XEngine\_StreamMedia 服务文档

目录

[XEngine\_StreamMedia 服务文档 1](#_Toc195085137)

[前言 4](#_Toc195085138)

[阅读者 4](#_Toc195085139)

[概述 4](#_Toc195085140)

[相关模块 4](#_Toc195085141)

[一 技术结构 4](#_Toc195085142)

[1.1 使用方法 4](#_Toc195085143)

[1.2 流协议 4](#_Toc195085144)

[1.2.1 推流协议 4](#_Toc195085145)

[1.2.2 拉流协议 5](#_Toc195085146)

[二 配置环 5](#_Toc195085147)

[2.1 WINDOWS 5](#_Toc195085148)

[2.2 LINUX 5](#_Toc195085149)

[2.3 MacOS 5](#_Toc195085150)

[三 接口协议 5](#_Toc195085151)

[3.1 X流协议 5](#_Toc195085152)

[3.3.1 创建流 6](#_Toc195085153)

[3.3.2 销毁流 6](#_Toc195085154)

[3.3.3 推送流 7](#_Toc195085155)

[3.3.4 拉取流 8](#_Toc195085156)

[3.3.5 通知协议 9](#_Toc195085157)

[3.2 JT1078流协议 9](#_Toc195085158)

[3.3 GB28181流协议 9](#_Toc195085159)

[3.4 转推流协议 9](#_Toc195085160)

[3.4.1 播放 10](#_Toc195085161)

[3.4.2 停止 10](#_Toc195085162)

[3.4.3 列举 11](#_Toc195085163)

[3.5 HTTP管理接口 11](#_Toc195085164)

[3.5.1 推流列表获取 11](#_Toc195085165)

[3.5.2 拉流列表获取 12](#_Toc195085166)

[四 配置说明 13](#_Toc195085167)

[4.1 服务器配置 13](#_Toc195085168)

[4.1.1 基本配置 13](#_Toc195085169)

[4.1.2 最大配置 13](#_Toc195085170)

[4.1.3 时间配置 14](#_Toc195085171)

[4.1.4 拉流配置 14](#_Toc195085172)

[4.1.5 日志配置 14](#_Toc195085173)

[5.1.6 报告配置 14](#_Toc195085174)

[五 高级说明 15](#_Toc195085175)

[5.1 GOP缓存 15](#_Toc195085176)

[5.2 优先拉流 15](#_Toc195085177)

[5.3 流转换 15](#_Toc195085178)

[5.4 M3U8 15](#_Toc195085179)

[附录 15](#_Toc195085180)

[附录 更新历史 15](#_Toc195085181)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布 | 文件名称： | XEngin\_StreamMedia 服务文档 | |
| 适用版本： | V2.11.0.1001 | |
| 发布日期： | 2025-04-09 | |
| 拟 制： qyt | | |

# 前言

## 阅读者

开发人员,测试人员,QA

## 概述

此文档包含了存储服务的相关技术说明和接口定义!

## 相关模块

此服务使用了XEngine作为开发包.要使用此服务代码必须配置安装好XEngine开发环境.

# 一 技术结构

服务器基于C/C++实现,协议使用TCP实现

## 使用方法

可以使用OBS或者FFMPEG推流,目前视频仅支持H264,音频AAC,H265视频将在后面支持

SRT推拉流不限制H264或者H265

## 流协议

### 1.2.1 推流协议

RTMP推流:rtmp://127.0.0.1/live/qyt

SRT推流:srt://127.0.0.1:5603?streamid=#!::h=live/qyt,m=publish

XStream推流:需要专用协议推流

JT1078推流:仅仅接受1078数据流,不处理命令,通过命令设置到我们服务器即可

### 1.2.2 拉流协议

FLV OVER HTTP:http://127.0.0.1:5600/api?stream=play&sms=live/qyt&type=flv

TS OVER HTTP拉流:http://127.0.0.1:5600/api?stream=play&sms=live/qyt&type=ts

RTSP拉流:rtsp://127.0.0.1:5600/api?stream=play&sms=live/qyt&type=rtsp

RTMP拉流:rtmp://127.0.0.1/live/qyt

SRT拉流:srt://127.0.0.1:5603?streamid=#!::r=live/qyt,m=request

XStream Over HTTP拉流:http://127.0.0.1:5601?api?stream=play&sms=live/qyt&type=xstream

# 二 配置环

## 2.1 WINDOWS

在完成环境配置后.你可以进入代码目录.使用VS打开XEngine.sln,然后编译.

