一标识",
    "widgetToken": "控件唯一标识",
    "offsetInMilliseconds": 25000
  }
}
```

AudioPlayer.Stoped

停止播放事件

节点	必填	说明
type	Y	AudioPlayer.Stoped
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.Paused

暂停播放事件

节点	必填	说明
type	Y	AudioPlayer.Paused
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.Finished

播放完成事件

节点	必填	说明
type	Y	AudioPlayer.Finished
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.Loading（需要自研播放器）

播放记载中事件

节点	必填	说明
----	----	----

type	Y	AudioPlayer.Loading
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.Closed

播放器关闭事件

节点	必填	说明
type	Y	AudioPlayer.Closed
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.StopRelativePoint

相对播放时长所触发的事件，即：相对 AudioPlayer.ACT_PLAY 指令参数 **offsetInMilliseconds** 开始，向后 **stopPointsInMilliseconds** 秒后停止播放，并触发上行此事件

节点	必填	说明
type	Y	VedioPlayer.StopRelativePoint
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.StopFixPoint

播放到指定时长所触发的事件，即：相对 AudioPlayer.ACT_PLAY 指令参数 **stopFixPointsInMilliseconds** 秒后停止播放，并触发上行此事件

节点	必填	说明
type	Y	VedioPlayer.StopFixPoint
其他同 AudioPlayer.Played 事件参数		

AudioPlayer.Exception

异常事件

节点	必填	说明
type	Y	AudioPlayer.Exception
requestId	Y	单次请求标识 Asr 协议中 sessionRequestID
timestamp	Y	此次请求时间戳,单位 ms
token	Y	音视频 token
widgetToken	N	音视频控件 token 客户端生成，上行给技能
code	Y	播放媒介异常码，见： 多媒体异常
msg		播放媒介异常描述，见： 多媒体异常

2. VedioPlayer 事件

VedioPlayer.Played

视频播放事件

节点	必填	说明
type	Y	VedioPlayer.Played
requestId	Y	单次请求标识 Asr 协议中 sessionRequestID
timestamp	Y	此次请求时间戳,单位 ms
token	Y	视频 token
widgetToken	N	视频控件 token 客户端生成, 上行给技能
offsetInMilliseconds	Y	事件上行时播放的时长

VedioPlayer.Stoped

视频停止播放事件

节点	必填	说明
type	Y	VedioPlayer.Stoped
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.Paused

视频暂停播放事件

节点	必填	说明
type	Y	VedioPlayer.Paused
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.Finished

视频播放完事件

节点	必填	说明
type	Y	VedioPlayer.Finished
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.Loading（需要自研播放器）

视频加载中事件

节点	必填	说明
type	Y	VedioPlayer.Loading
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.Closed

视频控件关闭事件

节点	必填	说明
type	Y	VedioPlayer.Closed
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.StopRelativePoint

相对播放时长所触发的事件，即：相对 VedioPlayer.Play 指令参数 **offsetInMilliseconds** 开始，向后 **stopPointsInMilliseconds** 秒后停止播放，并触发上行此事件

节点	必填	说明
type	Y	VedioPlayer.RelativePoint
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.StopFixPoint

播放到指定时长所触发的事件，即：相对 VedioPlayer.Play 指令参数 **stopFixPointsInMilliseconds** 秒后停止播放，并触发上行此事件

节点	必填	说明
type	Y	VedioPlayer.StopFixPoint
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.SkipOpened

跳过片头并播放事件

节点	必填	说明
type	Y	VedioPlayer.SkipOpened
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.SkipEnd

跳过片尾事件

节点	必填	说明
type	Y	VedioPlayer.SkipEnd
其他同 VedioPlayer.Played 事件参数		

VedioPlayer.Exception

异常事件

节点	必填	说明
type	Y	VedioPlayer.Exception
其他同 AudioPlayer.Exception 事件参数		

3. selected 事件

适用列表控件

Item.selected 事件参数

节点	必填	说明
type	Y	Item.selected
requestId	Y	单次请求标识

		Asr 协议中 sessionRequestID
timestamp	Y	此次请求时间戳,单位 ms
token	Y	列表中 Item 的 token
widgetType	Y	控件类型, 见: 控件类型
widgetToken	Y	控件 token 上行给技能

Webservice 响应数据协议

第三方技能响应 UniOS 请求的数据协议, 包含以下几个部分

属性	说明
version	数据协议版本
response	用户与技能的单次请求

response 参数

response 协议中有说明”对应 NLU 协议”备注的, nlu-manger 需要做协议整合到 nlu 协议中响应给终端。其他由 nlu-manager 运行过程中所用

参数	备注
speech	终端需要播报的语音 覆盖本次 UniOS 语义播报结果 对应 NLU 协议 general
intent	第三方技能认为用户语句更符合的意图 覆盖 UniOS 本次语义意图 (在 UniOS 已定义意图)
nextIntent	指定下一个意图 对应 NLU 协议 semantic.nextIntent(新增)
directives	指令, 下行控制终端, 支持同时下发多指令 对应 NLU 协议的 intent.operations
isEndSession	是否关闭当前会话, 1:是 0:否

