XEngine Message Service Docment

目录

[XEngine Message Service Docment 1](#_Toc32599)

[Preface 4](#_Toc13269)

[Reader 4](#_Toc25245)

[Overview 4](#_Toc2350)

[Associate Module 4](#_Toc30128)

[一 Technical structure 4](#_Toc25791)

[1.1 TCP 4](#_Toc14913)

[1.2 HTTP 4](#_Toc20570)

[1.3 WEBSOCKET 4](#_Toc3614)

[1.4 HeartBeat 5](#_Toc18646)

[1.5 Detailed Protocol 5](#_Toc8216)

[二 Configure Env 5](#_Toc6595)

[2.1 WINDOWS 5](#_Toc11611)

[2.1.1 Configure Environment 5](#_Toc13309)

[2.1.2 complie and run 5](#_Toc21062)

[2.2 LINUX 5](#_Toc15756)

[2.2.1 Evnironment Configure 5](#_Toc21317)

[2.2.2 Complie and Run 6](#_Toc19882)

[2.3 MacOS 6](#_Toc9533)

[2.4 Version Requirements 6](#_Toc7663)

[2.4.1 System Version 6](#_Toc31365)

[2.4.2 Software Version 6](#_Toc500)

[三 Interface Protocol 6](#_Toc132)

[3.1 TCP 6](#_Toc16618)

[3.1.1 Post Protocol 7](#_Toc7022)

[3.1.2 Get Protocol 8](#_Toc2286)

[3.1.3 Delete Protocol 9](#_Toc6326)

[3.1.4 Create Topic 9](#_Toc12852)

[3.1.5 Delete Topic 10](#_Toc22055)

[3.1.6 Subscribe Topic 11](#_Toc30772)

[3.1.7 Notification Message 12](#_Toc2765)

[3.1.8 Get Packet Set 12](#_Toc16758)

[3.1.9 Serial Number Get 13](#_Toc2061)

[3.1.10 User Authorize 14](#_Toc4803)

[3.1.11 User Register 15](#_Toc15520)

[3.1.12 User Delete 16](#_Toc28856)

[3.1.13 Message Modify 17](#_Toc20563)

[3.1.14 Topic Modify 18](#_Toc19214)

[3.1.15 UNRead Message 18](#_Toc9899)

[3.2 HTTP 21](#_Toc10009)

[3.2.1 Request 22](#_Toc18411)

[3.2.2 Reply 22](#_Toc11086)

[3.3 WEBSOCKET 23](#_Toc27407)

[四 Configure Description 23](#_Toc31684)

[4.1 Service Configure 23](#_Toc28842)

[4.1.1 basic configure 23](#_Toc6169)

[4.1.2 Max Configure 23](#_Toc24025)

[4.1.3 Time Configure 24](#_Toc11551)

[4.1.4 Log Configure 24](#_Toc8103)

[4.1.5 Database Configure 24](#_Toc8502)

[4.1.6 HTTP Pass Configure 24](#_Toc25223)

[五 Advanced configuration 25](#_Toc6214)

[六 Other Functions 25](#_Toc26340)

[6.1 Reply Configure 25](#_Toc21680)

[6.2 HTTP Protocol 25](#_Toc17097)

[appendix 25](#_Toc17591)

[Appendix 1 Type Define 25](#_Toc16944)

[Appendix 2 Protocol Define 25](#_Toc18473)

[Appendix 3 Transformation Definition 25](#_Toc25220)

[Appendix 4 update log 25](#_Toc22607)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Message Service Docment | |
| Be A Version： | V3.5 | |
| Released： | 2023-01-06 | |
| Writer： qyt | | |

# Preface

## Reader

Development ,test

## **Overview**

This document contains related technical descriptions and interface definitions

## Associate Module

The service used XEngine as Network Toolki.if you want to use code,you have to installed XEngine

And This service used library for jsoncpp

# 一 Technical structure

Should be bind Three port when Start service.these are tcp message port and http message port and websocket port

## TCP

The TCP protocol uses the standard protocol header of XEngine to realize message communication. The TCP protocol is faster and has more functions.

## HTTP

Send and recv Message through post.You only need to send the URL of the address and port, and then load the JSON. HTTP does not support subscription mode

## WEBSOCKET

Websocket can provide message support for the HTTP front end, using the standard WEBSOCKET protocol, and the type using TEXT. WEBSOCKET messages can be realized with JSON with the same load as HTTP, and WEBSOCKET messages support subscription mode.

