

使用方式

AI 助手模块提供智能对话能力，支持实时问答、聊天历史和音频处理功能。

主要方法

方法	功能
<code>setupAIAssistant</code>	初始化 AI 助手
<code>sendAIQuestionMsg</code>	发送问题给 AI 助手
<code>getAIAssistantChatHistory</code>	获取聊天历史
<code>createAiAssistantAudioTask</code>	创建音频合成任务
<code>getAiAssistantAudioTaskResult</code>	获取音频任务结果
<code>recognizeAudioBlobToText</code>	音频转文字

使用示例

```
// 初始化 AI 助手（独立模式）
await aiCore.aiAssistant.setupAIAssistant(
  {
    aiAssistantCode: 'your-code',
    aiAssistantToken: 'your-token',
  },
  { independent: true },
);

// 发送问题
await aiCore.aiAssistant.sendAIQuestionMsg({
  content: '请介绍一下视频内容',
});

// 获取聊天历史
const history = await aiCore.aiAssistant.getAIAssistantChatHistory();

// 音频合成（TTS）
const { taskId } = await aiCore.aiAssistant.createAiAssistantAudioTask({
  text: '你好，我是 AI 助手',
  rate: 1.0,
});

// 音频识别
const text = await aiCore.aiAssistant.recognizeAudioBlobToText({
  audioBlob: audioFile,
  audioType: 'wav',
});
```

是否支持 AI 助手聊天功能

Api 方法： `supportAIAssistantChat(): SupportResult`

返回值说明： `SupportResult` 类型

获取 AI 助手相关配置

Api 方法： `getAIAssistantConfig(): PolyvAIAssistantConfig`

返回值说明： AI 助手配置， `PolyvAIAssistantConfig` 类型，详细类型说明如下

属性名	说明	类型
<code>aiDigitalHumanEnabled</code>	AI 数字人开关	<code>boolean</code>

属性名	说明	类型
aiAssistantEnabled	AI 助手开关，默认关	boolean
aiAssistantExpired	AI 助手是否过期，默认为 false	boolean

获取 AI 助手信息

说明： 获取当前 AI 助手的配置信息，包括助手 ID、名称、数字人信息等

Api 方法： `getAIAssistantInfo()`: PolyvAIAssistantInfo

返回值说明： AI 助手信息对象， PolyvAIAssistantInfo 类型，详细类型说明如下

属性名	说明	类型
digitalHumanId	AI 数字人 Id,, 默认 -1	number
digitalHumanName	AI 数字人名称	string
videoSource	视频资源	Object
aiAssistantId	AI 助手Id, 默认 -1	number
aiAssistantName	AI 助手名称，默认为 'AI'	string
aiAssistantCode	AI 助手编码	string
aiAssistantToken	AI 助手 Token	string
welcomeSpeech	AI 助手欢迎语	string

示例：

```
const assistantModule = new AIAssistantModule();
await assistantModule.setupAIAssistant({ aiAssistantCode: 'xxx' });
const info = assistantModule.getAIAssistantInfo();
console.log('助手ID:', info.aiAssistantId);
console.log('助手名称:', info.aiAssistantName);
console.log('欢迎语:', info.welcomeSpeech);
console.log('数字人ID:', info.digitalHumanId);
console.log('数字人名称:', info.digitalHumanName);
console.log('说话视频地址:', info.videoSource?.speakUrl);
console.log('不说话视频地址:', info.videoSource?.noSpeakUrl);
```

初始化 AI 助手

Api 方法： `setupAIAssistant(aiAssistantInfo: Partial<PolyvAIAssistantInfo>, opts?: Object): Promise<void>`

参数说明：

- aiAssistantInfo: `Partial<PolyvAIAssistantInfo>` 类型，必传
- opts: `Object` 类型，选传，详细类型说明如下

参数名	说明	类型	必须	默认值
<code>independent</code>	-	<code>boolean</code>	否	-
<code>previewMode</code>	-	<code>boolean</code>	否	-

