

团 体 标 准

T/CES XXX-XXXX

人工智能开放服务能力集成规范

Open service capability integration specification for
artificial intelligence

(征求意见稿)

XXXX-XX-XX 发布

XXXX-XX-XX 实施

中国电工技术学会 发布

目 次

1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 整体结构	3
5 数据模型	4
6 业务定义	4
7 动作类型	6
7.1 REST 动作	6
7.2 消息发送动作	6
7.3 Groovy 动作	7
7.4 组合动作	7
8 消息种类	7
8.1 文本消息	7
8.2 音频消息	8
8.3 图片消息	8
8.4 视频消息	8
9 接口与认证方式	8
9.1 接口范围与内容	8
9.2 认证方式	9
10 接口说明	9
10.1 请求方式与数据格式	9
10.2 数据接口	9
10.3 语音识别接口	11
10.4 意图识别接口	14
10.5 图像识别接口	15
10.6 通用能力接口	16
10.7 临时素材库接口	17

前 言

本文件按照 GB/T1.1—2020《标准化工作导则 第1部分 标准的结构与编写》给出的规则起草。

本文件由四川中电启明星信息技术有限公司提出。

本文件由中国电工技术学会标准工作委员会能源智慧化工作组归口。

本文件起草单位：国网信息通信产业集团有限公司、四川中电启明星信息技术有限公司、国网重庆市电力公司、国网重庆市电力公司电力科学研究院。

本文件主要起草人：李强、宋卫平、邓大建、吕跃春、周孔均、赵峰、张捷、田鹏、钟加勇、黄浩、李欢欢、李炳森、田富强、尚忠玉、余文魁。

本文件为首次发布。

1 范围

本标准规定了服务能力接入的术语与定义、认证方式及服务接口说明，移动端智能交互对话机器人业务接入的术语和定义、触发的动作和支持的消息种类。

本标准适用于移动端智能交互开放服务能力的接入与应用，移动端智能交互对话机器人接入业务场景，指导业务开发人员完成业务集成工作。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T1.1-2020 标准的结构和编写

IETF RFC 2616 超文本传输协议 HTTP/1.1(Hypertext Transfer Protocol -- HTTP/1.1)

IETF RFC 5849 The OAuth 2.0 Authorization Framework

3 术语和定义

下列术语和定义适用于本文件。

3.1

对话机器人 Conversational AI

移动端智能交互对话机器人的简称，又称为聊天机器人，用来模拟人类对话或聊天的程序，识别用户输入信息的意图，并作出合理的反应。

3.2

用户 user

使用对话机器人的个人或者业务系统，在本文件中一般指个人。

3.3

业务定义 business definition

规范对话机器人能够识别与处理的业务内容，处理用户输入返回用户希望的结果。例如用户输入“明天上午九点半参加部门例会”，对话机器人调用日程系统的数据接口生成日程，给用户返回新增的日程信息。

