

0. BRTC实时交互智能体接入指引文档V1.2

目录

- 申请智能体AppID
- 智能体服务端API 接入指引
- 客户端接入指引
 - 微信小程序接入指引
 - 3.2. 安卓手机接入指引
 - 3.3. 苹果手机接入指引
 - 3.4. H5网页接入指引
 - 3.5. rtos sdk接入指引
- 4.接入曦灵数字人的说明

1. 申请智能体AppID

需要开通百度云账号，在百度云控制台上申请智能体AppID

目前在实时音视频RTC 产品中申请AppID

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

获得AppID 后开始进入第二步， BRTC 采用预付费方式，需要对BRTC进行充值后使用。

2. 智能体服务端API 接入指引

接入文档：**AI实时互动服务管理对外API.pdf**

[https://brtc-](https://brtc-sdk.cdn.bcebos.com/ai/agent/api/AI%E5%AE%9E%E6%97%B6%E4%BA%92%E5%8A%A8%E6%9C%8D%E5%8A%A1%E7%AE%A1%E7%90%86%E5%AF%B9%E5%A4%96API.pdf)

[sdk.cdn.bcebos.com/ai/agent/api/AI%E5%AE%9E%E6%97%B6%E4%BA%92%E5%8A%A8%E6%9C%8D%E5%8A%A1%E7%AE%A1%E7%90%86%E5%AF%B9%E5%A4%96API.pdf](https://brtc-sdk.cdn.bcebos.com/ai/agent/api/AI%E5%AE%9E%E6%97%B6%E4%BA%92%E5%8A%A8%E6%9C%8D%E5%8A%A1%E7%AE%A1%E7%90%86%E5%AF%B9%E5%A4%96API.pdf)

服务器地址：<https://rtc-aiagent.baidubce.com>

API config中的模型选择： config: {llm, llm_url},

API 简要介绍：

- **1. 创建智能体：/api/v1/aiagent/generateAIAgentCall**
 - 参数格式：{app_id: AppID, quick_start: true, config: "{}"};
- **2. 停止智能体：/api/v1/aiagent/stopAIAgentInstance**
 - 参数格式：{ app_id: AppID, ai_agent_instance_id: AgentID};
- **3. 打断功能：/api/v1/aiagent/interrupt**
 - 参数格式：{ app_id: AppID, ai_agent_instance_id: AgentID};

3. 客户端接入指引

3.1. 微信小程序接入指引

微信小程序接入SDK：<https://brtc-sdk.cdn.bcebos.com/ai/agent/miniapp/BRTC.Agent.MiniApp.SDK.V1.0.5.zip>

文档详见SDK压缩包中。

</> 微信小程序API简要介绍: JavaScript

```
1 1. 初始化
2 const brtc_agent = require('./agent.js');
3 var mAgent = brtc_agent.Agent;
4
5 2. 开始通话
6 // 启动Agent
7 var cfg = {
8     llm: 'LLMxxx',
9     llm_url: ''
10 };
11 mAgent.Start({
12     appid: 'AppIdxxx',
13     cfg: cfg,
14     success: function (pushurl, agentId) {
15     },
16     error: function (error) {
17     },
18     onmessage: function (msg) {
19         console.log('onmessage id: ' + msg.id + ' data: ' + msg.data);
20     },
21     remotevideocoming: function (id, display, attribute, pullurl) {
22     },
23 });
24
25 3. 发送文本消息
26 mAgent.sendMessageToUser('[T]:你好');
27
28 4. 结束通话
29 mAgent.Stop();
```

3.2. 安卓手机接入指引

安卓SDK：<https://brtc-sdk.cdn.bcebos.com/ai/agent/android/Android.ChatAgent.v1.0.25.zip>

文档详见SDK压缩包中。

</> 安卓SDK函数简介

Java

```
1 1. 初始化初始化AIEngine
2 public static AIEngineImpl init(Context context, AIEngineParams
  params)
3 2. 开启对话
4 public abstract void call(String token, long instanceId);
5 3. 结束对话
6 public abstract void hangup();
7 4. 发送文本
8 public abstract void setTextToAIEngine(String text);
```

3.3. 苹果手机接入指引

iOS SDK: https://brtc-sdk.cdn.bcebos.com/ai/agent/ios/BaiduChatAgent.IOS_V1.2.2.zip

