

Polyv 视频创作页面嵌入与通信协议

文档版本: v1.1 最后更新: 2026 年 6 月 16 日

1. 概述

本文档旨在帮助开发者将保利威 (Polyv) AI 视频创作功能以 `iframe` 的形式无缝集成到第三方业务平台中。通过遵循本文档中的指引，您可以实现：

- 界面定制：**通过 URL 参数控制嵌入页面中特定按钮的显示与隐藏。
- 功能调用：**使用 `Window.postMessage` API，在您的父页面与 Polyv 的 `iframe` 页面之间建立安全通信，以编程方式触发视频生成、草稿保存等操作。

本文档适用于需要在自有产品中提供 AI 视频创作能力，并希望对集成界面和流程进行自定义控制的客户。

2. 快速开始

要实现基础的嵌入和通信，您需要以下两个步骤：

- 嵌入 `iframe`：**将 Polyv AI 视频创作页面的 `iframe` 放置在您的页面中。
- 实现通信脚本：**编写 JavaScript 代码，用于监听 `iframe` 的就绪状态，并发送指令。

以下是一个基本的 HTML 结构示例：

```
<!DOCTYPE html>
<html>
<head>
  <title>集成 Polyv AI 视频创作</title>
</head>
<body>

  <!-- 1. 嵌入 Polyv AI 视频创作编辑页面 -->
  <iframe
    src="https://console.polyv.net/live/index.html#/ai-manager/video-production/create"
    frameborder="0"
    id="polyvAiFrame"
    style="width: 100%; height: 800px;">
  </iframe>

  <!-- 2. 父页面的控制按钮 -->
  <button id="createVideoButton" type="button">从外部触发生成</button>
  <button id="returnButton" type="button">返回</button>

  <!-- 3. 通信脚本 -->
  <script>
    // 详细逻辑见下文
  </script>

</body>
</html>
```

3. 功能详解

3.1. 界面定制 (URL 参数)

您可以通过在 `iframe` 的 `src` 属性链接后附加特定参数，来定制嵌入页面的外观。该方式灵活、易于实现。

参数列表

参数	类型	描述	示例
<code>video-production-submit</code>	String	N：隐藏页面右上角的【生成视频】按钮。	<code>...create?video-production-submit=N</code>
<code>video-production-return</code>	String	N：隐藏页面左上角的【返回】按钮。	<code>...create?video-production-return=N</code>

组合使用示例

若要同时隐藏“生成视频”和“返回”按钮，可以这样配置 `src`：

```
<iframe
  src="https://console.polyv.net/live/index.html#/ai-manager/video-production/create?video-production-submit=N&video-production-return=N"
  id="polyvAiFrame">
</iframe>
```

3.2. 交互通信 (PostMessage API)

为了实现父页面对 `iframe` 内操作的精准控制，我们定义了一套基于 `postMessage` 的双向通信协议。

核心流程：

1. `iframe` 页面加载完成后，会向父页面发送一个 `ready` 状态通知。
2. 父页面监听到 `ready` 通知后，即可确认 `iframe` 已准备就绪。
3. 此后，父页面可以向 `iframe` 发送 `create`、`save-draft` 等具体指令。
4. `iframe` 成功提交生成任务后，向父页面发送 `submit-success` 事件通知。
5. 当接入方隐藏 `iframe` 内置【返回】按钮，并使用父页面自己的【返回】按钮时，父页面可先向 `iframe` 发送 `save-draft`，等待 `iframe` 返回草稿保存结果后，再执行自身返回逻辑。

JavaScript 通信逻辑实现

```
const $iframe = document.getElementById('polyvAiFrame');
const $createVideoButton = document.getElementById('createVideoButton');
const $returnButton = document.getElementById('returnButton');
const targetOrigin = 'https://console.polyv.net';
let isFrameReady = false;

$createVideoButton.disabled = true;
$returnButton.disabled = true;

// 监听来自 iframe 的消息
window.addEventListener('message', (event) => {
  // 安全校验: 确保消息来自指定的源
  if (event.origin !== targetOrigin) {
    return;
  }

  try {
    const data = JSON.parse(event.data);

    // 只处理 Polyv AI 视频创作协议消息
    if (data.type !== 'ai-video-production') {
      return;
    }

    // ready: iframe 已完成初始化, 可以接收父页面指令
    if (data.type === 'ai-video-production' && data.event === 'ready') {
      isFrameReady = true;
      console.log('Polyv AI iframe 已准备就绪。');
      // 可在此处启用相关按钮
      $createVideoButton.disabled = false;
      $returnButton.disabled = false;
      return;
    }

    // save-draft: iframe 返回草稿保存结果
    if (data.type === 'ai-video-production' && data.event === 'save-draft') {
      if (data.success) {
        console.log('草稿保存成功, 可以执行父页面返回逻辑。');
        // TODO: 在这里执行接入方系统自己的返回逻辑
        // window.history.back();
      } else {
        console.warn('草稿保存失败: ', data.message || 'Fail reason');
        // TODO: 在这里提示用户, 父页面不应自动返回
      }
    }

    // submit-success: 视频生成任务已提交到后台处理队列
    if (data.type === 'ai-video-production' && data.event === 'submit-success') {
      console.log('视频生成任务已提交。');
    }
  } catch (error) {
    console.warn('无法解析来自 iframe 的消息: ', error);
  }
});
```

```
// 为父页面的按钮绑定点击事件
$createVideoButton.addEventListener('click', () => {
  if (!isFrameReady) {
    alert('视频创作页面尚未准备好，请稍候...');
    return;
  }

  // create: 请求 iframe 启动视频生成流程
  const message = {
    type: 'ai-video-production',
    event: 'create'
  };

  // 向 iframe 发送“生成视频”指令
  $iframe.contentWindow.postMessage(JSON.stringify(message), targetOrigin);
});

$returnButton.addEventListener('click', () => {
  if (!isFrameReady) {
    alert('视频创作页面尚未准备好，请稍候...');
    return;
  }

  // save-draft: 请求 iframe 保存当前编辑内容的草稿
  const message = {
    type: 'ai-video-production',
    event: 'save-draft'
  };

  // 向 iframe 发送“保存草稿”指令，等待 iframe 返回保存结果后再执行父页面返回逻辑
  $iframe.contentWindow.postMessage(JSON.stringify(message), targetOrigin);
});
```