3.4

动作 action

业务定义重要的组成部分，规定对话机器人对用户输入的直接反应，包括 REST 动作，消息发送动作等。

3.5

消息 message

用户与对话机器人交互的信息载体，包括文本消息、音频消息、图片消息、视频消息。

3.6

数据接口 data interface

对话机器人与业务系统交换数据的接口，一般是 REST 接口，采用 HTTP 协议。

3.7

业务系统 business system

满足企业或个人某些需求的系统程序，在本标准中业务系统为对话机器人提供数据接口，完成业务场景集成。

3.8

EL 表达式 EL expression

表达式语言，简称 EL(Expression Language)，是一种特殊用途的编程语言，用于访问数据对象属性、调用对象方法，真假结果计算。

3.9

REST Representational State Transfer

REST 即表述性状态传递是一种软件架构风格。它是一种针对网络应用的设计和开发方式，可以降低开发的复杂性，提高系统的可伸缩性。

3.10

URL Uniform Resource Locator

统一资源定位系统是因特网的万维网服务程序上用于指定信息位置的表示方法。

3.11

OAuth2.0 OAuth protocol

开放授权标准协议 OAuth 的第二个版本。

3.12

Groovy Groovy scripting language

Groovy 是用于 Java 虚拟机的一种敏捷的动态语言，用作纯粹的脚本语言。

3.13

SM4 SM4 block cipher algorithm

国家密码管理局颁布的对称加密算法。

3.14

Bearer Bearer Access Token

Bearer 访问令牌是 OAuth2.0 协议规定的一种访问令牌类型。

3.15

自动语音识别技术 Automatic Speech Recognition

自动语音识别技术 (Automatic Speech Recognition), 简称ASR, 是一种将人的语音转换为文本的技术。

3.16

短语音 Short Voice

短语音通常为一分钟以内的语音信息。此类语音信息通常的应用场景为实时交互场景，如：语音搜索、语音命令、语音短消息等。

3.17

长语音 Long Voice

长语音通常为超过一分钟的语音信息。此类语音信息通常不要求实时性，应用场景通常为生成字幕、会议纪要等内容。

3.18

自然语言 Natural Language

自然语言通常是指一种自然地随文化演化的语言，是人类交流和思维的主要工具。

3.19

自然语言处理 Natural Language Processing

通常而言，机器及应用程序是无法理解自然语言中蕴含的各类信息，这时就需要通过自然语言处理程序进行分析处理，从而使得计算机对其能够可读并理解。自然语言处理是一个大范围的概念，根据业务维度通常有意图识别等。

3.20

意图 Intention

指一段自然语言文本信息中，用户所表达的要达到的某种目的。

3.21

意图识别 Intention Recognition

通过技术手段识别出一段自然语言文本信息中，用户想要达到的某种目的，这种技术称之为意图识别。

3.22

光学字符识别技术 Optical Character Recognition

图像识别技术它是指对图像进行对象识别，以识别各种不同模式的目标和对像的技术。本标准中主要用于识别图像中但文本内容与坐标。

3.23

JavaScript 对象表示法 JavaScript Object Notation

JavaScript 对象表示法，即 JSON，是一种轻量级的数据交换格式。易于人阅读和编写，可以在多种语言之间进行数据交换。同时也易于机器解析和生成。

3.24

超文本传输协议 HyperText Transfer Protocol

超文本传输协议，即 HTTP，是一个简单的请求-响应协议，它通常运行在 TCP 之上。它指定了客户端可能发送给服务器什么样的消息以及得到什么样的响应。

3.25

二维码 2-Dimensional Bar Code

二维码，是用某种特定的几何图形按一定规律在平面（二维方向上）分布的、黑白相间的、记录数据符号信息的图形。最常见的二维码有 QR Code（快速响应码，即 Quick Response Code）等。

4 整体结构

用户使用移动终端即时通讯能力，向对话机器人发送消息。消息可以是一段文本，例如“查询我的日程安排”，也可以采用更便捷的语音交互方式。另外支持图片和视频消息，多运用在图像识别和视频分析等业务场景，例如设备二维码图片扫一扫查看设备详细信息运用了图像识别二维码技术。

对话机器人收到用户发送的消息，先调用意图识别模块识别用户的意图，判断用户想干什么，获得前期定义的意图编码和动作编码。对话机器人根据意图编码触发业务定义，如果未匹配上业务定义应该给用户返回一个友好的提示信息。

对话机器人在触发业务定义时，根据动作编码匹配业务定义中的动作。如果匹配上 REST 动作，根据 REST 动作规范构建 HTTP 请求对象，调用业务系统的数据接口，接口调用完成后，需要触发消息发送动作回复处理结果。例如“查询我的日程安排”业务场景中，意图识别为查询日程表，对话机器人调用日程系统的查询结果，将查询的日程表数据采用文本消息发送到用户移动终端，用户即可查看日程安排信息。