```
{
  "version": "1.0",
  "response": {
    "speech": {

    },
    "intent": {
```



```

    },
    "nextIntent": "意图唯一标识",
    "directives": [
      {

      }
    ],
    "isEndSession": 1
  }
}

```

speech 参数

speech 协议中内容

参数	备注
speech. type	播报内容数据格式 T (TEXT): 文本格式 SSML: 结构化语言 对应 NLU 协议 general. type
speech. text	合成文本内容 当 type 为 TEXT 时，必填 限 256 字符 对应 NLU 协议 general. text
speech. ssml	语音合成结构化数据 限 256 字符 对应 NLU 协议 general. ssml

```

"speech": {
  "type": "TEXT",
  "text": "内容",
  "ssml": "ssml expression"
},

```

intent 参数

- 技能平台认为更合理的意图，此数据将作为 UniOS 意图的语料持久处理，同时会展示在 UniOS 技术平台中，可以理解为辅助训练

name	意图唯一标识
slots	意图槽位
slots. name	槽位名称
slots. value	槽位确认对应语句的值
slots. dict	槽位对应技能平台词典标识

slots.isMust	槽位是否为必填 0:否 1:是
--------------	-----------------

```
{
  "intent":{
    "name":"意图唯一标识",
    "slots":[
      {
        "name":"地点",
        "value":"北京",
        "dict":"引用技能平台词典",
        "isMust":1
      }
    ]
  }
}
```

nextIntent 参数

- 技能指定下一次对话意图，即：终端执行完本次请求意图后，终端若无指定意图时，NLU 服务将执行此技能指定意图

name	意图唯一标识
------	--------

