这份文档是基于您提供的最新代码（main.go, tool.go, index.html）编写的完整开发与部署指南。

# 📺 视频播放器与验证系统 - 开发部署文档

## 1. 项目概述

这是一个基于 **Go + WebView (Edge/Chromium)** 开发的桌面应用程序。

* **功能**：播放本地/网络视频 (m3u8)、图片、PDF，支持拖拽播放。
* **特性**：无边框 UI、自定义标题栏颜色、老板键、置顶模式。
* **安全**：包含 MySQL 联网授权验证、本地 License 加密缓存、数据库连接字符串混淆。

## 2. 环境准备

在开始编译之前，请确保您的开发环境满足以下要求：

* **Go 语言环境**: Go 1.16 或更高版本。
* **CGO 编译器 (Windows)**: 因为使用了 webview 库，必须安装 **MinGW-w64** (推荐安装 TDM-GCC 或 MSYS2)，确保 gcc 命令可用。

## 3. 🛡️ 核心配置：数据库连接加密 (必须步骤)

为了防止反编译直接看到数据库账号密码，本项目使用了 **XOR 混淆** 机制。你不能直接在 main.go 里写明文密码。

### 第一步：准备数据库连接串

确定你的 MySQL 连接字符串（DSN），格式如下：

用户名:密码@tcp(IP地址:端口)/数据库名?charset=utf8mb4&parseTime=True&loc=Local

### 第二步：生成加密代码

1. 打开 tool.go 文件。
2. 找到 realDSN 变量，修改为你**真实**的连接字符串。
3. 确保 tool.go 中的 key 变量与 main.go 中的 xorKey 保持一致（默认都是 "MyObfuscationKey2025"）。

Go

// tool.go 示例修改  
func main() {  
 // 修改这里为你的真实连接  
 realDSN := "root:123456@tcp(127.0.0.1:3306)/yjzs?charset=utf8mb4..."  
   
 // 确保这个 Key 和 main.go 里的一样  
 key := []byte("MyObfuscationKey2025")   
 // ...  
}

1. 在终端运行工具：  
   Bash  
   go run tool.go
2. 工具会在终端输出一段 Go 代码（var dbDsnSecret = []byte{...}）。

### 第三步：替换到主程序

1. 复制 tool.go 输出的完整代码块。
2. 打开 main.go。
3. 找到 // ⚠️⚠️⚠️ [重要] 保持原有的数据库配置 ⚠️⚠️⚠️ 下方的 var dbDsnSecret 变量。
4. **删除旧的 dbDsnSecret，粘贴你刚刚生成的新代码。**

## 4. ⚙️ 其他参数配置 (main.go)

在 main.go 的 // --- 配置区域 --- 部分，你可以调整以下参数：

|  |  |  |
| --- | --- | --- |
| **变量名** | **说明** | **建议** |
| SecretKey | 用于加密本地 license.dat 文件的 AES 密钥 (32字节)。 | **发布前修改**，随便改几个字符即可，增加安全性。 |
| AppName | 应用程序名称，用于生成配置文件夹路径。 | 可修改为你的产品英文名，如 MyPlayer。 |
| xorKey | 数据库混淆密钥。 | 如果修改了这里，必须同步修改 tool.go 并重新生成 DSN。 |

## 5. 🗄️ 数据库表结构

确保你的 MySQL 数据库中存在 app\_licenses 表，否则程序无法验证。

SQL

CREATE TABLE `app\_licenses` (  
 `id` int(11) NOT NULL AUTO\_INCREMENT,  
 `code` varchar(64) NOT NULL COMMENT '激活码',  
 `expire\_date` datetime NOT NULL COMMENT '过期时间',  
 `is\_banned` tinyint(1) DEFAULT '0' COMMENT '是否封禁: 1=封禁, 0=正常',  
 PRIMARY KEY (`id`),  
 UNIQUE KEY `idx\_code` (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
-- 插入测试数据  
INSERT INTO `app\_licenses` (`code`, `expire\_date`, `is\_banned`)   
VALUES ('TEST-KEY-1234', '2030-12-31 23:59:59', 0);

## 6. 📦 编译与打包 (Build)

Go 编译出来的 .exe 默认会带一个黑色控制台窗口。发布正式版时需要隐藏它。

### 常用编译命令

**1. 调试编译 (带控制台，方便看日志)**

Bash

go build -o MyPlayer\_Debug.exe main.go

2. 正式发布编译 (无控制台，体积更小)

使用 -ldflags "-H windowsgui -s -w" 参数：

* -H windowsgui: 隐藏黑框（控制台）。
* -s -w: 去掉调试符号，减小体积，增加反编译难度。

Bash

go build -ldflags "-s -w -H windowsgui" -o MyPlayer.exe main.go

### 添加程序图标 (可选)

如果想给 .exe 加上图标，需要使用 rsrc 工具生成资源文件。

1. 准备一个图标文件 icon.ico。
2. 安装工具：go install github.com/akavel/rsrc@latest
3. 生成资源对象：rsrc -manifest MyPlayer.manifest -ico icon.ico -o rsrc.syso (如果没有 manifest 可忽略 -manifest)
4. 再次运行上面的 go build 命令，Go 会自动把目录下生成的 rsrc.syso 打包进去。

## 7. 🚀 部署核对清单

发布给用户前，请检查：

1. [ ] **数据库账号**：确保 tool.go 填的是只读账号（建议），并已重新生成加密块填入 main.go。
2. [ ] **服务器防火墙**：确保你的云服务器安全组开放了 MySQL 端口（如 23306），并允许用户 IP 访问。
3. [ ] **HTML 更新**：如果你修改了 index.html，直接编译即可（embed 特性会将 HTML 打包进 EXE，无需附带 html 文件）。
4. [ ] **依赖文件**：用户电脑不需要安装 Go，但如果是在全新的 Windows 7 上运行，可能需要安装 WebView2 Runtime（Win10/11 自带）。

## 8. 疑难解答

* **编译报错 constant -16 overflows uintptr**：
  + 这是 Go 版本的类型检查问题。
  + **解决**：你提供的最新代码中 SetTitleColor 函数已经包含了 gwlStyle := int(GWL\_STYLE) 的修复逻辑，直接编译即可，不会报错。
* **运行后白屏**：
  + 检查 index.html 是否被修改坏了。
  + 右键 -> Inspect (如果未禁用右键) 查看 Console 报错。
* **提示“数据库连接初始化失败”**：
  + 检查服务器防火墙。
  + 检查 main.go 里的 dbDsnSecret 是否是用 tool.go 最新生成的。

CREATE TABLE IF NOT EXISTS `app\_licenses` (

`id` int NOT NULL AUTO\_INCREMENT,

`code` varchar(64) NOT NULL COMMENT '激活码',

`expire\_date` date NOT NULL COMMENT '过期时间',

`is\_banned` tinyint DEFAULT '0' COMMENT '是否封禁:1是0否',

PRIMARY KEY (`id`),

UNIQUE KEY `uk\_code` (`code`)

);

-- 插入测试数据

INSERT INTO `app\_licenses` (`code`, `expire\_date`) VALUES ('VIP-8888-8888', '2026-12-31');

CREATE USER 'app\_licenses'@'%' IDENTIFIED BY 'RcEs8Rx4fPXmcjzP';

GRANT SELECT ON yjzs.app\_licenses TO 'app\_licenses'@'%';

FLUSH PRIVILEGES;