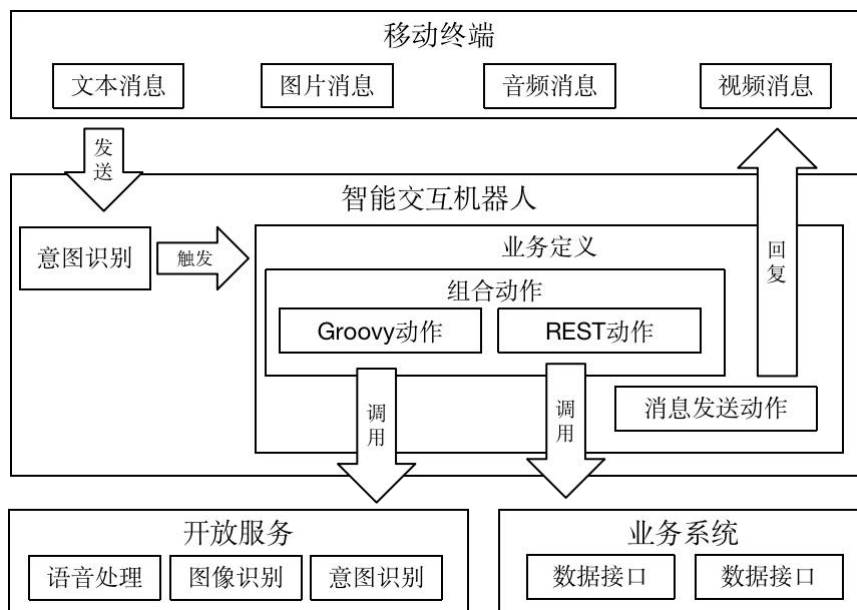


图1 整体结构图

不是所有的对话交互场景都是一问一答型的，其中相当一部分的业务场景需要多轮对话交互才能完成。示例“查询我的日程安排”也可以是多轮对话类型的，在第一轮对话时对话机器人会询问用户想查询几号的日程安排，在这一轮对话中只需要发送消息给用户。第二轮对话用户需要输入“明天”，对话机器人提取查询时间实体数据，在调用接口时将查询时间作为查询参数发送给业务系统数据接口。最后将查询的日程表数据采用文本消息发送到用户移动终端，用户即可查看日程安排信息。

5 数据模型

数据模型为业务动作的执行提供基础数据，也用于缓存持有动作执行的结果数据。数据模型是有多个键值对构成的 Map 数据结构，其中“lucas_”开头的键作为内置键，提供了以下内置键值对：

表1 数据模型内置键值对

键名称	值类型	描述
lucas_userId	字符串类型	用户身份标识主键
lucas_result	Map 或字符串类型	意图识别实体提取的结果
lucas_currentDatetime	Date 类型	当前时间
lucas_constants	Map 类型	业务定义中的常量

在业务定义中可以通过表达式语言使用数据模型中的数据，例如使用表达式“`${lucas_userId}`”获得用户身份标识的值。

6 业务定义

业务定义规范了智能交互对话中消息、意图识别、动作（业务逻辑）三者的关系和处理流程，业务定义的终极目的就是根据意图识别触发动作的执行，使用消息回复处理结果，完成业务功能场景，例如“查询我的日程安排”。业务定义文件采用 JSON 格式表示和存储。业务定义必须指定意图编码(code)和协议版本号(version)，业务定义的协议版本是 0.9.0，描述信息(description)是可选的。常量(constants)定义字面量键值对，为动作的触发执行提供数据，常量也是可选的。

表2 业务定义的意图编号和版本号

```
{
  "code": "general.查询日程",

```



```

"version": "0.9.0",
"description": "这是查询日程业务定义文件",
"constants": {
  "appld": "a7f5024fe8954f63b891581b010581a6"
}
}

```

为完成业务场景功能，一个业务定义至少包含一个动作定义，在本标准中定义了多种工作类型，满足各种业务接入场景。业务定义中的每个动作都必须指定动作编码（code）与意图识别的动作编码匹配。

表 3 业务定义的动作

```

{
  "code": "general.查询日程",
  "version": "0.9.0",
  "description": "这是查询日程业务定义文件",
  "constants": {
    "appld": "a7f5024fe8954f63b891581b010581a6",
    "queryUrl": "http://serverIP:port/query/path"
  },
  "action": [
    {
      "type": "rest",
      "definition": {
        "code": "querySchedule",
        "url": "${lucas_constants.queryUrl}",
        "method": "POST",
        "responseAttr": "scheduleData",
        "body": {
          "template": "{\"appld\": \"${lucas_constants.appld}\"}"
        }
      }
    }
  ]
}

```

业务系统开发人员依据对话交互模型和定义的意图编码与动作编码设计开发业务定义文件。移动端智能交互对话机器人不具备完善的业务二次开发能力，所以对话交互过程中涉及到的业务逻辑需要业务系统方自行开发实现，并根据 REST 动作要求完成与移动端智能交互对话机器人的业务集成。

动作除了被意图识别触发执行以外，动作可以定义出口流转，动作执行完成后根据出口流转执行指向的目标动作，实现动作的链式执行。在出口流转上需要设置 EL 表达式字符串，EL 表达式执行结果为真后，才会执行链接到工作。如果有多个出口流转的 EL 表达式执行结果都为真，目标动作会根据出口流转定义的顺序依次执行。

表 4 业务定义出口流转定义

```

{
  "transitions": [

```

```

{
  "source": "utter_search_schedule",
  "destination": "sendSuccessMessage",
  "expressionText": "![['scheduleData']['data'].isEmpty()]"
},
{
  "source": "utter_search_schedule",
  "destination": "sendEmptyDataMessage",
  "expressionText": "['scheduleData']['data'].isEmpty()"
}
]
}