文档详见SDK压缩包中。

</> iOS SDK 函数简介

Object-C

```
1 1. 初始化BaiduChatAgent
2 self.baiduChatAgent = [[BaiduChatAgent alloc]
  initWithParams:self.baiduChatAgentParams delegate:self];
3 2. 开启对话
4 - (void)call:(NSString *)token instanceId:(NSInteger)instanceId;
5 3. 结束对话
6 - (void)hangup;
7 4. 设置回调
8 - (id)initWithParams:(AgentEngineParams*) params delegate:
  (id<BaiduChatAgentDelegate>)delegate;
9 5. 发送文本
10 - (void)sendTextToAIEngine:(NSString *)text;
```

3.4. H5网页接入指引

H5 SDK: <https://brtc-sdk.cdn.bcebos.com/ai/agent/h5/BRTC.Agent.H5.SDK.V1.0.4.zip>

H5在线demo: https://brtc-sdk.cdn.bcebos.com/ai/agent/h5/brtc_agent.html

文档详见SDK压缩包中。

</> H5 SDK 函数简介

JavaScript

```
1 1. 初始化
2 var Agent = new BaiduRtcAgentClient();
```

```
3
4 2. 开始通话
5 var cfg = {
6     llm: 'LLMxxx', // 私有LLM 可以填入'LLMOpenAPI', llm_url传入私有化地址
7     llm_url: ''
8 };
9 Agent.Start({
10     appid: 'AppIdxxx',
11     cfg: cfg,
12     remotevideoviewid: 'therevideo',
13     localvideoviewid: 'herevideo',
14     success: function () {
15     },
16     error: function (error) {
17     },
18     onmessage: function (msg) {
19         console.log('onmessage id: ' + msg.id + ' data: ' + msg.data);
20     }
21
22 3. 发送文本消息
23 Agent.sendMessageToUser('[T]:你好');
24
25 4. 结束通话
26 Agent.Stop();
```

3.5. rtos sdk接入指引

1	芯片平台	SDK下载地址
2	乐鑫esp32-s3	https://brtc-sdk.cdn.bcebos.com/rtos-esp32%2Fchatagent%2FBRTC.RTOS.SDK.ESP32S3.V3.0.8B01.tmp20250305.zip

文档详见SDK压缩包中。

</> rtos sdk函数简介

Java

```
1 1. 初始化初始化AIEngine
2     BaiduChatAgentEngine* baidu_create_chat_agent_engine(const
   BaiduChatAgentEvent* events);
3     int baidu_chat_agent_engine_init(BaiduChatAgentEngine* engine, const
   AgentEngineParams* param);
4 2. 开启对话
5     void baidu_chat_agent_engine_call(BaiduChatAgentEngine* engine);
6 3. 发送文本
```

```

7 void baidu_chat_agent_engine_send_text(BaiduChatAgentEngine* engine, const
  char* text);
8 4. 结束对话
9 void baidu_chat_agent_engine_destroy(BaiduChatAgentEngine* engine);

```

</> 服务端创建智能体请求按照以下格式

Plain Text

```

1 #####注意事项1#####
2 #ifdef BRTC_ENABLE_G722
3 #define JSON_CONFIG_TEMPLATE "{\"app_id\": \"%s\", \"config\": \"{\\\"llm\\\" :
  \\\"%s\\\", \\\"llm_token\\\" : \\\"no\\\", \\\"rtc_ac\\\": \\\"g722\\\",
  \\\"lang\\\" : \\\"%s\\\"}\"\", \"quick_start\": true}"
4 #else
5 #define JSON_CONFIG_TEMPLATE "{\"app_id\": \"%s\", \"config\": \"{\\\"llm\\\" :
  \\\"%s\\\", \\\"llm_token\\\" : \\\"no\\\", \\\"rtc_ac\\\": \\\"pcmu\\\",
  \\\"lang\\\" : \\\"%s\\\"}\"\", \"quick_start\": true}"
6 #endif
7 //目前rtos 支持pcmu和g722格式, 需要按照上面格式将音频格式进行http请求发送.
8 #####
9
10
11 #####注意事项2#####
12 const char* json_str =
13 "{"
14   "\"ai_agent_instance_id\": 2230595646193664,\"
15   "\"context\": {"
16     "\"cid\": 1,\"
17     "\"token\":
18     \"00415f2bd1c3fe5dbc02cc49af726b18d6c6bfc56dc174238426178be806a1742470661\""
19   "}"
20 }; //这是服务端返回的参数, 需要将参数给客户端
21 void setUserParameters(AgentEngineParams *params) {
22   strncpy(params->agent_platform_url, SERVER_HOST_ONLINE, sizeof(params-
23   >agent_platform_url) - 1); //SERVER_HOST_ONLINE:https://rtc-aiagent.baidubce.com
24   strncpy(params->appid, BDCloudDefaultRTCApplID, sizeof(params->appid) - 1); //
  需要和服务端使用同一个appId
25   strncpy(params->cer, ".a.cer", sizeof(params->cer) - 1);
26   strncpy(params->workflow, "VoiceChat", sizeof(params->workflow) - 1);
27   strncpy(params->remote_params, json_str, sizeof(params->remote_params) - 1);
28   params->verbose = false;
29   params->enable_local_agent = false;
30   params->enable_voice_interrupt = true;

