

WebOBS 流媒体传输系统开发文档

2023 年 12 月 7 日

软件版本 v1.2.0

目录

1. 概览.....	4
1.1 项目简介	4
1.2 技术概要	4
2. 安装指南	4
2.1 环境要求	4
硬件环境.....	4
软件环境.....	4
2.2 安装步骤.....	4
3. 快速开始指南.....	5
3.1 基本用法	5
部署	5
使用	5
4. 使用手册	6
4.1 功能说明	6
客户端.....	6
服务端.....	6
4.2 界面描述	6
index.....	6
readme	7
streamer	7
receiver	8
5. 开发者文档	8
5.1 代码结构	8

5.2 API 文档.....	9
6. 版本记录	10
7. 许可信息	10

1. 概览

1.1 项目简介

WebOBS 流媒体传输系统是一个完全基于 HTTP3 协议传输的，轻量级的，即插即用的流媒体传输系统。客户端完全基于 Web API 实现。服务端基于 Python 实现，只需简单的几步配制，就能搭建一个简便的实时采集、监控平台。

1.2 技术概要

项目可分为客户端和服务端两部分。客户端整体采用 JavaScript 编写。其中，视频采集功能采用 WebRTC API 编写，编码与解码的功能采用 WebCodecs API 编写，网络传输部分采用 WebTransport API 编写。服务端部分采用 Python 编写，H3 与 QUIC 实现采用 aioquic 库，上层处理应用采用 starlette 库编写。

2. 安装指南

2.1 环境要求

硬件环境

任何运行 AMD64 指令集的 CPU，任意集成或独立显卡，50MB 以上的空闲硬盘控件，2GB 以上的 RAM

软件环境

操作系统需要 Windows 10 或更高，Linux 需要具有 GUI 的分支。Python3.10.8 或更高版本，Chrome 97 或更高版本。

2.2 安装步骤

你可以简单的克隆 main 分支到本地。如果你不太了解 Git 是如何运作的，选择网页上的 Code 按钮下载压缩包到本地也是一种选择。

确保您的电脑已经正确安装 Python，经过测试认为 Python3.10.8 及以上版本都可以正确运行本项目，不保证任何低于此版本的环境能够正常运行。如果您还没有安装 Python，您可以前往 [Download Python | Python.org](https://www.python.org/downloads/) 选择最新的稳定版下载安装。

确保您的电脑中安装了最新版的 Chrome，Edge 在当前是不可接受的，其暂且无法正常接收所需的命令行参数以作启动。

完成上述检查后，在项目根目录打开终端，运行

```
pip install -r requirements.txt
```

确保安装 `pip` 安装进程正常退出无错误，如果安装过程发生了异常，重新运行上述命令只到正常运行完成。

接下来，将您的域名认证证书和密钥分别重命名为 `server.pem`, `server_key.pem`，存放在项目的 `certifi` 文件夹下，覆盖原有密钥。

3. 快速开始指南

3.1 基本用法

部署

完成上述步骤之后，在终端中运行

```
python main.py
```

这个脚本的所有参数都是可选的，其均被赋予了默认值

- `--host default='localhost'`: 服务器运行的主机
- `--prot default=4433`: 服务器运行的端口
- `--certifi_file default='./certifi/server.pem'`: 域名认证证书文件路径
- `--certifi_key default='./certifi/server_key.pem'`: 域名认证私钥文件路径
- `--log default='./log/sl_log.log'`: 加密日志文件，可以用于网络抓包工具分析
- `--retry default=False`: 设定服务器是否尝试重连

如果你不了解上述参数含义，使用默认值即可。

如果一切顺利，你应该可以在终端中看到服务器成果运行的日志。

使用

打开文本编辑器，将 `OpenChrome.ps1` 中的域名与端口更改成你在启动服务器时填写的内容。然后使用 `PowerShell` 运行此脚本。

现在 `Chrome` 应该本被正常唤起，你已经可以看到项目的 `index` 页面了。

现在你可以通过这个窗口输入 `URL` 访问项目提供的服务了，请注意对于服务的访问必须通过脚本所打开的窗口进行，任何直接的访问请求都因项目所使用的强制协议需求而被阻挡。

4. 使用手册

4.1 功能说明

客户端

- 捕获摄像头、任意窗口或显示器画面的视频流。
- 将上述视频流加密推送至服务端。
- 拉取推送至服务端的视频流
- 在本地浏览器实时解密并渲染视频流。

服务端

- 接收客户端发送的视频数据
- 缓存视频数据
- 将加密视频流推送给请求客户端

4.2 界面描述

index



客户端的 index 页面主要提供了项目的简单介绍。

readme



客户端的 **readme** 界面提供了对于项目更加技术性的描述，对于项目相关技术，使用方法，遵循协议等内容都可以从这个页面中获取。

streamer



客户端的 **streamer** 页面提供了获取视频源和推流的功能，点击 **start** 按钮后就可以选择视频源并自动开始推流。



客户端的 receiver 页面提供了拉取视频流并进行渲染的功能。点击 **start** 按钮后开始从服务端接收视频流，点击 **render** 按钮后开始渲染视频。

5. 开发者文档

5.1 代码结构

- application 软件包，存放有关 starlette app 相关的源代码
 - templates 模板文件夹，存放了客户端的交互网页
 - ...
 - init.py 软件包入口
 - app.py 服务端 app 主要代码
 - handles.py 服务端应用媒体流处理接口代码
 - utils.py 服务端应用工具代码
- certifi 文件夹，存放认证文件和密钥用于加密通信
- docs 文件夹，存放项目相关文档
- log 文件夹，存放项目运行日志
- server 软件包，存放有关服务端的源代码
 - type 软件包，存放有关类型原型的源代码

- `init.py` 软件包入口
 - `handler.py` 服务端请求处理接口代码
 - `log.py` 服务端日志处理代码
 - `protocol.py` 服务端协议路由与处理代码
 - `server.py` 服务端服务示例代码
 - `session.py` 服务端请求凭证处理接口代码
- `.gitignore` 项目 Git 忽略文件
 - `LICENSE` 项目遵循的开源协议
 - `main.py` 项目服务端启动入口
 - `OpenChrome.ps1` 项目客户端启动入口
 - `readme.md` 项目自述文档
 - `requirements.txt` 项目服务端环境依赖列表

5.2 API 文档

`/|/index`

- 访问项目起始页。

`/wt/streamer`

- 访问捕获页面，这个页面用于捕获视频源并推流。

`/wt/receiver`

- 访问接收页面，这个页面用于拉取视频流并渲染。

`/wt/push`

- 用于使用 WebTransport 的 `UnidirectionalStream` 进行流推送，不能用于直接访问。

`/wt/get`

- 用于使用 WebTransport 的 `BidirectionalStream` 进行流确认和拉取，不能用于直接访问。

6. 版本记录

- v1.0.0 实现首次双工通信，完成基本功能，完成推拉流的测试。
- v1.1.0 重构工程结构，提高了流的 FPS。
- v1.2.0 重新编写客户端界面，提升人机交互体验，优化代码结构。

7. 许可信息

项目源代码遵循 **Apache 2.0** 协议进行开源，开发者及其合作者(一下统称开发者)保留对项目的软件著作权在内的多项权益。开发者对使用此项目以及任何在此项目基础上分发、修改、重构的项目所产生的影响不负任何责任。开发者对于不遵循协议使用项目者或滥用项目者保留追究法律责任的权力。更多许可细节请查看位于项目文件夹中的 **LICENSE** 许可文件。一切最终解释归主开发者 **IWaterLite** 所有。