```

在业务定义的“transitions”中定义动作出口流转，“source”指定源动作的编码，“destination”指定目标动作的编码，“expressionText”指定 EL 表达式字符串。

7 动作类型

动作是业务定义主要组成部分。本文件定义了 REST 动作、消息发送动作、Groovy 工作和组合动作，满足不同的业务场景。动作的执行根据人机对话的意图触发执行，与其在业务定义文件中的顺序没有关系。

7.1 REST 动作

REST 动作是对数据接口调用的定义，规定数据接口调用方式，输入参数和输出参数的处理。

表 5 REST 动作定义格式

```

{
  "type": "rest",
  "description": "查询业务数据",
  "definition": {
    "code": "token",
    "url": "业务系统提供对数据接口 URL 地址",
    "method": "POST",
    "responseAttr": "informationData",
    "body": {
      "template": ""
    }
  }
}

```

7.2 消息发送动作

消息发送动作是对消息接口的定义，用于反馈用户响应，规定了消息的类型。

表 6 消息发送动作定义格式

```

{
  "type": "message",
  "code": "sendMessage",
  "definition": {
    "template": "您没有需要处理的日程安排，可以问我一些其他的问题哦",
    "type": "text"
  }
}

```

```
}
}
```

7.3 Groovy 动作

Groovy 动作是对 Groovy 脚本执行的定义，让移动端智能交互对话机器人具备一定的编程能力，在动作触发过程中执行业务脚本，可以对数据模型中对数据进行修改、添加键值对数据等。

表 7 Groovy 动作定义格式

```
{
  "type": "groovy",
  "code": "modifySchedule",
  "definition": {
    "content": "Groovy 脚本的内容"
  }
}
```

7.4 组合动作

组合动作可以将 REST 动作、消息发送动作、Groovy 动作灵活的组合在一起作为一个动作被触发执行。被组合的动作根据定义的先后顺序依次执行。例如在“查询我的日程安排”多轮对话业务场景中，在执行 REST 动作调用数据接口之前可以先执行 Groovy 动作修改查询参数。

表 8 组合动作定义格式

```
{
  "type": "delegate",
  "code": "modifySchedule",
  "action": [
    {
      "type": "groovy",
      "definition": {
        "content": "Groovy 脚本的内容"
      }
    },
    {
      "type": "rest",
      "definition": {
        "url": "${lucas_constants.queryUrl}"
      }
    }
  ]
}
```

8 消息种类

消息是用户与移动端智能交互对话机器人之间沟通交流的信息载体，本标准规定了 4 个消息类型：文本消息、音频消息、图片消息、视频消息。消息体采用 JSON 格式表示，其中音频消息、图片消息、视频消息涉及到的文件需要通过临时素材库中转。

8.1 文本消息

表 9 文本消息定义格式

```
{
  "type": "text",
  "content": "文本内容",
  "touser": "消息接受者身份标识",
  "fromUserId": "消息发送者者身份标识"
}
```

8.2 音频消息

表 10 音频消息定义格式

```
{
  "type": "voice",
  "mediaId": "素材 ID，通过素材库上传接口获得",
  "touser": "消息接受者身份标识",
  "fromUserId": "消息发送者者身份标识"
}
```

8.3 图片消息

表 11 图片消息定义格式

```
{
  "type": "image",
  "mediaId": "素材 ID，通过素材库上传接口获得",
  "touser": "消息接受者身份标识",
  "fromUserId": "消息发送者者身份标识"
}
```

8.4 视频消息

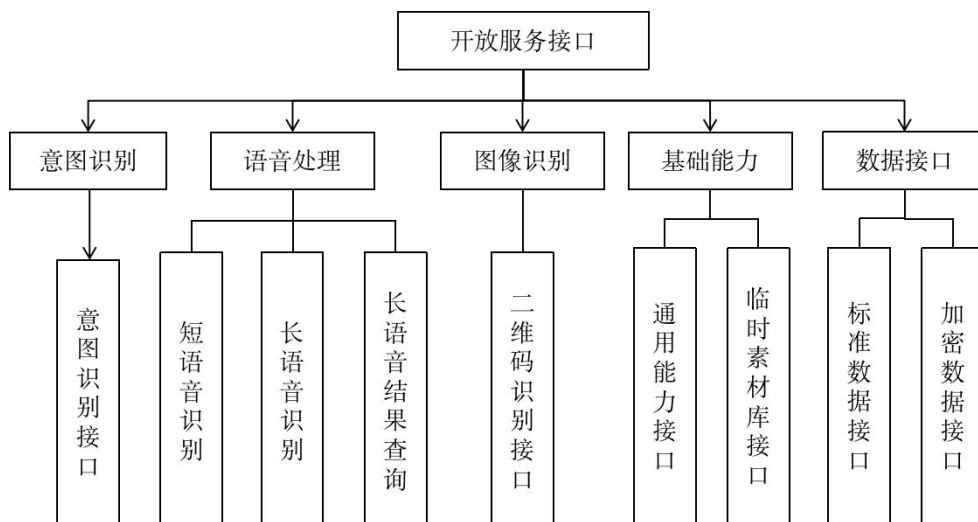
表 12 视频消息定义格式

```
{
  "type": "video",
  "mediaId": "素材 ID，通过素材库上传接口获得",
  "touser": "消息接受者身份标识",
  "fromUserId": "消息发送者者身份标识"
}
```