```

```

30     params->level_voice_interrupt = 80;
31     strncpy(params->llm, "LLMRacing", sizeof(params->llm) - 1);
32     strncpy(params->lang, "zh", sizeof(params->lang) - 1);
33     params->AudioInChannel = 1;
34     params->AudioInFrequency = 16000;
35 }

```

</> 客户端自己创建智能体

Plain Text

```

1 void setUserParameters(AgentEngineParams *params) {
2     strncpy(params->agent_platform_url, SERVER_HOST_ONLINE, sizeof(params->
>agent_platform_url) - 1); //SERVER_HOST_ONLINE:http://rtc-aiagent.baidubce.com
3     strncpy(params->appid, BDCloudDefaultRTCAppID, sizeof(params->appid) - 1);//需
要和服务端使用同一个appId
4     strncpy(params->cer, "./a.cer", sizeof(params->cer) - 1);
5     strncpy(params->workflow, "VoiceChat", sizeof(params->workflow) - 1);
6     params->instance_id = 10373;
7     params->verbose = true;
8     params->enable_local_agent = true;
9     params->enable_voice_interrupt = true;
10    params->level_voice_interrupt = 80;
11    strncpy(params->llm, "LLMRacing", sizeof(params->llm) - 1);
12    strncpy(params->lang, "zh", sizeof(params->lang) - 1);
13    params->AudioInChannel = 1;
14    params->AudioInFrequency = 16000;
15 }

```

4.接入曦灵数字人的说明

在曦灵管理平台创建数字人应用， 购买直播组件。

<https://xiling.cloud.baidu.com/open/widgetStore/list>

获得数字人的AppID 和AppKey后使用Token 生成工具生成Token（根据业务需要选择token有效期，建议为48小时）：

<https://open.xiling.baidu.com/token-gen-tool.html>

把Token填入xiling_auth字段。

</>

Bash

```
1 https://brtc-sdk.bj.bcebos.com/ai/agent/h5/brtc_agent.html?
  a=XX&it=DigitalHumanH&xiling_auth=YY
```

</> 配置曦灵数字人Token

JavaScript

```
1 var cfg = {
2   llm: 'LLMxxx', // 私有LLM 可以填入'LLMOpenAPI', llm_url传入私有化地址
3   llm_url: '',
4   xiling_auth: 'xiling-token-XXX',
5   xiling_bgimg: '',
6   fid: 'A2A_V2-muqing', // 数字人形象, 可选形象
  https://cloud.baidu.com/doc/AI_DH/s/2lyzilgsg
7   tts_url: 's?per=5132', // 数字人音色, 可选音色
  https://cloud.baidu.com/doc/AI_DH/s/Slywt3fxy
8 };
9 Agent.Start({
10   instance_type: 'DigitalHuman', // 指明是数字人智能体
11   appid: 'AppIdxxx',
12   cfg: cfg,
13   remotevideoviewid: 'therevideo',
14   localvideoviewid: 'herevideo',
15   success: function () {
16   },
17   error: function (error) {
18   },
19   onmessage: function (msg) {
20     console.log('onmessage id: ' + msg.id + ' data: ' + msg.data);
21   }
22 }
```