uni-app 微信小程序接入说明

本文档适用于已经使用 uni-app 开发微信小程序，并希望在小程序页面中接入保利威直播观看页能力的开发者。

通过本方案，uni-app 工程编译为微信小程序后，可在页面中以原生微信小程序自定义组件的方式挂载保利威观看组件，实现直播观看、回放观看、聊天室、观看条件、互动接收等观看页能力。

方案说明

uni-app 微信小程序观看组件属于微信小程序端接入方案，组件最终运行在 `mp-weixin` 产物中。

项目	说明
适用工程	uni-app 微信小程序工程，即编译目标为 `mp-weixin` 的工程。
接入方式	将保利威提供的 `polyv-live-watch` 原生小程序自定义组件包放入 uni-app 工程的 `wxcomponents` 目录。
观看流程	由 `polyv-watch-room` 组件内部承载，宿主页面只需要挂载组件并传入频道配置。
不适用范围	不适用于 App、H5、支付宝小程序、抖音小程序等端侧；也不是完整多端 uni-app SDK。

如果项目只需要低成本打开完整观看页，可优先评估微信小程序 WebView H5 观看页方案。如果项目只需要播放器能力，可评估保利威微信小程序直播播放插件、直播播放自定义组件或直播播放核心 SDK。

接入前准备

1. 保利威账号和频道

接入前需要准备：

- 可正常登录的保利威直播账号。
- 已创建的直播频道。

- 用于测试的频道 ID。

频道的观看条件、回放、聊天室、互动功能等能力，仍以保利威直播后台的频道配置为准。

2. 微信小程序主体和资质

直播播放、连麦、插件使用等能力与微信小程序主体、服务类目、接口权限和插件权限相关。相关审核和开通动作需要在客户自己的微信小程序 AppID 下完成。

建议接入前确认以下事项：

- 小程序主体已完成微信认证，且不是个人主体小程序。
- 小程序服务类目符合实际业务场景和微信审核要求。
- 已在微信公众平台后台添加保利威观看组件依赖的小程序插件。
- 如使用连麦能力，已确认 `live-pusher`、摄像头、麦克风等相关能力可用。
- 已在微信公众平台和保利威直播后台完成必要的业务域名配置。
- 如功能涉及用户信息、摄像头、麦克风等能力，已按微信要求配置用户隐私保护指引。

资质和类目要求会随微信平台规则变化而调整，最终以微信公众平台审核结果为准。

3. 获取组件包

接入方可以通过以下两种方式获取 `polyv-live-watch` 组件包。

获取方式	适用情况	操作说明
使用保利威交付包	保利威已提供可直接复制的 `polyv-live-watch` 目录。	无需自行构建，直接将 `polyv-live-watch` 复制到 uni-app 工程的 `src/wxcomponents` 目录。
从开源观看页项目构建	需要基于开源观看页项目自行生成 uni-app 微信小程序组件包。	克隆开源项目后安装依赖，并执行 `npm run build:uniapp-wxcomponents`。

从开源观看页项目构建时，可按以下步骤操作：

```
git clone https://gitee.com/polyv_ef/polyv-mp-live-watch-ui.git
cd polyv-mp-live-watch-ui
npm install
npm run build:uniapp-wxcomponents
```

构建完成后，组件包生成在：

```
dist-uniapp-wxcomponents/polyv-live-watch
```

将该目录完整复制到 uni-app 工程：

```
src/wxcomponents/polyv-live-watch
```

本文后续示例以复制后的以下目录结构为准：

```
src
  wxcomponents
    polyv-live-watch
      components
        watch-room
          watch-room.json
          watch-room.js
          watch-room.wxml
          watch-room.wxss
      polyv-live-watch.package.json
```

`polyv-live-watch` 是原生微信小程序自定义组件包，必须完整复制整个目录，不要只复制 `watch-room` 单个组件。

快速接入

1. 复制组件包

将 `polyv-live-watch` 目录复制到 uni-app 工程的 `src/wxcomponents` 下：

```
src/wxcomponents/polyv-live-watch
```

复制后，至少应存在以下入口文件：

```
src/wxcomponents/polyv-live-watch/components/watch-room/watch-room.json
src/wxcomponents/polyv-live-watch/components/watch-room/watch-room.js
src/wxcomponents/polyv-live-watch/components/watch-room/watch-room.wxml
src/wxcomponents/polyv-live-watch/components/watch-room/watch-room.wxss
```