9 接口与认证方式

9.1 接口范围与内容

开放服务平台提供的服务接口，应包括但不限于以下能力范围：意图识别、语音处理、图像识别、基础能力。其中语音处理又包括短语音识别、长语音识别、长语音结果查询，图像识别又包括二维码识别接口，通用能力接口是其他开放服务的底层接口。如图一所示。



9.2 认证方式

本标准的开放服务接口采用 OAuth2.0 授权协议保证接口的安全访问，接口使用方在调用接口前必须获得访问令牌，调用接口时需要在 HTTP 请求头中添加访问令牌。开放服务接口采用国密 SM4 的 CBC 模式加密传输的方式保障接口数据在不安全网络中数据传输的保密性。下面正文内容将详细说明接口安全访问控制和数据加密传输的具体细节。

10 接口说明

10.1 请求方式与数据格式

本文件中接口采用何种请求方式、何种数据类型作为数据传输格式，均以接口说明为准。

10.2 数据接口

智能交互对话机器人与业务系统之间主要采用数据接口进行业务场景集成，完成数据交换。

10.2.1 标准数据接口

对话机器人调用下表定义的接口发送给业务系统。

表 13 数据接口定义

接口名称	业务集成数据接口		
请求地址	接口地址由业务系统定义并提供		
HTTP 方法	POST	媒体类型	application/json;charset=UTF-8
请求头	Authorization	Bearer 访问令牌	
请求	<pre> { "属性 1": "字符串类型", "属性 2": 数值类型, "属性 3": "日期时间类型 yyyy-MM-dd HH:mm:ss" } </pre>		
响应	正常 200	<pre> { "message": "处理结果，可能会直接发送给用户手机端", "type": "消息的类型： text textcard", "subject": "消息主题名称", "url": "URL 地址, textcard 类型消息需要指定用户点击跳转的地址" } </pre>	

	异常 500	{ "message": "错误中文描述", "status": "HTTP 状态码" }
	异常 401	{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }
备注	<p>请求的主体采用 JSON 格式，由业务系统定义包括哪些字段。对话机器人负责收集定义的字段值并发送给业务系统。</p> <p>正常 200 响应主体如果执行了消息的类型，则将“message”发送给用户手机端。</p> <p>错误 500 响应主体中属性“message”可能会直接发送到用户手机端。状态码“status”与 HTTP 响应的状态码保持一致，例如 400、401、500 等，一般业务错误状态码为 500。</p>	
安全保障	数据接口支持 OAuth2.0 授权协议实现接口的授权访问，在 HTTP 请求头中添加 Authorization 请求头，使用 bearer 类型的访问令牌。	
示例	<p>下面演示对话机器人与日程助手集成新增日程场景。</p> <ol style="list-style-type: none"> 1. 从业务场景出发，新增日程需要知道主题和开始时间两个属性； 2. 对话机器人需要从用户对话中识别出日程主题和开始时间。并分别赋值给开始时间“startTime”和主题“title”两个属性。其中两个属性名称由对话机器人与日程助手一起约定； 3. 用户通过对话机器人，发送“明天上午 9 点半在 1205 会议室参加部门周例会”，识别出开始时间是“2021-12-25 09:30:00”，主题是“在 1205 会议室参加部门周例会”。在这个过程中对话机器人可能会与用户进行多轮对话； 4. 对话机器人成功收集到日程主题和开始时间后，封装成 JSON 对象： <pre> { "startTime": "2021-12-25 09:30:00", "title": "在 1205 会议室参加部门周例会" } </pre> <p>调用日程业务集成接口，将 JSON 对象发送给日程应用，日程完成自己的业务后，返回 200 响应数据：</p> <pre> { "message": "日程新增成功，您可以点击继续完善日程信息。", "type": "textcard", "subject": "日程新增通知", "url": "http://schedule.sgcc.com.cn/schedule/#update?id=日程主键 ID" } </pre> <ol style="list-style-type: none"> 5. 对话机器人收到日程返回的响应后，给用户发送日程新增通知； 6. 用户在对话机器人消息页面可以点击“日程新增通知”，跳转到 URL 地址“http://schedule.sgcc.com.cn/schedule/#update?id=日程主键 ID”界面，其中日程主键 ID 由接口成功返回获得。 7. 如果接口调用失败，返回 500 错误消息： 	