## HeartBeat

For session keepalive,The heartbeat must be sent at a fixed time.

## 1.5 Detailed Protocol

The TCP private protocol adopts the protocol defined by XEngine. You can refer to the protocol document of XEngine to learn more about the definition and interpretation of the protocol

# 二 Configure Env

## 2.1 WINDOWS

Need to download XEngine.

Complie and run and debug by vs.

### 2.1.1 Configure Environment

Follow the instructions in the XEngine Readme file to configure the environment.

### 2.1.2 complie and run

When you complete with configuration.you can come in code path.open XEngine\_MQServiceApp.sln by vs

If environment not have error.complie is succesed.contrain 5 module and 1 exe program

And you need copy file under XEngine\_Release to your complied dir.next step copy file under XEngine depend module to your complied dir.

Note: Xengine environment can be copied to your compilation directory through vscopy script, provided that you configure your xengine environment

## 2.2 LINUX

### 2.2.1 Evnironment Configure

If you use linux.you must running on ubuntu(20.04) or centos(8.x)...

### 2.2.2 Complie and Run

Configure complete.you can complie it.open terminal in you xengine\_storage dir and execute command.

complie:make

install:make FLAGS=InstallAll

clean:make FLAGS=CleanAll

If there is no error.you can see complied XEngine\_MQServiceApp file in XEngine\_Release

You can running at terminal.

## 2.3 MacOS

MacOS requires 12 or above versions, and the compilation and operation mode can refer to linux

## 2.4 Version Requirements

### 2.4.1 System Version

Minimum version requirements:

WINDOWS: win7 sp1

Ubuntu:20.04

Centos:8.x

MacOS:12

### 2.4.2 Software Version

Minimum version requirements:

XEngine:V7.38

# 三 Interface Protocol

## 3.1 TCP

Please Note:The wReserver field of the protocol header will be used as the server reply processing result. 0 means the processing is successful, other values mean failure  
 whether protocol load is general protocol or HTTP protocol,they can be load json data.

### 3.1.1 Post Protocol

To use the message queue, you need to deliver a packet to the message queue server first, so that other programs can get a packet from the message queue service.

#### 3.1.1.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQPOST

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ) + MSGLEN

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

DATA in tail

nPubTime mean is a timed message, only users who subscribe to this message and are online will receive it

byVersion mean is payload format,refer :ENUM\_XENGINE\_PROTOCOLHDR\_PAYLOAD\_TYPE

#### 3.1.1.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPPOST

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

### 3.1.2 Get Protocol

#### 3.1.2.1 Requestion

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQGET

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.2.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPGET

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ) + MSGLEN

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

Protocol Body:if sucess,protocol follow get data

byVersion mean is payload format,refer :ENUM\_XENGINE\_PROTOCOLHDR\_PAYLOAD\_TYPE

### 3.1.3 Delete Protocol

#### 3.1.3.1 Requestion

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQDEL

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.3.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPDEL

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

### 3.1.4 Create Topic

Post a packet to message queue,you must create a topic,you have to set topic name,data can be enter the message queue

#### 3.1.4.1 Requestion

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.4.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

### 3.1.5 Delete Topic

#### 3.1.5.1 Requestion

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQDELETE

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.5.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPDELETE

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

### 3.1.6 Subscribe Topic

Subscribing to the topic allows the user to actively push a message to the subscribed client every time there is new content in the topic after the request is successfully subscribed

#### 3.1.6.1 Requestion

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQNOTIFY

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.6.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPNOTIFY

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

### 3.1.7 Notification Message

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_MSGNOTIFY

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ) + message

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

...body

### 3.1.8 Get Packet Set

Support jump to the specified serial number to start

#### 3.1.8.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQSERIAL

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.8.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPSERIAL

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

### 3.1.9 Serial Number Get

Can be get specified topic count of number,start message number and tail message number

#### 3.1.9.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQSERIAL

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

Struct XENGINE\_PROTOCOL\_XMQ

#### 3.1.9.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQSERIAL

unPacketSize = sizeof(XENGINE\_MQNUMBER)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct tag\_XEngine\_MQNumber