4. 通信协议规范

所有通信消息均为 JSON 字符串格式，且 `type` 固定为 `ai-video-production`。草稿保存协议中，父页面发起请求和 `iframe` 返回结果均使用 `save-draft` 事件；父页面发起请求时不携带 `success`，`iframe` 返回结果时携带 `success` 和 `message`。

4.1. Iframe → 父页面

event 值	描述	消息体 (Payload) 示例
ready	通知父页面，iframe 已加载完毕，可以接收指令。这是一个关键的“握手”信号。	<pre>{ "type": "ai-video-production", "event": "ready" }</pre>

event 值	描述	消息体 (Payload) 示例
<code>submit-success</code>	通知父页面，当视频生成任务成功提交到后台处理队列后，通知父页面。表示用户的请求已被接受	<code>{"type": "ai-video-production", "event": "submit-success"}</code>
<code>save-draft</code>	通知父页面，本次草稿保存结果已返回，父页面可根据 <code>success</code> 判断是否继续执行自己的返回逻辑。	<code>{"type": "ai-video-production", "event": "save-draft", "success": true}</code>

4.2. 父页面 → Iframe

event 值	描述	消息体 (Payload) 示例
<code>create</code>	请求 iframe 启动视频生成流程。其效果等同于点击 iframe 内的【生成视频】按钮。	<code>{"type": "ai-video-production", "event": "create"}</code>
<code>save-draft</code>	请求 iframe 执行当前编辑内容的草稿保存。该动作通常由接入方自己的【返回】按钮触发。	<code>{"type": "ai-video-production", "event": "save-draft"}</code>

4.3. 草稿保存与外部返回流程

当接入方通过 `video-production-return=N` 隐藏 iframe 页面左上角【返回】按钮，并使用父页面自己的【返回】按钮时，应按以下流程处理：

1. 用户点击接入方系统自己的【返回】按钮。
2. 父页面确认已收到 iframe 的 `ready` 事件后，向 iframe 发送 `save-draft`。
3. Iframe 执行当前编辑内容的草稿保存。
4. Iframe 向父页面回调 `save-draft`，并通过 `success` 表达保存结果。
5. 当 `success` 为 `true` 时，父页面可以继续执行自己的返回逻辑。
6. 当 `success` 为 `false` 时，父页面不应自动返回；建议根据 `message` 提示用户或记录日志。

若当前页面没有新的未保存变更，iframe 仍应回调 `save-draft`，并返回 `success: true`。父页面不需要额外判断当前是否存在未保存变更。

5. 重要注意事项

1. **安全策略**：在 `addEventListener` 中，务必严格校验 `event.origin`，确保消息来源是 `https://console.polyv.net`。在 `postMessage` 时，也应明确指定 `targetOrigin`，避免使用 `*`，防止敏感数据泄露。
 2. **加载时序**：父页面必须等待并确认收到 `iframe` 的 `ready` 事件后，才能发送业务指令（如 `create`、`save-draft`）。建议在收到 `ready` 信号前，将父页面的控制按钮设置为禁用状态。
 3. **消息格式**：所有通信数据都必须通过 `JSON.stringify()` 序列化为字符串进行传输，并在接收端通过 `JSON.parse()` 解析。
 4. **外部返回**：在收到 `save-draft` 回调前，父页面不应直接关闭、跳转或销毁当前 `iframe` 页面。保存失败的典型场景包括但不限于网络或请求超时、登录态失效、服务端异常、当前页面数据不满足保存条件。
 5. **权限规则**：本期不新增额外权限判断，沿用视频创作页面当前已有的草稿保存权限与登录态规则。
-

如有任何疑问或需要进一步的技术支持，请联系我们的技术支持团队。