```
{
  "intent":"意图唯一标识"
}
```

directives 参数

说明：依据 directives.type 值的不同，拥有不同响应，无需关注 nlu 协议中的 service 和 code

type	指令，见： 指令类型 除了音频控件类型，视频控件类型，图文控件类型，列表控件，网页控件。 IoT 协议（对终端硬件本身能力的控制）由技能直接对设备下发指令，不经过 USK，不在此协议中。 Nlu 协议 intent.operations.deviceType
code	指令，见： 指令 NLU 协议 intent.operations.operator
playBehavior	播放行为，见： 播放行为 只有 Type=Widget. (Audio Video)时，有此参数 NLU 协议 intent.operations.playBehavior
token	控件唯一标识

	只有 Type=Widget.* 时，有此参数 技能生成留存，终端上行事件需要携带此参数 若 token 有值，控件重新渲染不改变此 token NLU 协议 data.token
templateCode	模版唯一编码 只有 Type=Widget.* 时有此参数 USK 提供，对每个控件类型至少有一个模版 templateCode 命名规则：控件类型缩写-自定义名称（如：IT-DEFAULT） templateCode 有值，采用云端模版渲染不下行具体数据，客户端渲染下行具体控件数据和样式数据 不同控件模版有不同的 data 数据协议，见：控件数据协议 NLU 协议 data.templateCode
style	控件样式设置, 提供程序级别控制控件样式，可在运行期覆盖技能平台设置的控件参数，见：style 参数 只有 Type=Widget.*时有此参数 Type 不同有不同的响应参数 NLU 协议 data.style
page	分页设置 只有 type=Widget.List，才有此参数 NLU 协议 data.page
controls	操作 当 Type=Widget.*时，才有此参数 为控件增加操作 NLU 协议 data.controls
data	指令响应内容数据参数，见：data 参数 Type 的不同有不同响应参数 NLU 协议 data.data

style 参数

控件样式设置，此节点只负责对控件中元素布局样式控制，不包含具体数据
通常把需要隐藏组件以及有布局设置配置在 sytle 中

图文控件

directives.type=Widget.ImageText & directives.data.type=IT-DEFAULT	
directives.style.titleShow	是否显示标题，默认 1 显示
directives.style.subTitleShow	是否显示副标题, 默认 1 显示
directives.style.providerShow	是否显示数据源，默认 1 显示

directives.style.controlsShow	是否显示标签操作，控制节点 widgets.directives.controllss 可见，默认 1 显示
-------------------------------	--

列表控件

directives.type=Widget.List& directives.data.type=LH-DEFAULT	
directives.style.snShow	是否显示序号
directives.style.titleShow	是否显示标题，默认 1 显示
directives.style.subTitleShow	是否显示副标题, 默认 1 显示
directives.style.tagShow	是否显示标签，默认 1 显示
directives.style.imageShow	是否显示图片，默认 1 显示
directives.style.imagePosition	图片相对标题/副标题的位置 同： 图文位置-图文控件
directives.style.providerShow	是否显示数据源，默认 1 显示

音频控件

directives.type= Widget.AudioPlayer	
directives.style.imageShow	是否显示图片，默认 1 显示
directives.style.imagePosition	图片相对标题/副标题的位置 同： 图文位置-图文控件
directives.style.titleShow	是否显示标题，默认 1 显示
directives.style.subTitleShow	是否显示副标题, 默认 1 显示
directives.style.providerShow	是否显示数据源，默认 1 显示
directives.style.prevShow	是否显示上一首操作，默认 1 显示
directives.style.nextShow	是否显示下一首操作，默认 1 显示
directives.style.playShow	是否显示播放操作，默认 1 显示
directives.style.stopShow	是否显示停止操作，默认 1 显示
directives.style.barShow	是否显示进度条，默认 1 显示

视频控件

directives.type= Widget.VideoPlayer	
directives.style.videoShow	是否显示视频
directives.style.titleShow	是否显示标题，默认 1 显示
directives.style.subTitleShow	是否显示副标题, 默认 1 显示
directives.style.providerShow	是否显示数据源，默认 1 显示
directives.style.prevShow	是否显示上一首操作，默认 1 显示
directives.style.nextShow	是否显示下一首操作，默认 1 显示
directives.style.playShow	是否显示播放操作，默认 1 显示
directives.style.stopShow	是否显示停止操作，默认 1 显示
directives.style.barShow	是否显示进度条，默认 1 显示

directives.style.fullScreenShow	是否显示全屏幕操作，默认 1 显示
---------------------------------	-------------------

page 参数

列表控件分页数量和每页显示 item 数量

directives.type= Widget.List & directives.templateCode = LH-DEFAULT	
directives.page.limit	每页数量，最大 20，最小 1，由于最大返回资源量为 100，建议此值设置为可被 100 整除，
directives.page.count	最多显示多少页，最大值 10 页，最小 1 页，资源数据少于 10 页面按实际页数显示

controls 参数

directives.type= Widget.*	
directives.controls	操作，支持 0 到 6 个为控件增加的功能操纵，其对应指定技能下的指定意图
directives.controls.name	功能名称
directives.controls.intent	对应的意图编码 具体排序规则由产品规划 通过此编码可执行对应意图

data 参数

说明：

- 第三方技能下发的控件指令
- 支持下发多个控件指令（directives 集合），支持混合指令
- 控件指令兼容无屏和有屏设备（终端）：
 - 控件云端渲染，下行 URL 数据，无需控件数据
 - 控件客户端渲染，下行控件数据
 - 兼容不渲染，只使用控件数据，这种情况终端可忽略样式关于相关参数

directives.type=Widget.*	
directives.data	控件数据，见： 控件数据协议 根据控件模版 directives.templateCode 不同，data 数据协议也不同 Nlu 协议 data.data 节点
directives.data.provider	数据来源 当 Type=Widget*时，且 data 节点有数据时才有此参数 NLU 协议 data.data.provider

provider 参数

directives.type= Widget.* & directives.data 参数有数据	
directives.prodiver.name	内容提供方名称
directives.prodiver.logoUrl	提供方 logo 地址

控件数据协议

- 同类型控件模版的数据协议相同
- 控件所需的资源数据，要求技能一次性响应给 USK，但最大不超过技能平台规定的数量。比如：列表控件中的数据按技能自我设置的 directives.page（此设置会在运行期，优先控件的参数设置，但不会覆盖控件 DB 中存储）中的量返回

1. 图文控件

directives.templateCode=IT-DEFAULT	
directives.data.title	标题
directives.data.subTitle	副标题
directives.data.imageUrl	首图地址 用以默认显示的图片
directives.data.bgImageUrl	背景图片地址
directives.data.linkUrl	图文源地址 点击图文控件跳转地址
<pre>{ "directives": [{ "type": "WIDGET.ImageText", "code": "ACT_OPEN", "token": "控件唯一标识", "templateCode": "模版编码", "style": { "subTileShow": 0, "imageShow": 1, "imagePosition": "LEFT" }, "controls": [{ "name": "功能名称，显示使用", "intent": "意图编码" }], "data": {</pre>	