	<pre>{ "message": "业务系统需要提示给用户的可读信息", "status": 500 }</pre> <p>默认将属性 <code>message</code> 的值返回给用户手机端。</p> <p>到此就完成了对话机器人日程新增场景集成。</p>
--	---

10.2.2 加密数据接口

在非安全的网络环境中调用数据接口,在访问控制的基础上需要额外的安全措施保障数据的安全。下表的数据接口规范在上面标准数据接口的基础上采用国密 SM4 算法对数据进行加密传输。

表 14 数据接口定义

接口名称	业务集成数据接口,支持 SM4 加密传输		
请求地址	接口地址由业务系统定义并提供		
HTTP 方法	POST	媒体类型	application/json;charset=UTF-8
请求头	Authorization	Bearer 访问令牌	
请求	<pre>{ "cryptoType": "SM4", "data": "标准数据接口请求 JSON 字符串经过 SM4 加密后的字符串数据" }</pre>		
响应	正常 200	<pre>{ "cryptoType": "SM4", "data": "标准数据接口响应 JSON 字符串经过 SM4 加密后的字符串数据" }</pre>	
	异常 500	<pre>{ "message": "错误中文描述", "status": "HTTP 状态码" }</pre>	
	异常 401	<pre>{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }</pre>	

10.3 语音识别接口

语音识别接口用于将语音文件中的语音信息转换为文本信息,主要功能应包括短语音文件识别、长语音识别,以及长语音识别任务结果查询。

语音识别区分短语音与长语音的主要原因是它们各自的业务场景通常对识别接口有着不同的要求。例如:短语音识别接口与长语音识别接口相比,前者需要尽快地获得识别结果,后者通常需要更长的耗时且时长不确定,所以还需要一个专有的接口用以查询识别的结果。

10.3.1 短语音识别

短语音识别的识别对象为一分钟以内的语音文件,该语音文件的来源通常是语音搜索、语音命令、语音消息等业务场景,这些场景下需要识别接口能够快速响应,利于业务能够更快地进行下一步的业务逻辑。

表 15 短语音识别接口定义

接口名称	根据语音文件，提供对短语音进行识别的接口，返回识别的结果		
请求地址	http://serverurl/open/v1/asr/convert		
HTTP 方法	POST	媒体类型	multipart/form-data
请求头	Authorization	Bearer 访问令牌	
请求	file 语音文件		
响应	正常 200	<pre>{ "cryptoType": "SM4", "data": "标准数据接口响应字符串经过 SM4 加密后的字符串数据" }</pre> 解密后数据是字符串类型的文本，例如“提醒我明早十点开会”	
	异常 500	<pre>{ "message": "错误中文描述", "status": "HTTP 状态码", "code": "错误编码" }</pre>	
	异常 401	<pre>{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }</pre>	
备注	请求的主体采用表单提交上传语音文件，属性名称是“file”。 正常 200 响应主体是 JSON 格式，实际数据采用 SM4 加密，有属性“cryptoType”制定加密类型。 错误 500 响应主体采用明文传输未加密，描述错误原因。状态码“status”与 HTTP 响应的状态码保持一致，例如 400、401、500 等，一般业务错误状态码为 500。		
错误编码	10006 获取某个参数失败 10043 音频解码失败 10700 系统内部错误		

10.3.2 长语音识别

长语音识别的识别对象为一分钟以上的语音文件，该语音文件的来源通常是生成字幕、会议纪要等业务场景，这些场景的识别过程耗时较长，不需要识别接口能立马返回识别结果，只需要后续进行结果查询即可。

表 16 长语音识别接口定义

接口名称	根据语音文件，提供对长语音进行识别的接口，返回当前识别请求的编号，用于识别结果的查询		
请求地址	http://serverurl/open/v1/lfasr/convert		
HTTP 方法	POST	媒体类型	multipart/form-data
请求头	Authorization	Bearer 访问令牌	
请求	file 语音文件		
响应	正常 200	<pre>{ "cryptoType": "SM4",</pre>	

