XEngine\_StreamMedia 服务文档

目录

[XEngine\_StreamMedia 服务文档 1](#_Toc724)

[前言 3](#_Toc22200)

[阅读者 3](#_Toc22354)

[概述 3](#_Toc8861)

[相关模块 3](#_Toc21410)

[一 技术结构 3](#_Toc22867)

[二 配置环境 3](#_Toc18868)

[2.1 XEngine环境 3](#_Toc20776)

[2.2 WINDOWS 4](#_Toc23089)

[2.2.2 编译运行 4](#_Toc9556)

[2.2 LINUX 4](#_Toc997)

[2.2.1 环境配置 4](#_Toc4151)

[2.2.2 编译运行 4](#_Toc31588)

[2.3 版本要求 4](#_Toc12516)

[2.3.1 系统版本 4](#_Toc29951)

[2.3.2 软件需求 4](#_Toc24009)

[三 接口协议 5](#_Toc6985)

[3.1 X流协议 5](#_Toc30116)

[3.3.1 推送协议 5](#_Toc32429)

[3.3.2 拉流协议 6](#_Toc26931)

[3.3.3 控制协议 7](#_Toc28400)

[3.3.4 通知协议 8](#_Toc19255)

[3.2 JT1078流协议 9](#_Toc30104)

[3.3 GB28181流协议 9](#_Toc25030)

[四 配置说明 9](#_Toc8318)

[4.1 服务器配置 9](#_Toc19202)

[4.1.1 基本配置 9](#_Toc1570)

[4.1.2 最大配置 10](#_Toc6316)

[4.1.3 时间配置 10](#_Toc31101)

[4.1.4 数据库配置 10](#_Toc8495)

[4.1.5 日志配置 10](#_Toc20823)

[4.2 JT1078服务 10](#_Toc31251)

[4.2.1 基本配置 10](#_Toc1910)

[4.2.2 最大配置 11](#_Toc28408)

[4.2.3 时间配置 11](#_Toc22048)

[4.2.4 客户端配置 11](#_Toc8590)

[附录 11](#_Toc28354)

[附录 更新历史 11](#_Toc8738)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布 | 文件名称： | XEngin\_StreamMedia 服务文档 | |
| 适用版本： | V1.0.0.1001 | |
| 发布日期： | 2022-04-29 | |
| 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

# 二 配置环境

## 2.1 XEngine环境

需要下载XEngine.

下载地址:https://gitee.com/xyry/libxengine

https://github.com/libxengine/xengine

按照XEngine Readme文件的说明配置环境

## 2.2 WINDOWS

### 2.2.2 编译运行

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\_\*App.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脚本拷贝依赖模块

## 2.2 LINUX

### 2.2.1 环境配置

如果使用LINUX来开发运行,那么你需要在UBUNTU或者CENTOS下面才可以使用,UBUNTU需要20.04 以上系统.CENTOS需要8.x版本(兼容所有类centos版本)

### 2.2.2 编译运行

配置完环境后,你可以编译它,打开终端,进入源码目录执行以下命令

编译:make

安装:make FLAGS=InstallAll

清理:make FLAGS=CleanAll

如果没有错误,你可以在XEngine\_Release目录下看到编译的XEngine\_\*App

## 2.3 版本要求

### 2.3.1 系统版本

最低版本要求:

WINDOWS: win7 sp1

Ubuntu:20.04

Centos:8.x

### 2.3.2 软件需求

最低版本要求:

XEngine:V7.34

# 三 接口协议

## 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 推送协议

流媒体推送协议的协议头TOKEN字段必填,在请求的时候填写,表明用户在创建成功后,这个流的唯一ID,这样用户可以清楚的知道自己的推流ID.比如流ID是123,那么你的推送地址就是:Rtsp://192.168.1.10/live/123

#### 3.3.1.1 请求

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\_REQXPUSH

unPacketSize = sizeof(XENGINE\_AVPROTOCOL)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

//音视频参数协议

XENGINE\_PROTOCOL\_AVINFO

#### 3.3.1.2 回复

wReserve:如果此值为0表示成功.其他值表示失败,参考如下信息:

* 1:协议验证失败
* 2:不是推流协议
* 3.没有找到SPS或者PPS信息,这个错误在推流第一个包如果出错会出现,如果没有错误,不会继续推送,也就是说,如果有错误,你会收到两个包头

