

使用方式

AI 总结模块提供获取回放视频智能分析内容的能力，包括回放大纲、字幕解析和互动答题。

主要方法

方法	功能
<code>getPlaybackOutline</code>	获取回放大纲数据
<code>getOutlineContent</code>	获取大纲详细内容
<code>getAISubtitleContent</code>	获取 AI 字幕内容
<code>parseAISubtitleContent</code>	解析 SRT 格式字幕
<code>getAIPlaybackQuestionData</code>	获取互动答题数据

使用示例

```
// 获取回放大纲
const outline = await aiCore.aiSummary.getPlaybackOutline({
  id: 'video-123',
  type: 'playback', // 'record' 或 'playback'
});

if (outline) {
  console.log('视频摘要:', outline.introduction);
  console.log('分段数量:', outline.outlineContent.length);
}

// 获取并解析字幕
const subtitleContent = await aiCore.aiSummary.getAISubtitleContent(
  'https://example.com/subtitle.srt',
);
const parsedSubtitles = aiCore.aiSummary.parseAISubtitleContent(subtitleContent);

// 获取互动答题
const questionData = await aiCore.aiSummary.getAIPlaybackQuestionData(
  'https://example.com/questions.json',
);
```

获取回放大纲内容

说明： 通过大纲 URL 获取回放大纲的详细内容，包括摘要和分段总结

Api 方法： `getOutlineContent(outlineUrl: string): Promise<PlaybackOutlineData>`

参数说明：

- outlineUrl： 大纲数据的 API URL， `string` 类型，必传

返回值说明： 回放大纲数据，包含 introduction（摘要）和 outlineContent（分段总结），`Promise<PlaybackOutlineData>` 类型，详细类型说明如下

属性名	说明	类型
<code>introduction</code>	摘要内容	<code>string</code>
<code>outlineContent</code>	分段总结	<code>PlaybackOutlineContentItem[]</code>

示例：

```
const summaryModule = new AISummaryModule();
const outlineUrl = 'https://api.example.com/outline/123';
try {
  const outlineData = await summaryModule.getOutlineContent(outlineUrl);
  console.log('大纲摘要:', outlineData.introduction);
  console.log('分段数量:', outlineData.outlineContent.length);
} catch (error) {
  console.error('获取大纲内容失败:', error);
}
```

获取回放大纲数据

说明： 根据回放 ID 和类型获取对应的大纲数据。首先通过 API 获取大纲 URL，然后使用 `getOutlineContent` 获取详细内容。

Api 方法： `getPlaybackOutline(options: PlaybackOutlineOptions): Promise<undefined | PlaybackOutlineData>`

参数说明：

- options： 获取参数，包含回放 ID 和类型， `PlaybackOutlineOptions` 类型，必传，详细类型说明如下

参数名	说明	类型	必须	默认值
id	回放 id	string	是	-
type	回放类型	string	是	-

返回值说明： 回放大纲数据，如果未找到大纲则返回 undefined，Promise<undefined | PlaybackOutlineData> 类型

示例：

```
const summaryModule = new AISummaryModule();

// 获取录制文件的大纲
const recordOptions = {
  id: 'file_123456',
  type: 'record' as const,
};
const recordOutline = await summaryModule.getPlaybackOutline(recordOptions);
if (recordOutline) {
  console.log('录制文件大纲:', recordOutline.introduction);
}

// 获取回放视频的大纲
const playbackOptions = {
  id: 'video_789012',
  type: 'playback' as const,
};
const playbackOutline = await summaryModule.getPlaybackOutline(playbackOptions);
if (playbackOutline) {
  console.log('回放视频大纲分段:', playbackOutline.outlineContent.length);
}
```

获取字幕内容

说明： 通过字幕 URL 获取 AI 生成的字幕文本内容（通常为 SRT 格式）

Api 方法： getAISubtitleContent(url: string): Promise<string>

参数说明：

- url: 字幕数据的 API URL，string 类型，必传

返回值说明： 字幕文本内容（字符串格式），Promise<string> 类型

示例：

```
const summaryModule = new AISummaryModule();
const subtitleUrl = 'https://api.example.com/subtitle/123.srt';
try {
  const subtitleContent = await summaryModule.getAISubtitleContent(subtitleUrl);
  console.log('字幕内容:', subtitleContent);
  // 通常返回 SRT 格式文本，可以进一步使用 parseAISubtitleContent 解析
  const parsedItems = summaryModule.parseAISubtitleContent(subtitleContent);
  console.log('解析后的字幕条目数:', parsedItems.length);
} catch (error) {
  console.error('获取字幕内容失败:', error);
}
```

解析字幕

说明： 将 SRT 格式的字幕文本解析为结构化的字幕条目数组

Api 方法： `parseAISubtitleContent(content: string): AISubtitleParsedItem[]`

参数说明：

- content: SRT 格式的字幕文本, `string` 类型, 必传

返回值说明： 解析后的字幕条目数组, 每个条目包含开始时间、结束时间、文本内容和索引, `AISubtitleParsedItem[]` 类型

示例：

```
const summaryModule = new AISummaryModule();
const srtContent = `
1
00:00:01,000 --> 00:00:04,000
欢迎观看本视频

2
00:00:05,000 --> 00:00:08,000
今天我们将学习 AI 技术
`;

const parsedItems = summaryModule.parseAISubtitleContent(srtContent);
console.log('解析结果:', parsedItems);
// 输出示例:
// [
//   {
//     start: 1000,
//     end: 4000,
//     text: '欢迎观看本视频',
//     index: '1'
//   },
//   {
//     start: 5000,
//     end: 8000,
//     text: '今天我们将学习 AI 技术',
//     index: '2'
//   }
// ]

// 使用解析后的数据
parsedItems.forEach(item => {
  console.log(`时间: ${item.start}-${item.end}ms, 内容: ${item.text}`);
});
```

获取互动答题列表

说明： 通过答题数据 URL 获取回放视频中的互动答题列表

Api 方法： `getAIPlaybackQuestionData(url: string): Promise<PlaybackQuestionData>`

参数说明：

- url: 互动答题数据的 API URL（JSON 格式）， `string` 类型，必传

返回值说明： 互动答题数据，包含问题列表， `Promise<PlaybackQuestionData>` 类型，详细类型说明如下

属性名	说明	类型
<code>questions</code>	问题列表	<code>PlaybackQuestionItem[]</code>

示例：

```
const summaryModule = new AISummaryModule();
const questionUrl = 'https://api.example.com/questions/123.json';
try {
  const questionData = await summaryModule.getAIPlaybackQuestionData(questionUrl);
  console.log('问题数量:', questionData.questions.length);

  questionData.questions.forEach((question, index) => {
    console.log(`问题 ${index + 1}: ${question.question}`);
    console.log(`类型: ${question.type}, 答案: ${question.answer}`);
    console.log(`弹题时间: ${question.questionTriggerTime}秒`);

    question.selectOptions.forEach(option => {
      console.log(` 选项 ${option.option}: ${option.content}`);
    });
  });
} catch (error) {
  console.error('获取答题数据失败:', error);
}
```