

# 智能综窗业务和设计介绍

作者: [LiuShun94](#)

原文链接: <https://ld246.com/article/1669526775280>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 智能综窗系统

## 业务介绍

该系统对接的金赋机器硬件。主要用于市民之家中给群众自行办理政务相关事项使用。其中包含的功能主要有：

1. 市民之家简介。
2. 视频客服坐席。
3. 办件进度查询。
4. 单事项专题办理。
5. 用户登录（小程序登录，人脸登录，身份证登录）三种方式。

为方便群众办理业务，系统采用了上下屏交互设计。下屏主要进行业务相关的办理功能，上屏实时的当前进行的业务进行各种引导提示。





## 代码设计

### 简介

该项目使用的vue3开发的单页面应用。为能管理上下屏的交互通信，采用了lsbridge插件来进行页签的通信。上下屏的业务通过路由进行管理，再开发阶段可以再浏览器分别开启上屏和下屏的路由地址行相应的开发调试工作。

### 代码介绍

上屏代码均管理在 /upper 路由的 upper 目录下，view目录中的其余内容为下屏业务代码。

```

import { createRouter, createWebHistory } from 'vue-router'

const router = createRouter({
  history: createWebHistory(),
  routes: [
    // ... other routes ...
    {
      path: "/upper",
      name: "上屏",
      component: () => import("../views/upper/index.vue"),
      meta: {},
      children: [
        {
          path: "",
          redirect: "/upper/welcome"
        },
        {
          path: "/upper/welcome",
          name: "02005",
          component: () => import("../views/upper/welcome.vue"),
        }
      ]
    }
  ]
})

export default router
  
```

主要介绍下视频坐席模块，因为是金赋提供的服务（vue2开发的插件）。没法直接以插件形式引入到vue3的项目中。所以采用的是以iframe嵌入的方式加入的。以蔡甸项目举例。部署路径为：

- 下屏 <https://10.140.41.251/iterminal/home>
- 上屏 <https://10.140.41.251/iterminal/upper>
- 视频坐席 <https://10.140.41.251/kamfuVideo>

注：视频通话需要需要使用浏览器调起麦克风和摄像头所以需要使用https协议才行。坐席后台管理地址：<https://10.41.215.239/office/#/agentSide/agentSideManager>

## 硬件介绍

金赋的机器使用的 win7 32位（部分为64位）系统。开机时硬件服务会自动启动在5005端口，具体口地址为<http://127.0.0.1:5005/Device>不同硬件带上对应参数传参。

上下双屏可理解为window的两个屏幕。目前设置的上屏作为主屏使用(因部分本地服务如人脸摄像头高拍仪会直接再主屏中开启窗口。接口也提供了相应的修改大小和位置的属性，如有需要可自行尝试置)。

目前业务使用到硬件有 身份证对卡器，双目摄像头，拾音器，高拍仪，文件打印机，小票打印机。

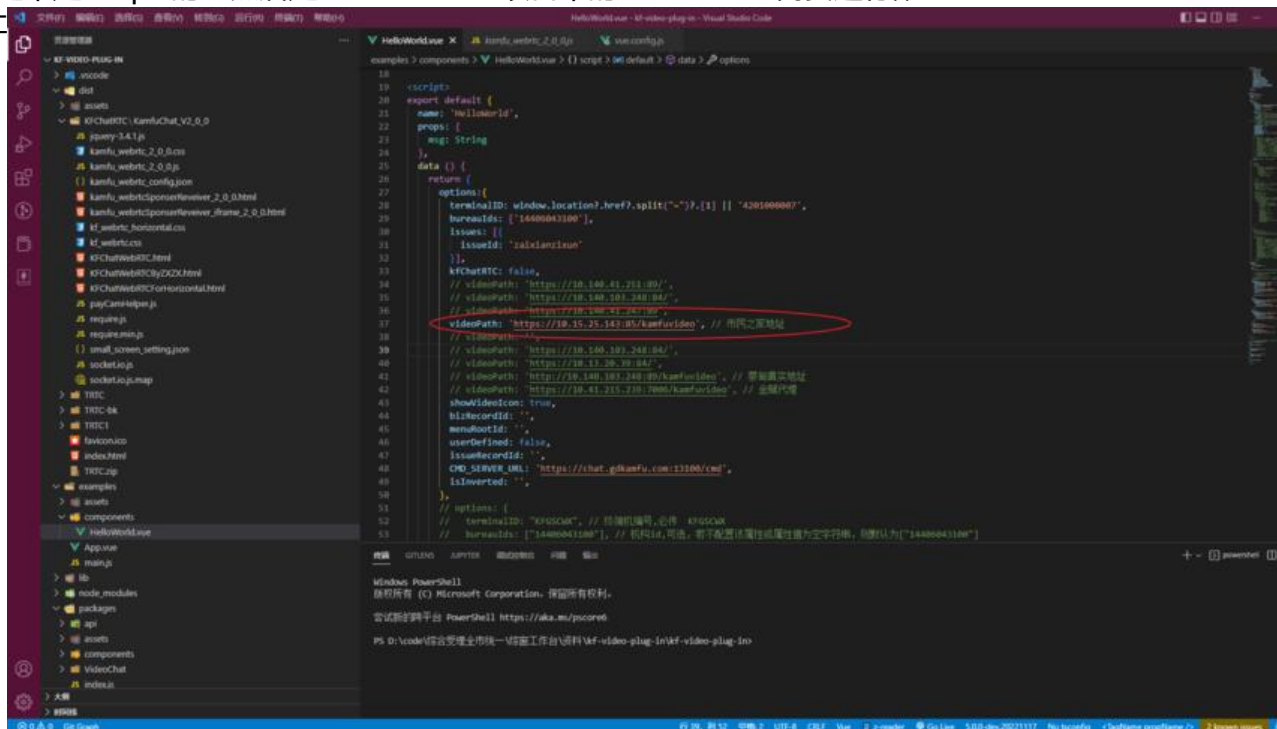
## 常见问题解决方案

### 上下屏通信失败，无法通信

插件 Isbridge.js 中的 send 方法会过滤掉重复参数，只触发一次。解决方法是在 send 中加入一个时戳属性，确保每次的传参不会重复。

### 视频坐席连接不上

先看部署地址是不是 https 的。然后是 kamfuvideo 项目中的 videoPath 需要进行配置 iterminal 务的https地址



因为是vue2的项目，部分插件都是旧版的。打包需要使用14版本及以下的nodejs安装依赖和打包。

## 打包成exe应用

最早对接时是使用的金赋的自带的容器，那时候只用部署一个web地址就可以了，上下屏通信也是使用的容器内部的服务，但是当时存在时不时通信无法接收到的bug,以及容器本身运行的chrome浏览器本很久才75版，无法升级，旧版浏览器无法打开pdf文件等其他浏览器兼容问题。后续版本迭代中已弃该方法。（介绍这个情况是想说明为何会使用 electron 来打包以及自己加入上下屏通信，若后金赋方面修复了相关bug，人员在进行版本迭代时可自行抉择是否对接第三方的服务）

使用 electron 打包的 exe 文件，根据硬件不同的win版本，选择其对应的版本打包参数。为了避免复打包，再生成的应用文件中加入了 nodejs 读取配置文件地址的操作，可在打包后的根目录下加入 config.json 文件

```
{  
  url: https://10.140.41.251:89/iterminal/home ,  
  url1: https://10.140.41.251:89/iterminal/upper  
}
```

配置后exe在打开时会按设定的屏幕位置自动开启配置中的两个浏览器页签应用。