

【IOS-SDK】 AI智能体接口(对外版本)

目录

- (一) 背景
 - 1.1 账号信息
 - 1.2 测试环境地址
 - 1.3 SDK下载地址及Demo地址下载
- (二) SDK接口封装
 - 2.1 初始化
 - 2.2 开启对话
 - 2.3 结束对话
 - 2.4 设置回调
 - 2.5 发送文本
 - 2.6 声音配置
 - 2.7 主动打断
 - 2.8 销毁
 - 2.9 function结果回调
 - 2.9 设置数字人 view的参数 音频智能体不用调用该接口
 - 2.10 上传文件
 - 2.11 相机相关
- (三) 音频相关
 - 3.1 外部音频
- (四) 事件回调
 - 4.1 AI Agent回调

(一) 背景

本文档描述AI智能体接口文档，本SDK是用于正式对接版本；

1.1 账号信息

app_id (仅仅用于本次测试，不能线上使用)： apppibr915pu75m

token: 可以默认输入 "no_token"

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 (仅仅用于本次测试, 不能线上使用): <http://182.61.62.159:8988>

正式地址: <https://https://ai.agent.kaywang.cn/>

1.3 SDK下载地址及Demo地址下载

手动下载SDK (替换成所需版本号及对应地址)

</>

Java

```
1 https://brtc-sdk.bj.bcebos.com/ios/BaiduChatAgent_V1.6.6.zip
2
```

(二) SDK接口封装

2.1 初始化

</> 初始化参数

Object-C

```
1 @interface AgentEngineParams : NSObject
2
3 /** 服务器端接口下发 cid, token, 当前调试传入空即可 { "cid": 1, "token": "xxx-aa-bb" } */
4 @property (nonatomic, copy) NSString *config;
5
6 /** appid */
```

```
7 @property (nonatomic, copy) NSString *appId;
8
9 /**  workflow类型, 默认用VoiceChat */
10 @property (nonatomic, assign) NSString * workflow;
11
12 /** 实例id  服务器端下发, 当前调试默认即可 */
13 @property (nonatomic, assign) NSInteger instanceId;
14
15 /**  调试开关 */
16 @property (nonatomic, assign) BOOL verbose;
17
18 /**
19  *  海外服务器配置, 参考 BaiduChatAgentRegionType
20  *  @discussion  中国大陆使用不用填写。
21  */
22 @property (nonatomic, copy) NSString *regionServer;
23
24 /**  智能互动服务地址 */
25 @property (nonatomic, copy) NSString *rtcServerUrl;
26
27 /**  授权 key */
28 @property (nonatomic, copy) NSString *licenseKey;
29
30 /**  用户 id */
31 @property (nonatomic, copy) NSString *userId;
32
33 /**  语音自动打断 */
34 @property (nonatomic, assign) BOOL enableVoiceInterrupt;
35
36 /**  本地音频打断能量值 默认80 */
37 @property (nonatomic, assign) NSInteger userAudioLevelInterrupt;
38
39 /**  是否使用外部音频进行采集和播放 */
40 @property (nonatomic, assign) BOOL enableExternalAudioDevice;
41
42 /**
43  音频采样率
44  @discussion  目前仅外部音频生效
45  */
46 @property (nonatomic, assign) int audioSampleRate;
47
48 /**
49  是否 dump 录音数据
```

```
50 @discussion 录制格式 pcm, 每次 call hangup 形成一个录音文件, 录制目录:
    Documents/audio_dump
51 */
52 @property (nonatomic, assign) BOOL enableDumpAudioInput;
53
54 @end
```

</> 初始化BaiduChatAgent

Object-C

```
1 self.baiduChatAgentParams = [[AgentEngineParams alloc] init];
2
3 self.baiduChatAgentParams.appId = req.appId;
4 self.baiduChatAgentParams.config = res.context;
5 self.baiduChatAgentParams.instanceId = res.instanceId;
6 self.baiduChatAgentParams.verbose = YES;
7 self.baiduChatAgentParams.enableVoiceInterrupt = YES;
8
9 // 鉴权信息
10 self.baiduChatAgentParams.licenseKey = @"xxxxx";
11 self.baiduChatAgentParams.userId = @"1234567890";
12
13 self.baiduChatAgent = [[BaiduChatAgent alloc]
    initWithParams:self.baiduChatAgentParams delegate:self];
```