如果环境没有错误,编译会直接成功

成功后,你需要拷贝XEngine\_Release下的文件到你编译的目录下.运行XEngine\_StreamMediaApp.exe即可.

注意：你可以直接运行程序，系统会提示你需要哪些，你直接进入XEngine目录搜索.也可以使用VSCopy-\*.bat脚本复制到你编译目录下运行,脚本会自动拷贝依赖模块

## 2.2 LINUX

如果使用LINUX来开发运行,那么你需要在UBUNTU或者RockyLinux下面才可以使用,UBUNTU需要22.04 以上系统.RockyLinux需要9.x版本(兼容所有类centos版本)

## 2.3 MacOS

MacOS需要13以上系统的x64 位系统

# 三 接口协议

具体协议可以参考头文件XEngine\_UserProtocol.h

## 3.1 X流协议

此协议是XEngine的流媒体推送拉流协议.非常方便简单

### 3.3.1 创建流

#### 3.3.1.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQCREATE

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

通过tszSMSAddr来指定推流地址

#### 3.3.1.2 回复

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REPCREATE

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

//回复内容为请求内容

### 3.3.2 销毁流

#### 3.3.2.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQDESTROY

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

XENGINE\_PROTOCOLSTREAM

#### 3.3.2.2 回复

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REPDESTROY

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

//回复内容为请求内容

### 3.3.3 推送流

#### 3.3.3.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQPUSH

unPacketSize = sizeof(XENGINE\_PROTOCOLDATA) + AVDATA

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:音视频编码数据

#### 3.3.3.2 回复

无

### 3.3.4 拉取流

用于从流媒体服务器拉取数据流,开始播放前,这个是必须的

#### 3.3.4.1 请求

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken =

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REQPULL

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

XENGINE\_PROTOCOLSTREAM

#### 3.3.4.2 回复

wReserver字段为0,表示成功,其他值表示失败.失败的值如下:

1:没有找到这个推流的ID

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 流媒体推流唯一ID

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_REPXPULL

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:根据请求回复

### 3.3.5 通知协议

通知协议用于处理推送拉流状态,告知另一方目前接受数据和处理数据情况,让发送端或者接收端调整发送频率!

拉流协议和推流协议是一样的,只是CODE值不同,下面不在另外演示.

另外注意:推流协议不用用于拉流,同样的,拉流通知协议不能用于推流

协议头:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 流媒体推流唯一ID

unOperatorType = ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_SMS

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_SMS\_STATPUSH

unPacketSize = sizeof(XENGINE\_PROTOCOLSTREAM) + sizeof(XENGINE\_SMSPROTOCOL)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:sizeof(XENGINE\_PROTOCOLSTREAM) + sizeof(XENGINE\_SMSPROTOCOL)

## 3.2 JT1078流协议

参考交通部平台定制的JT1078-2014(2016)文档

## 3.3 GB28181流协议

参考国标GB/T28181-2016文档

## 3.4 转推流协议

token是自己控制或者由系统生成,在你停止的时候需要这个token才能停止,否则这个流会一直存在并且造成可能多路同流问题.

此协议暂时不可用

### 3.4.1 播放

#### 3.4.1 请求

接口:HTTP

方法:POST

地址:http://127.0.0.1:5602/api?function=play&token=100010001

参数:function:操作方式

Token:流唯一编号

负载:RTSP或者其他协议播放地址

{

"tszAVUrl":"rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny\_115k.mp4"

}

#### 3.4.2 回复

{

"code":0,

"msg":"success",

"token":"100010001"

}

### 3.4.2 停止

#### 3.4.2.1 请求

接口:HTTP

方法:POST

地址:http://127.0.0.1:5602/api?function=stop&token=10001001

参数:function:操作方式

Token:流唯一编号

#### 3.4.2.2 回复

{

"code":0,

"msg":"success"

}

### 3.4.3 列举

可以枚举所有正在播放的列表信息

#### 3.4.3.1 请求

接口:HTTP

方法:GET

地址:http://127.0.0.1:5602/api?function=list

参数:function:操作方式

#### 3.4.3.2 回复

{

"Array":[

{

"tszToken":"100010001",

"tszAVUrl":"rtsp://wowzaec2demo.streamlock.net/vod/mp4:BigBuckBunny\_115k.mp4"

}

],

"Count":1,

"code":0,

"msg":"success"

