

中科院声学所mrcp

代码模块接口设计

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V1.0 | 段志亮 |  | 2021-11-25 | 创建文件 |
|  |  |  |  |  |

目录

[1.概述 3](#_Toc88752311)

[**1.1 编写目的** 3](#_Toc88752312)

[2.总体设计 4](#_Toc88752313)

[2.1 总体需求 4](#_Toc88752314)

[3.mrcp识别服务模块接口设计 5](#_Toc88752315)

[3.1网络通信模块 5](#_Toc88752316)

[3.1.2 网络通信接口 6](#_Toc88752317)

[3.2对象管理模块 6](#_Toc88752318)

[3.2.1对象管理接口 7](#_Toc88752319)

[3.3 流程控制模块 7](#_Toc88752320)

[3.3.1 资源分配接口 8](#_Toc88752321)

[3.3.2 语音输入通知接口 9](#_Toc88752322)

[3.3.3 检测到有效通知接口 9](#_Toc88752323)

[3.3.4 语音处理接口 10](#_Toc88752324)

[4.代码目录 11](#_Toc88752325)

[4.1 代码目录结构 11](#_Toc88752326)

[4.2 主要代码文件说明 12](#_Toc88752327)

[4.2.1数据封装及解析模块 12](#_Toc88752328)

[4.2.2http请求逻辑模块 12](#_Toc88752329)

[4.2.3 流程控制模块 13](#_Toc88752330)

[4.3 代码编译 13](#_Toc88752331)

# 1.概述

1

**1.1 编写目的**

本文档主要描述mrcp服务端内部接口设计及接口中参数的数据结构，本文档的读者对象为mrcp系统的开发人员、测试人员、系统维护人员，通过本文档能够从总体上了解mrcp内部的数据流向及处理方式。

本说明给出mrcp服务的接口设计说明，包括接口安全、版本兼容性、数据格式、服务端的异常处理等。

目的在于：

* 为开发人员提供依据；
* 为代码修改、维护提供条件；

# 2.总体设计

2

## 2.1 总体需求

接口需求：

1，接口参数简单明了，参数有实际意义；

2，接口功能定义明确；

3，接口的数据特性明确；

4，接口的规格和技术要求明确；

5，接口程序的数据处理逻辑清晰；

# 3.mrcp识别服务模块接口设计

3

mrcp识别服务包含对象管理模块、网络通信模块和流程控制模块。

模块间关系如下：



## 3.1网络通信模块

封装http请求方法，网络请求提供调用接口；

### 3.1.2 网络通信接口

**int** **GetJsonFromPostMethod**(std::string URL,

std::string jsonObj,

std::string &str)

函数功能：post方法请求http服务端，获取响应信息。

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| url | 字符串 | 请求路径 |
| jsonObj | json字符串 | 请求参数 |
| str | 字符串 | 响应信息 |

## 3.2对象管理模块

线程管理模块负责在服务启动时，创建一定数量的http客户端对象；有新的会话接入时，从创建好的http客户端对象中分配一个对象；当会话结束时，回收分配的对象。

### 3.2.1对象管理接口

**bool** **HttpClientManager::AllocAllHttpClients**(**int** num)

函数功能：该函数主要创建一定数量的http客户端对象。

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| num | 整数 | 对象数量 |

**HttpClient \*HttpClientManager::AllocHttpClient**()

函数功能：该函数为新接入的会话分配对象。

**bool** **HttpClientManager::FreeHttpClient**(HttpClient \*httpclient)

函数功能：该函数回收结束会话对象。

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| httpclient | 对象指针 | 待回收对象 |

## 3.3 流程控制模块

流程控制模块主要功能为处理会话信令，根据会话请求设置参数、分配会话资源、检测语音流和请求ASR服务端进行语音识别，返回识别信息等。

会话控制流程图如下：



### 3.3.1 资源分配接口

**static** apt\_bool\_t **RecogMrcpRecognize**(mrcp\_engine\_channel\_t \*channel,

mrcp\_message\_t \*request,

mrcp\_message\_t \*response)

函数功能：该函数接收客户端识别请求，初始化资源；

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| channel | 指针 | 存放会话通道信息 |
| request | 指针 | 存放请求信息 |
| response | 指针 | 存放响应信息 |

### 3.3.2 语音输入通知接口

**static** apt\_bool\_t **RecogMrcpTimersStart**(mrcp\_engine\_channel\_t \*channel,

mrcp\_message\_t \*request,

mrcp\_message\_t \*response)

函数功能：该函数接收客户端语音输入通知请求；

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| channel | 指针 | 存放会话通道信息 |
| request | 指针 | 存放请求信息 |
| response | 指针 | 存放响应信息 |

### 3.3.3 检测到有效通知接口

**static** apt\_bool\_t **RecogStartOfInput**(AsrRecogChannel \*recog\_channel)

函数功能：该函数通知客户端检测到有效语音；

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| recog\_channel | 指针 | 通知会话信息 |

### 3.3.4 语音处理接口

**static** apt\_bool\_t **RecogVoiceStream**(mpf\_audio\_stream\_t \*stream,

**const** mpf\_frame\_t \*frame)

函数功能：该函数检测语音流，请求asr识别服务，获取识别结果；

|  |  |  |
| --- | --- | --- |
| 接口参数 | 数据格式 | 参数说明 |
| stream | 指针 | 请求指令信息 |
| frame | 指针 | 语音帧信息 |

# 4.代码目录

4

## 4.1 代码目录结构

.

├── epd //vad 处理目录

├── include //头文件目录

│   ├── AsrCommonMethod.h

│   ├── AsrDefine.h

│   ├── ComDefine.h

│   ├── EmotionKeyword.h

│   ├── Encode.h

│   ├── gatewayrecog.h

│   ├── HighPass.h

│   ├── HttpClient.h

│   ├── HttpClientManager.h

│   ├── rapidjson //json解析

│   └── WavHeaderUtil.h

├── Makefile.am

├── src

   ├── AsrCommonMethod.cpp //数据封装

   ├── Encode.cpp //base64编解码

   ├── gatewayrecog.cpp //会话流程控制

   ├── gatewayrecog\_impl.cpp

   ├── HighPass.cpp

   ├── HttpClient.cpp //请求会话逻辑

   ├── HttpClientManager.cpp //http对象管理文件

   └── WavHeaderUtil.cpp //保存语音

## 4.2 主要代码文件说明

### 4.2.1数据封装及解析模块

AsrCommonMethod.cpp该cpp文件中主要包含请求ASR识别服务的数据封装接口和解析响应信息接口。

### 4.2.2http请求逻辑模块

HttpClient.cpp该cpp定义了和ASR交互的流程控制及结果解析。

### 4.2.3 流程控制模块

gatewayrecog.cpp该cpp功能为初始化资源，调用其他模块控制每个线程的处理流程，异常出现时能正常释放会话，新的会话进来后能正常处理。

## 4.3 代码编译

1，编译依赖库

执行命令 sh build-dep-libs.sh，编译出的库在当前用户的local目录下；

2，编译mrcp可执行文件

执行 sh bootstrap命令，在当前用户的local目录下生成unimrcp目录，在此unimrcp的bin目录下添加启动脚本，配置服务ip和端口，即可启动服务。