2.2 开启对话

</>

Object-C

```
1 /**
2  * 开启对话
3  * @param token 鉴权token
4  * @param instanceId 实例id
5  */
6 - (void)call:(NSString *)token instanceId:(NSInteger)instanceId;
```

2.3 结束对话

</>

Object-C

```
1 /**
2  * 挂断
3  */
4 - (void)hangup;
```

2.4 设置回调

</> 初始化时候同步设置回调函数

Object-C

```
1 /**
2  * 初始化
3  * @param params 参数
4  * @param delegate 回调
5  */
6 - (id)initWithParams:(AgentEngineParams*) params delegate:
    (id<BaiduChatAgentDelegate>)delegate;
```

2.5 发送文本

</>

Object-C

```
1 /**
2  * 发送文本给智能体
3  * @param text 文本
4  */
5 - (void)sendTextToAIAgent:(NSString *)text;
```

</>

Object-C

```
1 /**
2  * 发送文本给TTS，直接进行播报
3  * @param text 文本
4  */
5 - (void)sendTextToTTS:(NSString *)text;
```

</>

Java

```
1 /**
2  * 更新query 增强，通过向用户query 插入预置文本
3  *
4  * @param enhanceType 插入类型包括 0:不插入、1:前插、2:后插、3:前后插)
5  * @param preQuery 前插文本
6  * @param postQuery 后插文本
7  */
8 - (void)updateEnhanceQuery:(NSString *)enhanceType preQuery:(NSString *)preQuery
   postQuery:(NSString *)postQuery;
```

2.6 声音配置

</>

Object-C

```
1 /**
2  * 静音播放
3  * @param isMute 是否静音
4  * */
5 - (void)mutePlayback:(BOOL)isMute;
6
7 /**
8  * 静音mic
9  * @param isMute 是否静音
10 * */
11 - (void)muteMic:(BOOL)isMute;
12
13 /**
14  * 切换扬声器
15  * @param speaker 是否扬声器
16  * */
17 - (void)switchToSpeaker:(BOOL)speaker;
18
19 /**
20  * 预置听筒/扬声器
21  * @param isSpeaker true:扬声器 false:听筒 默认听筒
22  * @discussion 在 initWithParams 之后, call 之前调用, 预置听筒/扬声器播放语音
23  */
24 - (void)presetLoudSpeaker:(BOOL)isSpeaker;
```

2.7 主动打断

```
</> Object-C

1 /**
2  * 打断智能体播报
3  * */
4 - (void)interrupt;
5
6 /**
7  * 打断智能体播报
8  * @param isBegin 是否是开始打断
9  * @param delay 持续时长（毫秒）
10 * */
11 - (void)interrupt2:(BOOL)isBegin delay:(int)delay;
```

2.8 销毁

```
</> Object-C

1 /**
2  * 资源销毁
3  * */
4 - (void)destroy;
```

2.9 function结果回调

```
</> Object-C

1 /**
2  * function回调结果处理完通知大模型
3  * */
4 - (void)sendFunctionCallResult:(NSString*) functionID result:(NSString *)msg {
5
```

2.9 设置数字人 view的参数 音频智能体不用调用该接口

说明：RTCRemoteVideoView 来自BaiduRtcRoom.framework的头文件

</>

Object-C

```
1 /**
2  * 设置数字人 view的参数 音频智能体不用调用该接口
3  * @param remoteVideoView
4  * */
5 - (void)setDigitalDisplay:(RTCRemoteVideoView *) remoteVideoView;
6
```

2.10 上传文件

</>

Object-C

```
1 /**
2  * 上传文件
3  * @param path 文件路径
4  * @param expire 过期时间
5  * */
6 - (BOOL)uploadFile:(NSString *)path expire:(NSInteger)expire;
```

2.11 相机相关

</>

Object-C

```
1 /**
2  * 设置本地相机预览视图
3  * @param localVideoView 本地预览视图
4  * */
5 - (void)setLocalDisplay:(RTCLocalVideoView *)localVideoView;
6
```



```

7 /**
8  * 打开相机预览
9  */
10 - (void)startCameraPreview;
11
12 /**
13  * 关闭相机预览
14  */
15 - (void)stopCameraPreview;
16
17 /**
18  * 翻转相机
19  */
20 - (void)switchCamera;