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\_REPXPUSH

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.3.2 拉流协议

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

#### 3.3.2.1 请求

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\_REQXPULL

unPacketSize = 0

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 3.3.2.2 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\_AVPROTOCOL)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

//音视频参数协议

typedef struct tag\_XEngine\_AVProtocol

{

*CHAR* tszPktName[4]; //封装格式,如果没有,可以为NULL,封装格式为后缀.比如:mkv flv mp4

*BYTE* byPktFlag; //数据包封装格式,发送的音视频流后续格式,0:无协议裸流数据,1:标准协议头,2:扩展协议头

//视频信息

struct

{

BOOL bEnable; //是否启用

\_\_int64x nBitRate; //码率

int nWidth; //视频宽

int nHeight; //视频高

int enAvCodec; //使用的编码器

int nFrameRate; //帧率

int nVLen; //SPSPPS大小,为0将交由模块处理

*CHAR* tszVInfo[256]; //SPS与PPS,你可以通过AVHelp\_MetaInfo\_Get264Hdr来处理,在SPS和PPS前面添加起始字节后一起拷贝到这里面

}st\_PushVideo;

//音频信息

struct

{

BOOL bEnable; //是否启用

\_\_int64x nBitRate; //码率

int nChannel; //通道个数

int nSampleRate; //采样率.44100 ...

int nSampleFmt; //采样格式,S16 S32...

int nFrameSize; //采样大小

int enAvCodec; //编码器

int nALen; //大小

*CHAR* tszAInfo[256]; //推流扩展编码信息

}st\_PushAudio;

}XENGINE\_PROTOCOL\_AVINFO, \* LPXENGINE\_PROTOCOL\_AVINFO;

### 3.3.3 控制协议

控制类协议没有回复,只有请求,如果发现操作没有成功,可以再次发送,指导成功为止

#### 3.3.3.1 播放控制

只有向服务器请求了此协议,服务器才会推送数据给拉流端

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\_CTRLPLAY

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 3.3.3.2 播放暂停

如果不想接续接受数据,可以调用此函数来暂停服务器的拉流,如果想继续播放,可以调用播放协议

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\_CTRLPAUSE

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.3.4 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!

#### 3.3.4.1 拉流通知协议

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\_SMSPROTOCOL)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

typedef struct tag\_XEngine\_SMSProtocol

{

XNETHANDLE xhXTPTime; //同步时间戳

ULONGLONG nPacketCount; //当前推送(接受)数据大小

ULONGLONG nPacketHandle; //已处理(已发送)数据大小

ULONGLONG nPacketLeft; //剩余数据大小

}XENGINE\_SMSPROTOCOL, \* LPXENGINE\_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

# 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
* nCenterPort:TCP通信端口
* nHttpPort:HTTP通信端口

### 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
* nCenterThread:TCP处理线程个数
* nHttpThread:HTTP处理线程个数

### 4.1.3 时间配置

XTime配置

* nTimeCheck:检测次数
* nCenterTimeOut:TCP客户端过期时间
* nHttpTimeOut:HTTP客户端过期时间

### 4.1.4 数据库配置

XSQL配置,MYSQL服务

* tszSQLAddr:数据库地址
* nSQLPort:数据库端口
* tszSQLUser:用户名
* tszSQLPass:密码

### 4.1.5 日志配置

XLog 配置

* MaxSize:日志文件大小
* MaxCount:日志文件个数
* LogLeave:允许保存的级别

## 4.2 JT1078服务

配置文件:XEngine\_JT1078Config.json

### 4.2.1 基本配置

* tszIPAddr:本机IP地址
* bDeamon: 1为守护进程后台运行0为前台运行
* nAudio:音频通道号,大于0启用
* nStreamPort:直播流端口
* nRecordPort:录像流端口

### 4.2.2 最大配置

XMax 配置

* MaxClient 最大允许客户端
* MaxQueue 最大允许队列
* IOThread:网络IO处理线程个数
* StreamThread:直播流处理线程个数
* RecordThread:录像流处理线程个数

### 4.2.3 时间配置

XTime配置

* nTimeCheck:检测次数
* nStreamTimeout:直播流过期时间
* nRecordTimeout:录像流过期时间

### 4.2.4 客户端配置

XClient配置

* tszIPAddr:服务器地址
* nPort:服务器端口
* nMaxConnect:客户端连接个数

# 附录

## 附录 更新历史

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