		<pre>"data": "标准数据接口响应字符串经过 SM4 加密后的字符串数据" }</pre>
		解密后数据是字符串类型的识别任务编号，示例： “111111111111111111”
	异常 500	<pre>{ "message": "错误中文描述", "status": "HTTP 状态码", "code": "错误编码" }</pre>
	异常 401	<pre>{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }</pre>
备注	<p>请求的主体采用表单提交上传语音文件，属性名称是“file”。</p> <p>正常 200 响应主体是 JSON 格式，实际数据采用 SM4 加密，有属性“cryptoType”制定加密类型。</p> <p>错误 500 响应主体采用明文传输未加密，描述错误原因。状态码“status”与 HTTP 响应的状态码保持一致，例如 400、401、500 等，一般业务错误状态码为 500。</p>	
错误编码	<p>10006 获取某个参数失败</p> <p>10043 音频解码失败</p> <p>10700 系统内部错误</p>	

10.3.3 长语音识别结果查询

长语音识别结果查询接口用于查询长语音识别任务的识别结果。

表 17 长语音识别结果查询接口定义

接口名称	根据识别任务编号，提供对长语音识别任务结果查询的接口，返回识别任务的结果		
请求地址	http://serverurl/open/v1/lfasr/convert/result		
HTTP 方法	GET	媒体类型	
请求头	Authorization	Bearer 访问令牌	
请求	taskId 任务 ID		
响应	正常 200	<pre>{ "cryptoType": "SM4", "data": "标准数据接口响应字符串经过 SM4 加密后的字符串数据" }</pre>	
		解密后数据是 JSON 格式： <pre>{ "status": "状态 1 处理中 2 处理完成", "text": "语音文件转换获得的文本字符串" }</pre>	
	异常 500	<pre>{</pre>	

		<pre>"message": "错误中文描述", "status": "HTTP 状态码", "code": "错误编码" }</pre>
	异常 401	<pre>{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }</pre>
备注	<p>查询参数“taskId”是语音转换的任务编号。</p> <p>正常 200 响应主体是 JSON 格式,实际数据采用 SM4 加密,有属性“cryptoType”制定加密类型。</p> <p>错误 500 响应主体采用明文传输未加密,描述错误原因。状态码“status”与 HTTP 响应的状态码保持一致,例如 400、401、500 等,一般业务错误状态码为 500。</p>	
错误编码	<p>10006 获取某个参数失败</p> <p>10700 系统内部错误</p> <p>10801 任务不存在</p>	

10.4 意图识别接口

意图识别接口主要是用于将用户以自然语言形式的文本信息,转换为应用程序代码可识别的、包含了该用户想要表达的意图信息的结构化数据,从而应用程序可通过该结构化数据进行进一步的分析、判断、处理、给出用户需要的解决方案等诸多业务处理。

10.4.1 意图识别

表 18 意图识别接口定义

接口名称	根据请求参数,提供意图识别的接口,返回意图识别结果		
请求地址	http://serverurl/open/v1/semantic/analyze		
HTTP 方法	POST	媒体类型	application/json;charset=UTF-8
请求头	Authorization	Bearer 访问令牌	
请求	<pre>{ "fromUserId": "用户编号", "msgType": "text", "textContext": "意图识别的文本内容" }</pre>		
响应	正常 200	<pre>{ "cryptoType": "SM4", "data": "标准数据接口响应字符串经过 SM4 加密后的字符串数据" }</pre>	
		<p>解密后数据是 JSON 格式:</p> <pre>{ "user_id": "用户编号", "intent": "意图编码", "result": {} #意图识别提取的实体属性,不同的意图有不同</pre>	

		的属性 }
	异常 500	{ "message": "错误中文描述", "status": "HTTP 状态码", "code": "错误编码" }
	异常 401	{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }
备注	<p>请求的主体采用 JSON 数据格式，属性“fromUserId”指定用户身份标识，属性“msgType”指定识别的消息类型，这里是固定值“text”，属性“textContext”指定意图识别的文本内容。</p> <p>正常 200 响应主体是 JSON 格式，实际数据采用 SM4 加密，有属性“cryptoType”制定加密类型。</p> <p>错误 500 响应主体采用明文传输未加密，描述错误原因。状态码“status”与 HTTP 响应的状态码保持一致，例如 400、401、500 等，一般业务错误状态码为 500。</p>	
错误编码	<p>10006 获取某个参数失败</p> <p>10700 系统内部错误</p>	

