

【Rtos-SDK】智能体接口文档介绍

一、背景介绍

本文档描述AI智能体Rtos接口文档，本SDK是用于正式对接版本；

1.1 账号信息

正式版本app_id 需要在百度智能云控制台开通RTC服务申请；

地址：<https://cloud.baidu.com/doc/RTC/index.html>

1.2 测试环境地址

沙盒地址：<https://ai.agent.kaywang.cn/>

正式环境地址后续RD给出

创建智能体：</api/v1/aiagent/generateAIAgentCall>

【Demo演示环境客户端调用创建智能体，正式环境需要业务服务器调用该接口】

请求参数，**config 支持是json string; 当前支持配置如下；**

1	名称	类型	是否必填	示例值	描述
2	app_id	String	是	xxx	BRTC应用ID
3	quick_start	bool	否	true	快速启动（不需要调用start指是true
4	config	String	否	{ "lang": "zh", "location": "北京市海淀区", "tts_sayhi": "主动打招呼的话，"你好！", "role": "你是旅游达人" }	ai instance配置 lang: zh 中文， en 英文 location: 位置信息 tts_sayhi: 联通后首次播报 role: 人设

1.3 SDK下载地址

1	芯片平台	sdk下载地址
2	esp32-s3	https://brtc-sdk.cdn.bcebos.com/rtos-esp32%2Fchatagent%2FBRTC.RTOS.SDK.ESP32S3.V3.0.8B01.zip

二、详细SDK接口封装

2.1 初始化设置回调

</> 生成BaiduChatAgentEngineJava

```
1  /* 构造智能体对象*/
2  typedef struct AgentEngineParams {
3      char agent_platform_url[256];          // 登陆agent智能体中心的地址
4      char config[256];                     // 配置文件路径
5      char appid[64];                       // 应用ID
6      char cer[256];                       // 默认证书路径
7      char token[256];                     // 鉴权token
8      char workflow[64];                   // 工作流类型，默认VoiceChat
9      char llm[64];                        // 大模型类型
10     char lang[16];                       // 语言类型，默认zh
11     char cid[64];                        // 通讯id
12     long long instance_id;                // 实例ID，服务器下发，当前默认即可
13     int level_voice_interrupt;            // 本地音频打断能量值，默认80
14     int AudioInChannel;                  // 音频输入通道数，默认1
15     int AudioInFrequency;                // 音频输入采样率，默认16000
16     char AudioIncodecType[16];           // 音频输入的格式，默认是pcm
17     char AudioOutcodecType[16];          // 音频播放的格式，默认是g711u
18     int VideoWidth;                     // 视频宽度
19     int VideoHeight;                    // 视频高度
20     bool verbose;                       // 是否开启详细日志，默认关闭
21     bool enable_voice_interrupt;         // 是否开启本地音频打断，默认关闭
22     bool enable_local_agent              // true: 本地请求创建智能体，false: 服务端请求
        创建智能体
23 } AgentEngineParams;
24
25 BaiduChatAgentEngine* baidu_create_chat_agent_engine(const BaiduChatAgentEvent*
    events);
26 /**
```

```
27 * @brief 初始化 engine, 内部会进行异步登录等操作
28 * @param engine engine 实例指针
29 * @param param 登录相关参数
30 * @return 0 表示初始化成功, 非 0 表示失败
31 */
32 int baidu_chat_agent_engine_init(BaiduChatAgentEngine* engine, const
    AgentEngineParams* param);
```

2.2 开启对话

</>

Java

```
1 /**
2 * @brief 发起呼叫 (上层在 init 后调用)
3 * @param engine engine 实例指针
4 */
5 void baidu_chat_agent_engine_call(BaiduChatAgentEngine* engine);
```

2.3 发送音频

</>

Java

```
1 /**
2 * @brief 向 AI agent 发送音频数据
3 * @param engine engine 实例指针
4 * @param data 音频数据
5 * @param len 数据长度
6 */
7 void baidu_chat_agent_engine_send_audio(BaiduChatAgentEngine* engine, const
    uint8_t* data, size_t len);
```

2.4 发送文本

</>

Java

```
1 /**
2 * @brief 向 AI agent 发送文本消息
3 * @param engine engine 实例指针
4 * @param text 文本消息字符串
```

```
5 */
6 void baidu_chat_agent_engine_send_text(BaiduChatAgentEngine* engine, const char*
    text);
```

2.5 发送FunctionCall结果

</>

Plain Text

```
1 /**
2  * 发送functionCall 结果
3  * @param id 唯一标识
4  * @param result 结果 , 例如: {"result":"ok"}
5  * */
6 void baidu_chat_agent_engine_send_functioncall_result(BaiduChatAgentEngine*
    engine, const char* id, const char* result);
```

2.6 主动打断

</>

Java

```
1 /**
2  * @brief 中断呼叫 (例如发送中断信号给对端)
3  * @param engine engine 实例指针
4  * */
5 void baidu_chat_agent_engine_interrupt(BaiduChatAgentEngine* engine);
```

2.7 结束对话和销毁

</>

Java

```
1 /**
2  * @brief 销毁 engine 实例并释放所有资源
3  * @param engine engine 实例指针
4  * */
5 void baidu_chat_agent_engine_destroy(BaiduChatAgentEngine* engine);
```

2.8 事件回调

</>

Java

```
1    /**
2     * 状态变化
3     * @param errCode 错误码
4     * @param errMsg 错误原因
5     */
6    void (*onError)(int errCode, const char* errMsg);
7
8
9    /**
10   * 状态变化
11   * @param state 状态
12   * 智能体流程开始 {@link Constants.CallState#ON_CALL_BEGIN }
13   * 智能体流程结束 {@link Constants.CallState#ON_CALL_END }
14   */
15   void (*onCallStateChange)(AGentCallState state);
16
17
18   /**
19   * 链接状态变化
20   * @param state 状态 {@link Constants.ConnectionState }
21   * */
22   void (*onConnectionStateChange)(AGentConnectState state);
23
24
25
26   /**
27   * 用户端音频ASR结果
28   * @param text 文本
29   * @param isFinal 标记是否最终结果
30   * */
31   void (*onUserAsrSubtitle)(const char* text, bool isFinal);
32
33
34
35   /**
36   * functionCall
37   * @param id functionCall 单次唯一标识
38   * @param params json参数, 例如: {"function_name":"phone_call","parameter_list":
39   [{"called_number":"1891017000"}, {"called_name":"我的父亲"}
40   * */
41   void (*onFunctionCall)(const char* id, const char* params);
```

```
42
43  /**
44   * 智能体传给端侧的音频数据
45   * @param data 数据指针
46   * @param len 数据长度
47   * */
48 void (*onAudioData)(const uint8_t* data, size_t len);
49
```

三、SDK调用流程图

