XEngine Message Service Docment

目录

[XEngine Message Service Docment 1](#_Toc198908716)

[Preface 4](#_Toc198908717)

[Reader 4](#_Toc198908718)

[Overview 4](#_Toc198908719)

[Associate Module 4](#_Toc198908720)

[一 Technical structure 4](#_Toc198908721)

[1.1 TCP 4](#_Toc198908722)

[1.2 HTTP 4](#_Toc198908723)

[1.3 WEBSOCKET 4](#_Toc198908724)

[1.4 MQTT 5](#_Toc198908725)

[1.5 HeartBeat 5](#_Toc198908726)

[1.6 Detailed Protocol 5](#_Toc198908727)

[1.7 Network Error Codes 5](#_Toc198908728)

[二 Configure Env 5](#_Toc198908729)

[2.1 WINDOWS 5](#_Toc198908730)

[2.1.1 Configure Environment 5](#_Toc198908731)

[2.1.2 complie and run 5](#_Toc198908732)

[2.2 LINUX 6](#_Toc198908733)

[2.3 MacOS 6](#_Toc198908734)

[三 Interface Protocol 6](#_Toc198908735)

[3.1 TCP 6](#_Toc198908736)

[3.1.1 Post Protocol 6](#_Toc198908737)

[3.1.2 Get Protocol 7](#_Toc198908738)

[3.1.3 Delete Protocol 8](#_Toc198908739)

[3.1.4 Create Topic 9](#_Toc198908740)

[3.1.5 Delete Topic 10](#_Toc198908741)

[3.1.6 Notification Message 11](#_Toc198908742)

[3.1.7 Bind Topic 11](#_Toc198908743)

[3.1.9 Serial Number Get 12](#_Toc198908744)

[3.1.10 User Authorize 13](#_Toc198908745)

[3.1.11 Message Modify 14](#_Toc198908746)

[3.1.12 Topic Modify 15](#_Toc198908747)

[3.1.13 UNRead Message 15](#_Toc198908748)

[3.1.14 Delete Message 16](#_Toc198908749)

[3.2 WEBSOCKET 17](#_Toc198908750)

[3.2.1 Request 18](#_Toc198908751)

[3.2.2 Reply 18](#_Toc198908752)

[3.3 MQTT 19](#_Toc198908753)

[3.4 HTTP 20](#_Toc198908754)

[3.2.1 Query User List 20](#_Toc198908755)

[3.2.2 Query Topic List 21](#_Toc198908756)

[3.2.3 Online List 22](#_Toc198908757)

[3.2.4 Delete Topic 22](#_Toc198908758)

[3.2.5 Delete User 23](#_Toc198908759)

[3.2.6 Create Topic 23](#_Toc198908760)

[3.2.7 Get Topic 24](#_Toc198908761)

[四 Configure Description 24](#_Toc198908762)

[4.1 Service Configure 24](#_Toc198908763)

[4.1.1 basic configure 25](#_Toc198908764)

[4.1.2 Max Configure 25](#_Toc198908765)

[4.1.3 Log Configure 25](#_Toc198908766)

[4.1.4 Database Configure 25](#_Toc198908767)

[4.1.5 HTTP Pass Configure 25](#_Toc198908768)

[4.1.6 Memory Cach Configure 26](#_Toc198908769)

[4.1.7 Report Configure 26](#_Toc198908770)

[4.1.8 Verification Configure 26](#_Toc198908771)

[4.2 Database Configure 26](#_Toc198908772)

[4.2.1 User Database 26](#_Toc198908773)

[4.2.2 MQ Database 26](#_Toc198908774)

[五 Advanced configuration 27](#_Toc198908775)

[5.1 Message Protocol 27](#_Toc198908776)

[5.1.1 Message Attritube 27](#_Toc198908777)

[六 Other Functions 27](#_Toc198908778)

[6.1 HTTP Protocol 27](#_Toc198908779)

[appendix 28](#_Toc198908780)

[Appendix 1 Type Define 28](#_Toc198908781)

[Appendix 2 Protocol Define 28](#_Toc198908782)

[Appendix 3 Transformation Definition 28](#_Toc198908783)

[Appendix 4 update log 28](#_Toc198908784)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Message Service Docment | |
| Be A Version： | V3.16.0.1001 | |
| Released： | 2025-05-24 | |
| 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

Provide HTTP interface for server information query and management

## WEBSOCKET

Websocket type can be realized by using TEXT. Load JSON, WEBSOCKET message supports subscription mode

## 1.4 MQTT

The MQTT protocol is implemented based on TCP, using the MQTT5.0 version protocol. Load JSON data, the JSON data is the same as WEBSOCKET, because it needs to be converted to our own communication standard, and the reply is also

## 1.5 HeartBeat

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

## 1.6 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

## 1.7 Network Error Codes

Network error codes are now available in a common protocol header file. The returned server errors will be defined here, and information about the errors can be found here.