{

TCHAR tszMQKey[256]; //topic name

\_\_int64x nCount; //total number

\_\_int64x nFirstNumber; //start number

\_\_int64x nLastNumber; //tail number

}XENGINE\_MQNUMBER, \* LPXENGINE\_MQNUMBER;

### 3.1.10 User Authorize

#### 3.1.10.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_AUTH\_REQLOGIN

unPacketSize = sizeof(XENGINE\_PROTOCOL\_USERAUTH)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct tag\_XEngine\_Protocol\_Auth

{

CHAR tszUserName[64]; //User

CHAR tszUserPass[64]; //Pass

ENUM\_PROTOCOLCLIENT\_TYPE enClientType; //user type

ENUM\_PROTOCOLDEVICE\_TYPE enDeviceType; //device type

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

#### 3.1.10.2 Reply

Protocol Header:wReserve 0 is sucess,ohter fail

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_AUTH\_REPLOGIN

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.1.11 User Register

#### 3.1.10.1 Rqeuest

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQUSERREG

unPacketSize = sizeof(*XENGINE\_PROTOCOL\_USERINFO*)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

*XENGINE\_PROTOCOL\_USERINFO*

#### 3.1.10.2 Reply

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPUSERREG

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.1.12 User Delete

#### 3.1.10.1 Rqeuest

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQUSERDEL

unPacketSize = sizeof(*XENGINE\_PROTOCOL\_USERINFO*)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

*XENGINE\_PROTOCOL\_USERINFO*

#### 3.1.10.2 Reply

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPUSERDEL

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.1.13 Message Modify

#### 3.1.13.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_XMQ*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQMSGMODIFY

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:Determine the modified message based on the message Key and sequence number

XENGINE\_PROTOCOL\_XMQ

#### 3.1.13.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_XMQ*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPMSGMODIFY

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.1.14 Topic Modify

#### 3.1.14.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_XMQ*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQTOPICMODIFY

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ) + topicname

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:The XMQ protocol only needs to fill in the topic name, followed by the string of the topic name to be modified at the end

XENGINE\_PROTOCOL\_XMQ + tocpiname

#### 3.1.14.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_XMQ*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPTOPICMODIFY

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.1.15 UNRead Message

#### 3.1.15.1 Request

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_XMQ*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REQUNREAD

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:just need topic name,or can be not fill mean is get all

XENGINE\_PROTOCOL\_XMQ

#### 3.1.15.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

unOperatorType = *ENUM\_XENGINE\_COMMUNICATION\_PROTOCOL\_TYPE\_XMQ*

unOperatorCode = XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_MQ\_REPUNREAD

unPacketSize = JSON

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

{

"Array":[

{

"Array":[

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":9,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 10:46:13",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey",

"tszQueuePublishTime":""

},

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":10,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 10:46:13",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey",

"tszQueuePublishTime":""

}

],

"Count":2,

"Name":"XEngine\_NotifyKey"

},

{

"Array":[

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":6,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 15:10:43",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey2",

"tszQueuePublishTime":""

},

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":7,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 15:10:44",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey2",

"tszQueuePublishTime":""

},

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":8,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 15:10:45",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey2",

"tszQueuePublishTime":""

},

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":9,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 15:10:46",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey2",

"tszQueuePublishTime":""

},

{

"byMsgType":5,

"nMsgLen":12,

"nQueueGetTime":0,

"nQueueSerial":10,

"tszMsgBuffer":"123456789aaa",

"tszQueueCreateTime":"2023-01-04 15:10:46",

"tszQueueLeftTime":"",

"tszQueueName":"XEngine\_NotifyKey2",

"tszQueuePublishTime":""

}

],

"Count":5,

"Name":"XEngine\_NotifyKey2"

}

],

"Count":2

}

## 3.2 HTTP

HTTP support post requestion now.make sure event through json

HTTP uses TOKEN to keep session, and the TOKEN obtained after logging in must be brought with each request

Header field same to tcp protocol

st\_MQProtocol Only exists when the message is loaded

st\_Auth Only exists when you log in

st\_User Exists when registering and deleting users

st\_Payload mean is load data

If no have payload,,st\_Payload can not be exist.