}

## 3.5 HTTP管理接口

### 3.5.1 推流列表获取

#### 3.5.1.1 请求

接口:HTTP

方法:GET

地址:http://127.0.0.1:5600/api?function=publish&method=get

参数:function:操作方式

#### 3.5.1.2 回复

{

"Array":[

{

"enStreamType":3,

"nClientCount":0,

"st\_AudioInfo":{

"bEnable":true,

"enAVCodec":10,

"nBitRate":160,

"nChannel":2,

"nSampleFmt":16,

"nSampleRate":48000

},

"st\_VideoInfo":{

"bEnable":true,

"enAVCodec":7,

"nBitRate":5950,

"nFrameRate":30,

"nHeight":1080,

"nWidth":1920

},

"tszSMSAddr":"live/qyt"

}

],

"code":0,

"msg":"success"

}

### 3.5.2 拉流列表获取

#### 3.5.2.1 请求

接口:HTTP

方法:GET

地址:http://127.0.0.1:5600/api?function=pull&method=get

参数:function:操作方式

#### 3.5.2.2 回复

{

"Array":[

{

"enStreamType":17,

"tszPushAddr":"127.0.0.1:50795",

"tszSMSAddr":"live/qyt"

}

],

"code":0,

"msg":"success"

}

# 四 配置说明

## 4.1 服务器配置

基础配置文件:XEngine\_Config.json

### 4.1.1 基本配置

* tszSMSUrl:推流地址
* tszIPAddr:本机IP地址
* bDeamon: 1为守护进程后台运行0为前台运行
* nRTMPPort:RTMP推流服务器
* nHttpPort:HTTP拉流服务器
* nXStreamPort:XStream通信端口
* nJT1078Port:JT 1078 通信端口
* nSrtPort:srt端口
* nRTCPort:RTC端口

### 4.1.2 最大配置

XMax 配置

* nMaxClient 最大允许客户端
* nMaxQueue 最大允许队列
* nIOThread:网络IO处理线程个数
* nHttpThread:HTTP处理线程个数
* nXStreamThread:XStream处理线程个数
* nRTMPThread:RTMP处理线程个数
* nJT1078Thread:1078协议处理线程个数

### 4.1.3 时间配置

XTime配置

* nTimeCheck:检测次数
* nHttpTimeOut:HTTP客户端过期时间
* nXStreamTimeOut:XStream客户端过期时间
* nRTMPTimeout:RTMP客户端超时时间
* nJT1078Timeout:1078客户端超时时间
* nRTCTimeout:rtc 客户端超时时间

### 4.1.4 拉流配置

XPull配置,是否启用和是否启用预拉流支持

* XStream:是否启用XStream拉流支持
* RTMP:是否启用RTMP拉流支持
* FLV:是否启用FLV拉流支持
* RTSP:是否启用RTSP拉流支持
* RTC:是否启用RTC拉流支持,并且设置绑定的端口
* SRT:是否开启SRT转流拉流支持
* TS:是否开启TS OVER HTTP拉流支持
* HLS:是否开启M3U8拉流支持,并且配置是否关闭后清理以及时长和路径

### 4.1.5 日志配置

XLog 配置

* LogFile:日志保存路径
* MaxSize:日志文件大小
* MaxCount:日志文件个数
* LogLeave:允许保存的级别
* LogType:输出的日志类型

### 5.1.6 报告配置

XReport 配置

* bEnable:是否启用
* tszServiceName:服务名称
* tszAPIUrl:报告地址

# 五 高级说明

## 5.1 GOP缓存

GOP缓存目前并不支持,所以在客户端拉流播放的时候会出现黑屏或者花屏问题,当遇到I帧后就可以正常播放了.

也就是说目前是没有延迟的.

GOP缓冲将在以后实现.

## 5.2 优先拉流

现在支持预拉流了,预拉流可以通过每个Pull的配置开关,一般名称为bPre开头.

## 5.3 流转换

流转换作为XStream的核心功能之一,其意义重大,他可以把推流的SRT,XSTREAM,RTMP等流协议转换为其他拉流协议,比如他们都可以转为XStream,RTMP,Srt,HTTP-FLV等等,方便用户处理推拉流端协议限制的问题

## 5.4 M3U8

目前流媒体服务器暂时不支持直接拉M3U8,仅仅支持转换,大家可以使用第三方工具,比如NGINX来拉流.

时长设置为15秒不一定是15秒,关键看关键帧位置,一般可能会延长个几秒

# 附录

## 附录 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新说明 | 修改者 | 适用版本 |