# 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

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

Supports most release version systems, you can view them on the github publishing page, or view the github action script

## 2.3 MacOS

MacOS requires 13 or above versions, and the compilation and operation mode can refer to linux

# 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.

wReserve:if the message is time release,the feild of protocol header mean is whether it recv

#### 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\_REQTOPICCREATE

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\_REPTOPICCREATE

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\_REQTOPICDELETE

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\_REPTOPICDELETE

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 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

### 3.1.7 Bind Topic

Bind Support recv notify message and get message without serial,Support jump to the specified serial number to start

#### 3.1.7.1 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\_REQTOPICBIND

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

#### 3.1.7.2 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\_REPTOPICBIND

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 Message Modify

#### 3.1.11.1 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\_REQMODIFY

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

#### 3.1.11.2 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\_REPMODIFY

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.1.12 Topic Modify

#### 3.1.12.1 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

#### 3.1.12.2 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

### 3.1.13 UNRead Message

#### 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\_REQUNREAD

unPacketSize = 0

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:null

#### 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\_REPUNREAD

unPacketSize = JSON

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

{

"Array" :

[

{

"Count" : 11,

"Name" : "XEngine\_NotifyKey"

}

],

"Count" : 1

}

### 3.1.14 Delete Message

#### 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\_REQDELETE

unPacketSize = sizeof(XENGINE\_PROTOCOL\_XMQ)

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 and serial numbe

XENGINE\_PROTOCOL\_XMQ

#### 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\_REPDELETE

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

## 3.2 WEBSOCKET

WebSocket Protocol payload is json,return too,WebSocket request need heartbeat to keepalive 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

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,byVersion indicates the payload type, refer to the definition of the protocol header, if it is binary, it needs to be base64 encoded before transmission

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,

"byIsReply":1,

"xhToken":2123123123123,

"st\_MQProtocol":{

"tszMQKey":"主题名",

"tszMQUsr":"所属用户,可为NULL",

"nSerial":0,

"nKeepTime":0,

"nPubTime":0,

"nMSGAttr":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,

"byVersion":2,

"st\_MQProtocol":{

"tszMQKey":"Topic Name",

"tszMQUsr":"belong user ,can be NULL",

"nSerial":0,

"nKeepTime":0,

"nPubTime":0,

"nMSGAttr":0

},

"st\_Payload":{

"nPayType":5,

"nPayLen":6,

"tszPayData":"123456"

}

}

Example:publish message

{

"unOperatorType": 7,

"unOperatorCode": 28672,

"wReserve": 0,

"byVersion": 2,

"byIsReply": 0,

"xhToken": 0,

"st\_MQProtocol": {

"tszMQKey": "XEngine\_CommKey",

"tszMQUsr": null,

"nSerial": 0,

"nKeepTime": 0,

"nPubTime": -1,

"nMSGAttr": 0

},

"st\_Payload": {

"nPayLen": 6,

"tszPayData": "123456"

}

}

## 3.3 MQTT

The MQTT protocol now supports user login and user message publishing and bind and un bind. Other protocols will be supported in subsequent versions

MQTT Not support user register,user register thought tcp protocol.

MQTT uses version 5.0 protocol for communication. Users and passwords must be populated.

MQTT does not support SSL at this time.

MQTT's message publish protocol requires JSON to be populated according to the WEBSOCKET protocol.

MQTT publish message Support QOS1 and qos0

## 3.4 HTTP

HTTP is used as a management interface and can be used to manage the server via the HTTP protocol.

HTTP comes with built-in TOKEN authentication. The requesting user must have level 0 administrator privileges, which need to be manually set in the database by the user. The default level is 20, which corresponds to a regular user. The request method is as follows:

Method: GET

URL: http://127.0.0.1:5202/api?function=login&user=123123aa&pass=123123

The returned value will include a msg field containing the token, which is a numeric value. This token must be saved, and all subsequent requests must include it.

