

AI 助手快速开始

本文档介绍如何在项目中快速集成 AI 助手组件。本篇文档以 AI 助手独立模式为示例

前置条件

- UI 层仅支持 Vue 2.7 版本，不兼容 Vue 3.x
- 从 Polyv 平台获取有效的 AI 助手凭证（`aiAssistantCode` 和 `aiAssistantToken`）

引入资源

在 HTML 文件中引入以下三个核心脚本：

```
<!-- 核心 SDK 链接 -->
<script src="https://polyv-frontend-sdk-test-01.oss-cn-shenzhen.aliyuncs.com/live-watch-ai-sdk/v1-latest/live-
watch-ai-sdk.umd.js"></script>

<!-- 仅支持 Vue2.7 版本 -->
<script src="https://s4.videooc.net/library/vue/2.x/vue-2.7.16.min.js"></script>

<!-- UI SDK 链接 -->
<script src="https://polyv-frontend-sdk-test-01.oss-cn-shenzhen.aliyuncs.com/live-watch-ai-ui/v1-latest/live-
watch-ai-ui.umd.js"></script>
```

核心 SDK 初始化

使用 `PolyvWatchAICore` 初始化核心 SDK：

```
const { PolyvWatchAICore, AIAssistantEvents } = window.PolyvLiveWatchAISdk;

let watchAICore = null;

function initWatchAICore() {
  watchAICore = new PolyvWatchAICore({
    domainInfo: {
      polyvApiDomain: `https://develop-3-api.polyv.net`,
    },
  });
}

function getWatchAICore() {
  if (!watchAICore) throw new Error('not init watchAICore');
  return watchAICore;
}
```

UI SDK 渲染

使用 Vue 2.7 组件 `AiAssistantChat` 渲染 AI 助手界面：

```

const { AiAssistantChat, useAIGlobalProvider } = window.PolyvLiveWatchAIUI;

function renderApp() {
  const appInstance = new window.Vue({
    components: {
      AiAssistantChat,
    },
    data: {
      renderEnabled: false,
    },
    setup() {
      useAIGlobalProvider({
        getWatchAICore,
        theme: 'light',
      });
    },
    created() {
      const watchAICore = getWatchAICore();

      watchAICore.aiAssistant.setupAIAssistant(
        {
          aiAssistantCode: 'your-ai-assistant-code',
          aiAssistantToken: 'your-ai-assistant-token',
        },
        { independent: true },
      );

      watchAICore.aiAssistant.eventEmitter.on(
        AIAssistantEvents.AIAssistantChatSetupedIndependent,
        () => {
          this.renderEnabled = true;
        },
      );
    },
    template: `
      <div v-if="renderEnabled" class="c-ai-assistant-container">
        <div class="c-ai-assistant-chat-wrapper">
          <AiAssistantChat direction="row" />
        </div>
      </div>
    `,
  });
  appInstance.$mount('#app');
}

```

完整示例

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1.0"
    />
    <title>AI 助手</title>
    <style>
      .c-ai-assistant-container {
        width: 100vw;
        height: 100vh;
        display: flex;
        align-items: center;
        justify-content: center;
        box-sizing: border-box;
        overflow: hidden;
        background-color: #f2f2f2;
      }

      .c-ai-assistant-chat-wrapper {
        width: 80%;
        height: 80%;
      }
    </style>
  </head>
  <body>
    <div id="app"></div>
  </body>
  <!-- 核心 SDK 链接 -->
  <script src="https://polyv-frontend-sdk-test-01.oss-cn-shenzhen.aliyuncs.com/live-watch-ai-sdk/v1-latest/live-watch-ai-sdk.umd.js"></script>
  <!-- 仅支持 Vue2.7 版本 -->
  <script src="https://s4.videooc.net/library/vue/2.x/vue-2.7.16.min.js"></script>
  <!-- UI SDK 链接 -->
  <script src="https://polyv-frontend-sdk-test-01.oss-cn-shenzhen.aliyuncs.com/live-watch-ai-ui/v1-latest/live-watch-ai-ui.umd.js"></script>
  <script>
    // ===== 核心 SDK 处理逻辑 =====
    const { PolyvWatchAICore, AIAssistantEvents } = window.PolyvLiveWatchAISdk;

    let watchAICore = null;
    function initWatchAICore() {
      watchAICore = new PolyvWatchAICore({
        domainInfo: {
          polyvApiDomain: `https://develop-3-api.polyv.net`,
        },
      });
    }
    function getWatchAICore() {
      if (!watchAICore) throw new Error('not init watchAICore');
    }
  </script>
</body>
</html>
```

```

    return watchAICore;
}

// ===== UI SDK 处理逻辑 =====
const { AiAssistantChat, useAIGlobalProvider } = window.PolyvLiveWatchAIUI;

function renderApp() {
  const appInstance = new window.Vue({
    components: {
      AiAssistantChat,
    },
    data: {
      renderEnabled: false,
    },
    setup() {
      useAIGlobalProvider({
        getWatchAICore,
        theme: 'light',
      });
    },
    created() {
      const watchAICore = getWatchAICore();

      watchAICore.aiAssistant.setupAIAssistant(
        {
          aiAssistantCode: 'your-ai-assistant-code',
          aiAssistantToken: 'your-ai-assistant-token',
        },
        { independent: true },
      );

      watchAICore.aiAssistant.eventEmitter.on(
        AIAssistantEvents.AIAssistantChatSetupIndependent,
        () => {
          this.renderEnabled = true;
        },
      );
    },
    template: `
      <div v-if="renderEnabled" class="c-ai-assistant-container">
        <div class="c-ai-assistant-chat-wrapper">
          <AiAssistantChat direction="row" />
        </div>
      </div>
    `,
  });
  appInstance.$mount('#app');
}

// ===== 主流程 =====
function main() {
  initWatchAICore();
  renderApp();
}

main();

```

```
</script>  
</html>
```