```
        "title": "标题",
        "subTitle": "副标题",
        "imageUrl": "首张图片",
        "bgImageUrl": "背景图片",
        "linkUrl": "图文链接地址"
    },
    "provider": {
        "name": "数据源名称",
        "logoUrl": "数据源 Logo 地址"
    }
}
]
```

2. 列表控件

directives.templateCode=LH-DEFAULT	
directives.data.token	Item 的唯一标识 由技能提供
directives.data.title	标题
directives.data.subTitle	副标题
directives.data.tag	单条右侧文本，仅支持 1 个
directives.data.imageUrl	单条图标地址
directives.data.bgImageUrl	背景图片地址
directives.data.linkUrl	单条链接地址
<pre>{ "directives": [{ "type": "Widget.List", "code": "ACT_OPEN", "token": "控件唯一标识", "templateCode": "模版编码", "style": { "tagShow": 0, "subTitleShow": 0, "imageShow": 1, "imagePosition": "RIGHT" }, "page": { "limit": 5, "count": 3 }, "controls": [{ "name": "功能名称，显示使用",</pre>	

```
        "intent": "意图编码"
    }
],
"data": [
    {
        "token": "item 唯一标识",
        "title": "标题",
        "subtitle": "副标题",
        "imageUrl": "首张图片",
        "bgImageUrl": "背景图片",
        "linkUrl": "图文链接地址",
        "tag": "标签",
        "provider": {
            "name": "数据源名称",
            "logoUrl": "数据源 logo 地址"
        }
    }
]
}
]
```

3. 音频控件

directives.templateCode=AD-DEFAULT	
directives.data.token	音频唯一标识
directives.data.title	标题
directives.data.subTitle	副标题
directives.data.imageUrl	首图地址 用以默认显示的图片
directives.data.bgImageUrl	背景图片地址
directives.data.author	歌手 Nlu 协议 data.author
directives.data.lyric	歌词，见： 歌词数据协议
directives.data.stream	音频流信息
directives.data.stream.url	音频地址
directives.data.stream.expiryTime	音频地址过期时间, 单位秒
directives.data.stream.totalContentPlayTime	总时长, 单位秒
directives.data.stream.offsetInMilliseconds	跳过时长播放, 单位秒, 默认 0
directives.data.reports	上报事件条件

directives.data.reports.stopPointsInMilliseconds	跳过 offsetInMilliseconds 播放指定时长停止播放，单位秒，支持指定多个时长触发 RelativePoint 事件
directives.data.reports.stopFixPointsInMilliseconds	播放到指定音频时间点停止播放，单位秒，支持指定多个时间点 如：1 分钟音频，播放到 25 秒和 50 秒停止播放
<pre> { "directives": [{ "type": "Widget.AudioPlayer", "code": "ACT_OPEN", "playBehavior": "REPLACE_ALL", "token": "控件唯一标识", "templateCode": "模版编码", "controls": [{ "name": "功能名称，显示使用", "intent": "意图编码" }], "data": [{ "token": "音频唯一标识", "title": "标题", "subtitle": "副标题", "imageUrl": "首张图片", "bgImageUrl": "背景图片", "author": "演唱者", "lyric": { "url": "歌词地址", "format": "歌词格式" }, "stream": { "url": "音频地址", "expiryTime": 60000, "totalContentPlayTime": 60000, "offsetInMilliseconds": 100000 }, "reports": { "stopPointsInMilliseconds": [25000, </pre>	