发送 AI 提问消息

用于观众发送提问消息给到 AI 助手，调用过程中会触发 `AIAssistantEvents.AIAssistantChatStatusChange`、`AIAssistantEvents.AIAssistantChatMessage`、`AIAssistantEvents.ReplaceAIAssistantChatMessage` 等事件。

Api 方法： `sendAIQuestionMsg(__namedParameters: SendAIQuestionMsgOptions): Promise<void>`

参数说明：

- __namedParameters: `SendAIQuestionMsgOptions` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
<code>content</code>	提问内容	<code>string</code>	是	-

获取 AI 助手聊天历史

说明： 获取当前用户与 AI 助手的聊天历史记录，包括问题和答案

Api 方法： `getAIAssistantChatHistory(): Promise<PolyvAIAssistantChatMsgType[]>`

返回值说明： 聊天消息数组，按时间顺序排列， `Promise<PolyvAIAssistantChatMsgType[]>` 类型

示例：

```
const assistantModule = new AIAssistantModule();
try {
  const history = await assistantModule.getAIAssistantChatHistory();
  console.log('历史消息数量:', history.length);

  history.forEach(msg => {
    if (msg.msgSource === PolyvAIAssistantChatMsgSource.Question) {
      console.log('【用户提问】:', msg.content);
    } else {
      console.log('【AI 回答】:', msg.content);
    }
  });
} catch (error) {
  console.error('获取聊天历史失败:', error);
}
```

创建 AI 助手音频合成任务

说明： 将文本转换为语音并生成音频文件，支持自定义语速和语音类型

Api 方法： `createAiAssistantAudioTask(opts: Object): Promise<Object>`

参数说明：

- opts： 音频合成参数， `Object` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
rate	语速倍率，默认 1（正常速度），范围 0.5-2	number	否	-
text	要转换的文本内容，必填	string	是	-
ttsVoiceId	语音 ID，可选，不同的 ID 代表不同的语音风格	string	否	-

返回值说明： 包含任务 ID 的对象，用于后续获取音频结果， `Promise<Object>` 类型，详细类型说明如下

属性名	说明	类型
taskId	-	string

示例：

```
const assistantModule = new AIAssistantModule();

// 创建音频合成任务
const { taskId } = await assistantModule.createAiAssistantAudioTask({
  text: '欢迎观看直播，这是 AI 助手为您生成的语音内容',
  rate: 1,
  ttsVoiceId: 'female_zh_1'
});
console.log('任务ID:', taskId);

// 等待一段时间后获取音频结果
const audioResult = await assistantModule.getAiAssistantAudioTaskResult(taskId);
console.log('音频URL:', audioResult.audioUrl);
```

获取 AI 助手音频合成任务结果

说明： 根据任务 ID 获取之前创建的音频合成任务的结果，包括音频 URL 和状态

Api 方法： `getAiAssistantAudioTaskResult(taskId: string):`

`Promise<PostAiAssistantGetAudioTaskResultResponseData>`

参数说明：

- taskId：音频合成任务 ID（通过 createAiAssistantAudioTask 获取），string 类型，必传

返回值说明： 音频任务结果对象，包含音频地址和状态信息，

`Promise<PostAiAssistantGetAudioTaskResultResponseData>` 类型

识别音频转文字

说明： 将用户上传的音频文件识别并转换为文字内容

Api 方法： `recognizeAudioBlobToText(params: Object): Promise<string>`

参数说明：

- params：音频识别参数，Object 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
audioBlob	音频数据 Blob 对象，必填	Blob	是	-

参数名	说明	类型	必须	默认值
audioType	音频格式类型，默认 'wav'，可选 'mp3'、'wav'、'm4a' 等	string	否	-

返回值说明：识别出的文字内容，Promise<string> 类型

示例：

```
const assistantModule = new AIAssistantModule();

// 假设已有音频 blob
const audioBlob = new Blob([audioData], { type: 'audio/wav' });

try {
  const text = await assistantModule.recognizeAudioBlobToText({
    audioBlob,
    audioType: 'wav'
  });
  console.log('识别结果:', text);

  // 可以将识别结果发送给 AI 助手
  await assistantModule.sendAIQuestionMsg({ content: text });
} catch (error) {
  console.error('音频识别失败:', error);
}
```