XEngine\_StreamMedia 服务文档

目录

[XEngine\_StreamMedia 服务文档 1](#_Toc3397)

[前言 3](#_Toc16809)

[阅读者 3](#_Toc17364)

[概述 4](#_Toc28864)

[相关模块 4](#_Toc23392)

[一 技术结构 4](#_Toc25673)

[二 配置环境 4](#_Toc22627)

[2.1 XEngine环境 4](#_Toc21507)

[2.2 WINDOWS 4](#_Toc3082)

[2.2.2 编译运行 4](#_Toc2858)

[2.2 LINUX 5](#_Toc24669)

[2.2.1 环境配置 5](#_Toc20602)

[2.2.2 编译运行 5](#_Toc15024)

[2.3 版本要求 5](#_Toc28508)

[2.3.1 系统版本 5](#_Toc20132)

[2.3.2 软件需求 5](#_Toc29527)

[2.4 使用方式 5](#_Toc20041)

[三 接口协议 5](#_Toc28739)

[3.1 X流协议 6](#_Toc2070)

[3.3.1 创建流 6](#_Toc17611)

[3.3.2 销毁流 7](#_Toc18897)

[3.3.3 推送流 8](#_Toc29247)

[3.3.4 拉取流 8](#_Toc22438)

[3.3.5 通知协议 9](#_Toc15307)

[3.2 JT1078流协议 9](#_Toc249)

[3.3 GB28181流协议 10](#_Toc4235)

[3.4 转推流协议 10](#_Toc26802)

[3.4.1 播放 10](#_Toc7956)

[3.4.2 停止 10](#_Toc1309)

[3.4.3 列举 11](#_Toc9830)

[四 配置说明 12](#_Toc15249)

[4.1 服务器配置 12](#_Toc30573)

[4.1.1 基本配置 12](#_Toc24182)

[4.1.2 最大配置 12](#_Toc29553)

[4.1.3 时间配置 12](#_Toc18539)

[4.1.4 数据库配置 12](#_Toc30508)

[4.1.5 日志配置 13](#_Toc26001)

[4.2 JT1078服务 13](#_Toc24409)

[4.2.1 基本配置 13](#_Toc17485)

[4.2.2 最大配置 13](#_Toc31483)

[4.2.3 时间配置 13](#_Toc20735)

[4.2.4 客户端配置 13](#_Toc8373)

[4.3 流转发配置 14](#_Toc15484)

[4.3.1 基本配置 14](#_Toc2012)

[4.3.2 最大配置 14](#_Toc19279)

[4.3.3 时间配置 14](#_Toc19250)

[附录 14](#_Toc31754)

[附录 更新历史 14](#_Toc27243)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布 | 文件名称： | XEngin\_StreamMedia 服务文档 | |
| 适用版本： | V1.4.0.1001 | |
| 发布日期： | 2022-12-02 | |
| 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,每个协议有自己的单独处理进程,通过统一的CenterApp来推流到流媒体服务.

XEngine\_CenterApp:推流服务器

XEngine\_JT1078App:JT1078收流服务

XEngine\_GB28181App:GB28181收流服务器

XEngine\_ForwardApp:流转推流服务,拉取其他设备或者服务的流转发到指定服务(推荐)

# 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

### 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

MacOS:12

### 2.3.2 软件需求

最低版本要求:

XEngine:V7.44

## 2.4 使用方式

根据你的需求,你需要启用不同服务,XEngine\_CenterApp为必须启用,其他服务为可选.每个服务有单独的配置文件,需要用户配置成功后才可以使用.

# 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

创建流协议TOKEN字段选填,如果填写,表明这个流的唯一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:http://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\_REQCREATE

unPacketSize = sizeof(XENGINE\_PROTOCOLDEVICE)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

//设备信息,如果没有指定token,那么根据这个字段来组成播放地址:\*/device\_channel\_live,并且不需要XENGINE\_PROTOCOLDEVICE结构.

XENGINE\_PROTOCOLDEVICE

//音视频参数协议.这个选项不是必须的,可以附加也可以不附加,不附加没有音频

XENGINE\_PROTOCOL\_AVINFO

#### 3.3.1.2 回复

只有在reply为1的时候才有回复

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\_REPCREATE

unPacketSize = sizeof(XENGINE\_PROTOCOLDEVICE)

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 = 流媒体推流唯一ID

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOLDEVICE)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

XENGINE\_PROTOCOLDEVICE

#### 3.3.2.2 回复

只有在reply为1的时候才有回复

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\_REPDESTROY

unPacketSize = sizeof(XENGINE\_PROTOCOLDEVICE)

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 = 流媒体推流唯一ID

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

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

unPacketSize = 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 = 流媒体推流唯一ID

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOLDEVICE)

byVersion = 0

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:

XENGINE\_PROTOCOLDEVICE

#### 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\_PROTOCOLDEVICE)

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\_PROTOCOLDEVICE) + sizeof(XENGINE\_SMSPROTOCOL)

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

协议体:sizeof(XENGINE\_PROTOCOLDEVICE) + 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.

此协议适用于XEngine\_ForwardApp

### 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"

}

# 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

## 4.3 流转发配置

配置文件:XEngine\_ForwardConfig.json

### 4.3.1 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
* nHttpPort:HTTP通信端口

### 4.3.2 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
* HttpThread:业务处理线程个数

### 4.3.3 时间配置

XTime配置

* nTimeCheck:检测次数
* nHttpTimeout:Http客户端超时时间

# 附录

## 附录 更新历史

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