```

（三）音频相关

3.1 外部音频

</>

Object-C

```

1 @interface AgentEngineParams : NSObject
2
3 /** 是否使用外部音频进行采集和播放 */
4 @property (nonatomic, assign) BOOL enableExternalAudioDevice;
5
6 @end

```

</>

Object-C

```

1 @interface BaiduChatAgent : NSObject
2
3 /**
4  * 设置外部音频设备进行采集和播放
5  *
6  * @param audioExternalDelegate 外部音频设备代理
7  * @discussion 需要在 call 之前设置 AgentEngineParams.enableExternalAudioDevice 为 YES

```

启用该功能

```
8  */
9  - (void)setAudioExternalDeviceDelegate:
    (id<BaiduRtcRoomApiAudioExternalDeviceDelegate>)audioExternalDelegate;
10
11 @end
```

(四) 事件回调

4.1 AI Agent回调

</>

Java

```
1
2  /*****1*****/
3  * AI智能体音频状态变化
4  *  STOPPED = 1 大模型停止说话;  SPEAKING = 2 大模型说话;
5  */
6  - (void)onAIAgentAudioStateChange:(AGentAudioState)newState;
7
8  /*****2*****/
9  * 用户声音状态变化
10 *  STOPPED = 1 用户停止说话; SPEAKING = 2 用户说话
11 */
12 - (void)onUserAudioStateChange:(AGentAudioState)newState;
13
14 /*****3*****/
15 * 智能体音频能量
16 */
17 - (void)onAIAgentAudioLevel:(NSInteger)level;
18
19 /*****4*****/
20 * 用户音频能量
21 */
22 - (void)onUserAudioLevel:(NSInteger)level;
23
24 /*****5*****/
```

```

25  * AI智能体结果
26  * isFial False 一句话一部分, YES 一句话结束
27  */
28 - (void)onAIAgentSubtitle:(NSString *)text isFinal:(BOOL)isFial;
29
30 /*****6*****/
31  * 用户端音频ASR结果
32  * isFial False 一句话一部分, YES 一句话结束
33  */
34 - (void)onUserAsrSubtitle:(NSString *)text isFinal:(BOOL)isFinal;
35
36 /*****7*****/
37  * 链接状态变化 详细参考结构体AGentConnectState
38  */
39 - (void)onConnectionStateChange:(AGentConnectState)state;
40
41 /*****8*****/
42  * 状态变化
43  */
44 - (void)onCallStateChange:(AGentCallState)state;
45
46 /*****9*****/
47  * 错误信息回调
48  * @param errCode 错误号码
49  * @param errMsg 错误消息提示
50  * @param extInfo 附加信息
51  */
52 - (void)onError:(NSInteger)errCode errMsg:(NSString *)errMsg extInfo:(nullable
    NSDictionary*)extInfo;
53
54 /*****10*****/
55  * functionCall
56  * @param id functionCall 单次唯一标识
57  * @param params json参数
58  * */
59 - (void) onFunctionCall:(NSString*) functionID param:(NSString*) params;
60
61 /*****11*****/
62 /**
63  文件上传状态回调
64  * @param code 状态码
65  */
66 - (void)onUploadFileStatus:(AgentErrorCode)code fileName:(NSString *)fileName;
67

```

```
68 /*****12*****/
69 /**
70  * Agent 意图
71  * @param type 意图类型, 暂时支持 [image, video]
72  * @param extInfo 附加信息
73  */
74 - (void)onAgentIntent:(NSString *)type extInfo:(NSDictionary * __nullable)extInfo;
75
76 /**
77  * 收到标签消息
78  * @param message 自定义消息内容 (包含标签格式)
79  */
80 - (void)onRecvTagMessage:(NSString *)message;
81
82 /**
83  * license 授权状态回调
84  * @param status 授权状态: 0 成功; -1 失败
85  */
86 - (void)onAgentLicenseStatus:(int)status;
87
88 /**
89 内部日志回调, 通过 AgentEngineParams.enableLogCallback 开启
90 @param log 日志内容
91 */
92 - (void)onAgentLogPrint:(NSString *)log;
```