

# 一 技能

---

[什么是技能](#)

[技能的多样化](#)

[技能的自然交互](#)

[技能的组成部分](#)

## 什么是技能

技能 (intents) 与手机APP相似，通过语音指令使晓乐机器人完成一个或多个特定的功能。创建一个技能，用户首先要有自己的[意图](#)，即想要做什么。

## 技能的多样化

可以在该技能开放平台上创建各种技能，为用户提供多种的服务，例如：

1. 晓乐为用户提供休闲娱乐服务
  - user：你好晓乐，打开剪刀石头布
  - user：你好晓乐，打开猜数字
2. 晓乐为用户提供控制智能家居的服务
  - user：你好晓乐，请帮我打开卧室灯
  - user：你好晓乐，请帮我打开窗帘
3. 晓乐为用户提供智能问答服务
  - user：你好晓乐，打开工业互联网
  - user：你好晓乐，打开故宫问答

## 技能的自然互动

技能提供的是对话式交互服务。用户就好像与人一样沟通，通过语音就可以完成与技能的交互，享用技能提供的服务。例如：

user：你好晓乐，打开电影播放  
晓乐：你想看什么电影  
user：请帮我播放速度与激情3  
晓乐：搜索到的结果已经为你呈现在屏幕上  
user：第三个  
晓乐：即将为你播放第三个

## 技能组成

[意图 \(intent\)](#)

## 二 意图

---

### 2.1 概述

意图是对话交互的目的，用户通过对晓乐的每一次语音指令想要达到的效果，即晓乐对指令做出的反应。用户对晓乐的每一条指令,都可以认为是一个意图。

### 3.2 举例

音乐播放就是一个[技能](#)它由4个意图构成：

- 1) 听音乐
- 2) sys.用户选择
- 3) sys.确认
- 4) sys.取消

## 三 用户说法

---

### 3.1 概述

用户说法就是用户在对话交互过程中提出的问题，请求或命令。作为语音对话定制平台，我们需要开发者录入可能的用户说法（放在intents的samples中），并确保说法足够丰富。

### 3.2 组成

一个技能由一到多个意图组成，如点菜技能,它的意图可以有查看菜单、点菜、结账等。每个意图中都有各种各样的用户说法。给每个意图一个别名“name”，将例举出的用户说法放在该意图后的“samples”中

```
{
  "name": "GetFoodIntent",
  "samples": [
    "你们有哪些菜可以点",
    "介绍一下你们的特色菜"
  ]
},
{
  "name": "PayMoney",
  "samples": [
    "请帮我结账",
    "请算一下我们一共的消费"
  ]
}
```

### 3.3 标注

用户的说法多种多样我们不可以一一列举完，所有我们要对用户说法中的关键字进行标注。这时候[语义槽](#)（slots）便发挥作用了。比如用户点菜时的用户说法有‘我想点鱼香肉丝’，将‘鱼香肉丝’标注为一个叫‘菜名’的语义槽，通过替换绑定[词典](#)（slot\_types）中的其他菜名值（values），从而这个说法就可以匹配‘我想点水煮肉片’、‘我想点红烧肉’等说法

开发者添加用户说法时，可以在添加例句中手动标注，也可以使用我们平台提供一些[系统词典](#)。

### 3.4 举例

没有去标注用户说法中的关键字，这样的用户说法比较累赘不建议使用。

```
{
  "name": "eat_arr",
  "samples": [
    "介绍一下鱼香肉丝",
    "我想点鱼香肉丝",
    "鱼香肉丝怎么样"
  ]
}
```

使用语义槽取出用户说法中的关键词，开发者是想开发出点菜的功能，所以将菜品这个关键词提取出来。

```
{
  "name": "FounderIntent",
  "samples": [
    "介绍一下{菜品}",
    "我想点{菜品}",
    "{菜品}怎么样"
  ],
  "slots": [
    {
      "name": "菜品",
      "type": "menuName"
    }
  ]
}
```

## 四 语义槽

---

### 4.1 概述

语义槽（slots）就是从用户说法中提取的关键词并将其抽象化一类总称并为该类词起一个英语名称。

### 4.2 组成

slots是一个数组，该数组中存放着json字段‘name’和‘type’其中‘name’就是开发者在用户说法中标注的内容如‘菜名’、‘颜色’，‘type’就是给name取一个英文的代号如‘menuName’、‘color’。在这里的之所以要给‘name’一个对应的‘type’值，是因为在‘slot\_types’中将会用到。

### 4.3 举例

```
{
  "name": "FounderIntent",
  "samples": [
    "介绍一下{菜品}",
    "{菜品}怎么样"
  ],
  "slots": [
    {
      "name": "菜品",
      "type": "menuName"
    }
  ]
}
```

## 五 词典slot\_types

### 5.1 概述

词典是同类词的集合，如：地址、颜色、菜名等。在对话编辑中通过对词典的绑定和调用，进行重要信息的提取，语义通过词典进行理解。

### 5.2 组成

slot\_types也是一个数组，该数组中存放着json字段‘name’和‘values’。其中‘name’值对应着slots中的‘type’值如‘menu’，‘values’是一个数组用于保存该类词的多个实例如‘鱼香肉丝’、‘红烧肉’、‘宫保鸡丁’等。

### 5.3 举例

```
{
  "slot_types": [
    {
      "name": "menuName",
      "values": [
        "鱼香肉丝",
        "宫保鸡丁",
        "水煮肉片"
      ]
    }
  ],
  "intents": [
    {
      "name": "FounderIntent",
      "samples": [
        "介绍一下{菜品}",
        "{菜品}怎么样"
      ],
      "slots": [
        {
          "name": "菜品",
          "type": "menuName"
        }
      ]
    },
  ]
}
```

\*自定义的词典需要自己写slot\_types中的内容。

## 5.4 系统词典

以FX开头的是我们平台提供的系统词典，已经预置了一些词条提供给开发者直接使用。  
使用举例：

```
"slots": [  
  {  
    "name": "动物",  
    "type": "FX.animal"  
  }  
]
```

\*使用系统词典可以不用再描述slot\_types的内容。

系统词典有：

国家：FX.country

动物：FX.animal

国外城市：FX.foregincity

电视剧：FX.teleplay

综艺节：FX.vs

成语：FX.idiom

朝代：FX.dynasty

诗人：FX.poet

百家姓：FX.bjx

品牌：FX.brand

古诗：FX.potry

大学：FX.university

历史名人：FX.historicalfigure

化学元素：FX.chemical\_element

卡通动漫：FX.cartoon

电影：FX.movie

小说：FX.fiction

颜色：FX.color

商品名称：FX.item

名人：FX.celebrity。