2. 配置 `manifest.json`

在 `manifest.json` 的 `mp-weixin` 节点中开启自定义组件支持，并声明保利威观看组件依赖的小程序插件：

```
{
  "mp-weixin": {
    "usingComponents": true,
    "plugins": {
      "polyv-live-plugin": {
        "version": "1.3.1",
        "provider": "wxfb2e591959a8bacf"
      },
      "polyv-player": {
        "version": "1.17.0",
        "provider": "wx4a350a258a6f7876"
      }
    }
  }
}
```

如果工程中已经存在 `mp-weixin` 配置，只需要合并 `usingComponents` 和 `plugins`，不要覆盖工程原有的 `appid`、`setting`、`permission` 等配置。

3. 配置 `pages.json`

在需要展示观看页的 uni-app 页面中注册 `polyv-watch-room` 组件：

```
{
  "path": "pages/polyv-watch/index",
  "style": {
    "navigationBarTitleText": "直播观看",
    "usingComponents": {
      "polyv-watch-room": "/wxcomponents/polyv-live-watch/components/watch-room/watch-room"
    }
  }
}
```

路径以 uni-app 编译后的微信小程序根目录为基准，通常使用 `/wxcomponents/...` 形式。

4. 在页面中挂载组件

以下示例为 Vue 3 写法。Vue 2 工程可按相同的参数和事件名改写。

```
<script setup lang="ts">
import { onLoad, onUnload, onShareAppMessage } from '@dcloudio/uni-app'
import { reactive } from 'vue'

const configs = reactive({
  channelId: '',
  forceLayout: 'portrait',
})

onLoad((query) => {
  configs.channelId = typeof query?.channelId === 'string' ? query.channelId : ''
})

onUnload(() => {
  const pages = getCurrentPages()
  const currentPage = pages[pages.length - 1]
  const room = currentPage?.selectComponent?.('#polyvWatchRoom')
  room?.destroy?.()
})

onShareAppMessage(() => ({}))

function handleReady(event) {
  console.log('polyv watch ready', event.detail)
}

function handleViewChange(event) {
  console.log('polyv watch view change', event.detail)
}

function handleWebview(event) {
  const url = event.detail?.url

  if (!url) {
    return
  }

  uni.navigateTo({
    url: `/pages/webview/index?url=${encodeURIComponent(url)}`,
  })
}

function handleLoginRequired(event) {
  console.log('polyv watch login required', event.detail)
}
</script>

<template>
  <view class="polyv-watch-page">
    <polyv-watch-room
      id="polyvWatchRoom"
      class="polyv-watch-room-host"
      :configs="configs"
      @ready="handleReady"
      @view-change="handleViewChange"
```

```
      @webview="handleWebview"
      @login-required="handleLoginRequired"
    />
  </view>
</template>

<style>
page {
  height: 100%;
}

.polyv-watch-page {
  height: 100vh;
  min-height: 100vh;
  overflow: hidden;
}

.polyv-watch-room-host {
  display: block;
  width: 100%;
  height: 100%;
  min-height: 100vh;
}
</style>
```

页面只需要挂载一次 `polyv-watch-room`。进入引导页、观看页或错误页的过程由组件内部处理，不需要宿主页面根据事件重新挂载组件。

参数说明

`polyv-watch-room` 通过 `configs` 接收初始化配置。

字段	类型	必填	说明
`channelId`	`string`	是	直播频道 ID。
`forceLayout`	`'normal' | 'portrait'`	否	指定观看页布局。`portrait` 表示竖屏布局，`normal` 表示按频道或组件默认逻辑展示。

如后续交付包新增配置项，以对应版本的组件包说明为准。

事件说明

uni-app Vue 模板中建议使用 kebab-case 绑定事件。例如组件内部事件 `viewChange`，模板中写作 `@view-change`；`loginRequired` 写作 `@login-required`。

事件	触发时机	建议处理
`ready`	观看组件初始化完成。	可用于记录日志、埋点或更新宿主页面状态。
`viewChange`	观看组件内部视图发生变化，例如 `loading`、`splash`、`watch`、`error`。	可用于记录状态变化。不需要根据该事件重新挂载观看组件。
`webview`	观看组件需要打开外部网页。	宿主页面跳转到业务方自己的 WebView 页面，并传入 `event.detail.url`。
`loginRequired`	当前观看流程需要宿主侧处理登录。	宿主页面可跳转到业务方登录页，或打开自有登录组件。
`destroy`	观看组件销毁。	可清理宿主侧临时状态。