wReserve feild Must existed When subscribe message protocol

### 3.2.1 Request

{  
    **"unOperatorType"**:**7**,  
    **"unOperatorCode"**:**7001**,  
    **"wReserve"**:**1**,  
    **"byVersion"**:**2**,  
    **"st\_MQProtocol"**:{  
        **"tszMQKey"**:**"主题名"**,  
        **"nSerial"**:**0**,  
        **"nKeepTime"**:**0**,  
        **"nGetTimer"**:**0**,  
        **"nPubTime"**:**0**  
    },  
    **"st\_Auth"**:{  
        **"tszUserName"**:**"123123aa"**,  
        **"tszUserPass"**:**"123123"**,  
        **"enClientType"**:**10**,  
        **"enDeviceType"**:**20**  
    },  
    **"st\_User"**:{  
        **"nUserLevel"**:**0**,  
        **"nUserState"**:**0**,  
        **"nPhoneNumber"**:**1333333**,  
        **"nIDNumber"**:**511111**,  
        **"tszUserName"**:**"123123aa"**,  
        **"tszUserPass"**:**"123123"**,  
        **"tszEMailAddr"**:**"41adawd@1111.com"**  
    },  
    **"st\_Payload"**:{  
        **"nPayLen"**:**6**,  
        **"tszPayData"**:**"123456"**  
    }  
}

### 3.2.2 Reply

{  
    **"unOperatorType"**:**7**,  
    **"unOperatorCode"**:**7001**,  
    **"wReserve"**:**0**,

**"xhToken"**:**2123123123123**,

**"byVersion"**:**2**,  
    **"st\_MQProtocol"**:{  
        **"tszMQKey"**:**"主题名"**,  
        **"nSerial"**:**0**,  
        **"nKeepTime"**:**0**,  
        **"nGetTimer"**:**0**,

**"nPubTime"**:**0**  
    },  
    **"st\_Payload"**:{  
        **"nPayLen"**:**6**,  
        **"tszPayData"**:**"123456"**  
    }  
}

## 3.3 WEBSOCKET

WebSocket Protocol same load as http,return too,WebSocket request need ping and pong to heartbeat when is not have data change.

Websocket payload type only is TEXT.websocket can be used subscribe mode.

Websocket as long connection.It operates in the same way as TCP, except that the payload type is JSON.

# 四 Configure Description

## 4.1 Service Configure

Basic Configure File:XEngine\_Config.json

### 4.1.1 basic configure

* bDeamon: 1 deamon process run 0 is terminal run
* tszTopic:Default Topic Name
* nTCPPort:tcp port
* nHttpPort:http port
* nWSPort:websocket port

### 4.1.2 Max Configure

XMax Configure

* MaxClient Allow Max Client Count
* MaxQueue Allow Max Queue
* IOThread:network io process threads number
* nTCPThread:tcp process threads number
* nHttpThread:http process threads number
* nWSThread:websocket process thread number

### 4.1.3 Time Configure

XTime Configure

* nDBMonth:database save time,default month
* nSessionTime:check time

### 4.1.4 Log Configure

XLog Configure

* MaxSize:Log file size
* MaxCount:Log File Number
* LogLeave:Allow save level

### 4.1.5 Database Configure

XSql Configure,need to mysql

* SQLAddr:database address
* SQLPort:database port
* SQLUser:user
* SQLPass:pass

### 4.1.6 HTTP Pass Configure

XPass Configure

* nTimeout:timeout,second.
* tszPassRegister:Register address,like is:http://127.0.0.1:5600/api/v1/register
* tszPassUNReg:unregister address,like is:http://127.0.0.1:5600/api/v1/unregister
* tszPassLogin:login address,like is:http://127.0.0.1:5600/api/v1/login
* tszPassLogout:logout address,like is:http://127.0.0.1:5600/api/v1/logout

# 五 Advanced configuration

# 六 Other Functions

## 6.1 Reply Configure

suppoted set request reply and not reply protocol is:

* POST
* CREATE
* NOTIFY

## 6.2 HTTP Protocol

Supported http pass protocol is:

* Login
* Logout
* Register
* Unregister

# appendix

## Appendix 1 Type Define

reference file:XEngine\_CommHdr.h

## Appendix 2 Protocol Define

reference file:XEngine\_ProtocolHdr.h

## Appendix 3 Transformation Definition

reference file:XEngine\_Types.h only LINUX

## Appendix 4 update log