10.5 图像识别接口

对于计算机及应用程序来说，图像仅仅是一个二进制文件，它们不能识别出图像中所蕴含的信息。图像识别接口用于识别图像中对象信息，并结构化对象信息进行返回。

主要功能应包括二维码识别接口。

10.5.1 二维码识别接口

表 19 二维码识别接口定义

接口名称	根据请求参数，提供二维码识别的接口，返回二维码识别的结果		
请求地址	http://serverurl/open/v1/ocr/qrcode		
HTTP 方法	POST	媒体类型	multipart/form-data
请求头	Authorization	Bearer 访问令牌	
请求	file 图像文件，支持 png、jpg、jpeg 格式		
响应	正常 200	<pre>{ "cryptoType": "SM4", "data": "标准数据接口响应字符串经过 SM4 加密后的字符串数据" }</pre> <p>解密后数据是二维码识别后的字符串文本内容。 示例：020020000000000000449931</p>	
	异常 500	<pre>{ "message": "错误中文描述", "status": "HTTP 状态码", }</pre>	

		<pre>"code": "错误编码" } </pre>
	异常 401	<pre>{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 } </pre>
备注	<p>请求的主体采表单提交上传图像文件，支持 png、jpg、jpeg 格式，属性名称是“file”。</p> <p>正常 200 响应主体是 JSON 格式，实际数据采用 SM4 加密，有属性“cryptoType”制定加密类型。</p> <p>错误 500 响应主体采用明文传输未加密，描述错误原因。状态码“status”与 HTTP 响应的状态码保持一致，例如 400、401、500 等，一般业务错误状态码为 500。</p>	
错误编码	<p>10006 获取某个参数失败</p> <p>10700 系统内部错误</p>	

10.6 通用能力接口

通用能力同步调用接口定义，作为开放服务平台底层通用接口，可以调用大部分支持同步调用的服务能力。

表 20 通用能力接口定义

接口名称	根据服务能力编号与参数调用开放平台的服务能力，返回能力处理结果		
请求地址	http://serverurl/open/v1/function/common		
HTTP 方法	POST	媒体类型	multipart/form-data
请求头	Authorization	Bearer 访问令牌	
请求	<pre>{ "code": "服务能力编码，由开放服务能力平台制定", "params": {} } </pre>		
响应	正常 200	<pre>{ "cryptoType": "SM4", "data": "标准数据接口响应字符串经过 SM4 加密后的字符串数据" } </pre> <p>解密后数据是二维码识别后的字符串文本内容。</p> <pre>{ "data": { "value": "02002000000000000449931" }, "code": "qrcode" } </pre>	
	异常 500	<pre>{ "message": "错误中文描述", "status": "HTTP 状态码", "code": "错误编码" } </pre>	

		}
	异常 401	{ "error_message": "非法的访问令牌", "error": "invalid_request", "status": 401 }
备注	请求的主体采表单提交上传语音文件，属性名称是“file”。 正常 200 响应主体是 JSON 格式，实际数据采用 SM4 加密，有属性“cryptoType”制定加密类型。 错误 500 响应主体采用明文传输未加密，描述错误原因。状态码“status”与 HTTP 响应的状态码保持一致，例如 400、401、500 等，一般业务错误状态码为 500。	
错误编码	10006 获取某个参数失败 10700 系统内部错误	

10.7 临时素材库接口

临时素材库是用于缓存音频、视频、图片等文件的服务，支持上传和下载，素材库中的文件可以缓存 2 天。临时素材库主要用于配套音频消息、图片消息和视频消息一起使用。

表 21 上传临时素材接口定义

接口名称	上传临时素材文件		
请求地址	/open/v1/media/upload?type=TYPE		
HTTP 方法	POST	媒体类型	multipart/form-data
请求	"media": "素材文件"		
响应	{ "errcode": 0, "errmsg": "ok", "type": "image", "media_id": "1G6nrLmr5EC3MMb_-zK1dD" }		
备注	type 媒体文件类型，分别有图片（image）、语音（voice）、视频（video） media_id 媒体文件上传后获取的唯一标识，3 天内有效 errcode 出错返回码，为 0 表示成功，非 0 表示调用失败 errmsg 返回码提示语 所有上传的文件大小必须大于 5 个字节，图片（image）：2MB，支持 JPG、PNG 格式，语音（voice）：2MB，播放长度不超过 60s，仅支持 AMR 格式，视频（video）：20MB，仅支持 MP4 格式。		