```
        500000
    ],
    "stopFixPointsInMilliseconds": [
        15000,
        30000
    ]
},
"provider": {
    "name": "数据源名称",
    "logoUrl": "数据源 logo 地址"
}
}
]
}
]
}
```

4. 视频控件

directives.templateCode=VD-DEFAULT	
directives.data.token	视频唯一标识
directives.data.title	标题
directives.data.subTitle	副标题
directives.data.imageUrl	首图地址 用以默认显示的图片
directives.data.bgImage.url	背景图片地址
directives.data.stream	音频流信息
directives.data.stream.url	音频地址
directives.data.stream.expiryTime	音频地址过期时间, 单位秒
directives.data.stream.totalContentPlay Time	总时长, 单位秒
directives.data.stream.offsetInMilliseconds	跳过时长播放, 单位秒, 默认 0
directives.data.reports	上报事件条件
directives.data.reports.stopPointsInMilliseconds	跳过 offsetInMilliseconds 播放指定时长停止播放, 单位秒, 支持指定多个时长 触发 RelativePoint 事件
directives.data.reports.stopFixPointsInMilliseconds	播放到指定音频时间点停止 播放, 单位秒, 支持指定多 个时间点 如: 1 分钟音频, 播放到 25 秒和 50 秒停止播放

```
{
  "directives": [
    {
      "type": "WIDGET.Video",
      "code": "ACT_PLAY",
      "playBehavior": "REPLACE_ALL",
      "token": "控件唯一标识",
      "templateCode": "模版编码",
      "controls": [
        {
          "name": "功能名称，显示使用",
          "intent": "意图编码"
        }
      ],
      "data": [
        {
          "token": "视频唯一标识",
          "title": "标题",
          "subtitle": "副标题",
          "imageUrl": "预览图片地址",
          "bgImageUrl": "背景图片",
          "stream": {
            "url": "音频地址",
            "expiryTime": 60000,
            "totalContentPlayTime": 60000,
            "offsetInMilliseconds": 15000
          },
          "reports": {
            "stopPointsInMilliseconds": [
              10000,
              30000
            ],
            "stopFixPointsInMilliseconds": [
              15000,
              30000
            ]
          },
          "provider": {
            "name": "数据源名称",
            "logoUrl": "数据源 logo 地址"
          }
        }
      ]
    }
  ]
}
```