All requests should include the token. The following request format assumes the token is included:

{

“xhToken”:123456789

}

### 3.2.1 Query User List

#### 3.2.1.1 Request

Method:POST

Address:<http://127.0.0.1:5202/api?function=getuser>

Payload:

{

"xhToken": 123456789

}

#### 3.2.1.2 Reply

{

"Array":[

{

"nIDNumber":511000000000101000,

"nPhoneNumber":13699999999,

"nUserLevel":0,

"nUserState":0,

"tszCreateTime":"2023-08-11 13:36:54",

"tszEMailAddr":"486179@qq.com",

"tszLoginTime":"2023-08-11 13:37:22",

"tszUserName":"123123aa",

"tszUserPass":"123123"

}

],

"Count":1,

"code":0

}

### 3.2.2 Query Topic List

#### 3.2.2.1 Request

Method:PSOT

Address:http://127.0.0.1:5202/api?function=getlist

Payload:

{

"xhToken": 123456789

}

#### 3.2.2.2 Reply

{

"Array":[

{

"tszTopicName":"XEngine\_CommKey"

},

{

"tszTopicName":"XEngine\_NotifyKey"

}

],

"Count":2,

"code":0

}

### 3.2.3 Online List

#### 3.2.3.1 Request

Method:POST

API:http://127.0.0.1:5202/api?function=online

Payload:

{

"xhToken": 123456789

}

#### 3.2.3.2 Reply

{

"Array":[

{

"tszUserAddr":"123123aa"

}

],

"Count":1,

"code":0

}

### 3.2.4 Delete Topic

#### 3.2.4.1 Request

Method:POST

API:http://127.0.0.1:5202/api?function=deletetopic

Payload:

{

"xhToken": 123456789,

"Object": {

"name": "operator name"

}

}

#### 3.2.4.2 Reply

{

"msg": "success",

"code": 0

}

### 3.2.5 Delete User

#### 3.2.5.1 Request

Method:POST

API:http://127.0.0.1:5202/api?function=deleteuser

Payload:

{

"xhToken": 123456789,

"Object": {

"name": "operator name"

}

}

#### 3.2.5.2 Reply

{

"msg": "success",

"code": 0

}

### 3.2.6 Create Topic

#### 3.2.6.1 Request

Method:POST

API:http://127.0.0.1:5202/api?function=createtopic

Payload:

{

"xhToken": 123456789,

"Object": {

"name": "operator name"

}

}

#### 3.2.6.2 Reply

{

"msg": "success",

"code": 0

}

### 3.2.7 Get Topic

#### 3.2.7.1 Request

Method:POST

API:http://127.0.0.1:5202/api?function=gettopic

Payload:

{

"xhToken": 123456789,

"Object": {

"name": "operator name"

}

}

#### 3.2.7.2 Reply

{

"msg": "success",

"lpszTopicName": “operator name”,

“Count”:1

}

# 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
* nMQTTPort:mqtt 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
* nMQTTThread:mqtt process thread number

### 4.1.3 Log Configure

XLog Configure

* MaxSize:Log file size
* MaxCount:Log File Number
* LogLeave:Allow save level
* LogType:lop type
* tszLOGFile:log file path

### 4.1.4 Database Configure

XSql Configure,need to mysql

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

### 4.1.5 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

### 4.1.6 Memory Cach Configure

XMemory Configure

* bDataQueryEnable:whether to enable message memory query
* bDataInsertEnable:whether to enable message memory insert
* bUserQueryEnable:whether to enable user query memory support
* nTimeLast:message timeout at last time
* nTimeCount:message total time

### 4.1.7 Report Configure

XReport Configure

* bEnable:whether to enable
* tszServiceName:serivce name
* tszAPIUrl:API address

### 4.1.8 Verification Configure

XAuthorize Configure

* bHTTPAuth:whether to enable http verification

## 4.2 Database Configure

Configure File:XEngine\_DBConfig.json

### 4.2.1 User Database

#### 4.2.1.1 User Time Table

* bPubClear:clear database when time release completed

### 4.2.2 MQ Database

MQData Configure

* nDBMonth:DB Save Time,default is month
* bCommSub:default bind comm topic when user register

# 五 Advanced configuration

## 5.1 Message Protocol

typedef struct

{

*XCHAR* tszMQKey[*MAX\_PATH*];

*XCHAR* tszMQUsr[*MAX\_PATH*];

\_\_int64x nSerial;

\_\_int64x nPubTime;

int nKeepTime;

XENGINE\_PROTOCOL\_MSGATTR st\_MSGAttr;

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

The meaning can be found in the related header file. Note the description and meaning of nPubTime.

### 5.1.1 Message Attritube

Message Attritube supported basic funtion now.

Simply set its flag bit to 1 when configuring the message to enable the

The following two message properties are currently supported

* byAttrAll:Notify All client.the message typoe nPubTime Attr is 0,tszMQUsr is Ignore
* byAttrSelf:Notify include yourself.

# 六 Other Functions

## 6.1 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