观看流程和宿主边界

本方案中，观看页主体由组件内部承载：

```
polyv-watch-room
  loading
  splash
  watch
  error
```

组件内部会根据频道配置和观看条件决定展示引导页、授权页、观看页或错误页。宿主 uni-app 页面只负责提供容器、传入配置和处理跨出组件边界的事件。

组件内部处理的行为包括：

- 进入引导或授权视图。
- 进入观看视图。
- 展示错误视图。
- 播放器、聊天室、互动组件等观看页内部状态流转。

宿主页面需要处理的行为包括：

- 打开业务方自己的 WebView 页面。
- 跳转到业务方自己的登录页。
- 按业务方规则配置小程序分享。
- 按业务方规则处理页面级路由和埋点。

不要在收到 `viewChange` 后再次创建或挂载新的观看组件，否则可能导致组件重复初始化、聊天室重复连接或播放状态异常。

样式和资源说明

`polyv-live-watch` 组件包已内置观看页所需的业务 WXSS 和静态资源。接入方通常不需要额外引入保利威原始工程的全局样式。

宿主页面建议只提供容器高度：

- `page` 设置 `height: 100%`。
- 页面根节点设置 `height: 100vh`。
- `polyv-watch-room` 宿主节点设置 `display: block`、`width: 100%`、`height: 100%`。

不建议在 uni-app 工程的全局样式中覆盖 `polyv-live-watch` 组件包内部类名，例如播放器、聊天室、互动卡片、图标、按钮等业务类名。全局覆盖可能造成图标尺寸异常、文字省略失效、弹层错位、按钮状态异常等问题。

如果接入后出现明显样式异常，建议按以下顺序排查：

1. 确认复制的是完整的 `polyv-live-watch` 目录。
2. 确认旧版本组件包已被完整覆盖，不存在新旧文件混用。
3. 确认 `src/wxcomponents/polyv-live-watch` 下的 `common`、`assets`、`package-watch`、`package-splash` 等目录完整存在。
4. 确认宿主页面没有覆盖组件包内部业务类名。
5. 使用微信开发者工具检查最终 `mp-weixin` 产物中组件、WXSS 和图片资源是否存在。

构建和预览

完成接入后，执行 uni-app 微信小程序构建命令。例如：

```
pnpm build:mp-weixin
```

实际命令以项目使用的包管理器和工程脚本为准，也可能是 `npm run build:mp-weixin` 或 HBuilderX 内置构建。

构建完成后，使用微信开发者工具打开 `dist/build/mp-weixin` 目录，并使用客户自己的 AppID 进行预览和真机调试。

建议至少完成以下验证：

- 能通过频道 ID 打开观看页。
- 无条件观看、验证码观看、登记观看、白名单观看等观看条件按频道配置正常展示。
- 直播或回放可正常播放。
- 聊天室可正常连接、接收和发送消息。
- 点赞、签到、卡片推送、条件抽奖、商品库等互动能力按频道配置正常展示。
- WebView 链接可通过 `webview` 事件交给宿主页面处理。
- 如使用连麦能力，真机上可正常申请连麦，并能获得摄像头、麦克风授权。
- 小程序分享、页面返回、页面销毁后再次进入等流程正常。

常见问题

1. 这是完整的 uni-app 多端 SDK 吗?

不是。本方案只面向 uni-app 编译到微信小程序后的 `mp-weixin` 端。App、H5、支付宝小程序、抖音小程序等端侧不在本方案覆盖范围内。

2. 是否必须由客户小程序完成资质和插件配置?

是。微信服务类目、插件添加、接口权限、隐私协议和审核结果都绑定到客户自己的小程序 AppID，不能由保利威测试 AppID 代替。

3. `viewChange` 事件触发后是否需要宿主页面跳转?

不需要。`viewChange` 只是通知宿主当前内部视图变化。观看流程由 `polyv-watch-room` 内部承载，宿主页面不需要根据 `viewChange` 重新路由或重新挂载观看组件。

4. 为什么点击商品、广告或外部链接时需要宿主处理?

这类行为已经离开观看组件边界，目标页面通常属于客户自己的业务页面。组件会通过事件把链接或业务信息抛给宿主页面，宿主页面再使用 `uni.navigateTo`、`uni.redirectTo` 或自有路由方案处理。

5. 微信开发者工具可以预览，但真机不可用怎么办?

优先检查以下配置：

- 当前预览使用的是否为客户自己的 AppID。
- 微信公众平台后台是否已添加所需插件。
- 小程序服务类目和接口权限是否满足当前功能要求。
- request、socket、upload、download、web-view 等业务域名是否已配置。
- 保利威直播后台是否已配置当前小程序的访问域名或相关开发者信息。
- 真机是否已允许摄像头、麦克风等权限。

相关文档

- 小程序方案选择指南
- 微信小程序观看页 SDK（新版）产品介绍
- 微信小程序观看页 SDK（新版）快速上手
- 保利威云直播小程序播放插件
- 微信观看小程序 SDK 类目申请资质要求
- uni-app 小程序自定义组件支持
- uni-app 微信小程序插件使用说明