```
    ]
}
```

5. 网页控件

directives.templateCode=WEB-DEFAULT	
directives.data.url	标题
<pre>{ "directives": [{ "type": "Widget.WEB", "code": "ACT_OPEN", "token": "控件唯一标识", "templateCode": "模版编码", "data": { "url": "网页地址" } }] }</pre>	

歌词数据协议

lyric	歌词
url	歌词地址
format	歌词格式
<pre>"lyric": { "url": "歌词地址", "format": "歌词格式" }</pre>	

通用参数枚举

支持音频格式/编码/采样率

分类	值	说明
格式	audio/x-wav	
	pcm	
	speex	

编码	opus	
	opus-nb	
	adpcm	
	speex	
	pcm	
采样率	16k	
	8k	
	6k	

支持视频格式/编码/分辨率

分类	值	说明
格式	H.264	
	H.265	
	MPEG-2	
	VP8	
	VP9	
编码	AAC	
	MP3	
分辨率	360P	
	480P	
	720P	
	1080P	
	4K	

多媒体异常

分类	值	说明
异常	MEDIA_404	音视频地址无效，可能的原因：请求失败/地址错误/无授权等
	MEDIA_500	音视频服务器发生异常
	MEDIA_601	无法解析音视频，可能原因：不支持的格式/编码/采样率/分辨率等
	MEDIA_0	未知异常

控件类型

类型	值（全写）	值（缩写）
图文	IMAGE-TEXT	IT
列表-横向	LIST-H	LH
音频播放	VOICE-PLAYER	VI

视频播放	VEDIO-PLAYER	VD
内嵌网页	WEB	WEB

图文位置-图文控件

值	说明
LEFT	图片在标题左
RIGHT	图片在标题右，默认
TOP	图片在标题和副标题上
CENTER	图片在标题和副标题中间
BOTTOM	图片在标题和副标题下

图文位置-列表控件

值	说明
LEFT	图片在标题左，默认
RIGHT	图片在标题右

指令类型

类型	值
音频控件	Widget.AudioPlayer
视频控件	Widget.VedioPlayer
图文控件	Widget.ImageText
列表控件	Widget.List
Url 控件	Widget.Url
IoT 设备	IoT 相关设备，见 IoT 协议

指令

- 表格中黄色指令，实际为播放模式，为了简化技能开发者使用，合并到指令集中。
- nlu 协议中对执行类指令和设置类指令协议有所不同，故 nlu-manager 需要做 NLU 协议转化为，具体指令协议对应关系见：[指令与 nlu 协议对应关系](#)

适用对象	指令	指令类别	说明
音频/视频控件	ACT_PLAY	执行	整合 nlu 转换为 nlu 协议 播放 立刻播放指令音视频，停播当前音视频
	ACT_REPLAY		重播

	ACT_STOP		停止
	ACT_PAUSE		暂停
	ACT_PREV		上一个
	ACT_NEXT		下一个
	MODE_SHUFFLE	设置	随机播放
	MODE_ORDER		顺序播放
	MODE_ALL_REPEAT		循环播放
	MODE_REPEAT_ONCE		单曲循环播放 是循环播放当前音视频还是指令音视频，由 播放行为 决定
	MODE_ONCE		单曲单次播放 当前控件播放歌曲单曲循环
视频控件	ACT_SKIP_OPENING	执行	跳过片头并播放
	ACT_SKIP_END		跳过片尾
	ACT_SKIP_AD		跳过广告并播放
控件	ACT_OPEN	执行	显示控件
	ACT_CLOSE		关闭控件

播放行为

适用对象	指令	说明
音频/视频控件	REPLACE_ALL	清空 控件播放列表，立即播放指令音视频，停播当前音视频 指令 Audio/Video.PLAY 的默认播放行为
	REPLACE_ENQUEUED	把指令音视频 替换 控件播放列表，当前音视频播放不受影响
	ENQUEUE_FRONT	把指令音视频 加入 控件播放列表前部，当前音视频播放不受影响
	ENQUEUE_BEHIND	把指令音视频 加入 控件播放列表尾部，当前音视频播放不受影响

控件上行事件类型

控件类型	事件类型	说明
音频/视频控件 AudioPlayer/AudioPlayer	Played	播放
	Stoped	停止
	Paused	暂停
	Loading	加载中

视频控件 AudioPlayer	Finished	播放完成
	Closed	关闭播放器
	SkipOpening	跳过片头
	SkipEnd	跳过片尾

控件指令与 nlu 协议对应关系

指令分类	适用控件	NLU 协议	
设置	音乐 视频	<code>code</code>	SETTING_EXEC 固定值，同 NLU 协议
		<code>intent.operatios.operator</code>	ACT_SET 固定值，同 NLU 协议
		<code>intent.operatios.operands</code>	ATTR_PLAY_MODE 固定值，同 NLU 协议
		<code>intent.operatios.value</code>	指令 章节中的指令
执行	所有	<code>code</code>	SETTING_EXEC 固定值，同 NLU 协议
		<code>intent.operatios.operator</code>	指令对应的 NLU 协议中的同功能值