XEngine Authorize Service Docment

目录

[XEngine Authorize Service Docment 1](#_Toc21784)

[Preface 5](#_Toc19526)

[Reader 5](#_Toc19917)

[Overview 5](#_Toc9522)

[Associate Module 5](#_Toc6026)

[一 Technical structure 5](#_Toc29083)

[1.1 Directory Structure 5](#_Toc16390)

[1.2 Program Directory 5](#_Toc12059)

[1.3 User Leave 6](#_Toc16975)

[1.4 Detailed Protocol 6](#_Toc22800)

[二 Configure Env 6](#_Toc3746)

[2.1 WINDOWS 6](#_Toc32667)

[2.1.1 Configure Environment 6](#_Toc14869)

[2.1.2 complie and run 6](#_Toc23332)

[2.2 LINUX 6](#_Toc23666)

[2.2.1 Evnironment Configure 6](#_Toc26953)

[2.2.2 Complie and Run 7](#_Toc5614)

[2.3 MacOS 7](#_Toc17975)

[2.4 Version Requirements 7](#_Toc17317)

[2.4.1 System Version 7](#_Toc19885)

[2.4.2 Software Version 7](#_Toc24843)

[三 TCP Interface Protocol 7](#_Toc10561)

[3.1 Delete Protocol 8](#_Toc17828)

[3.1.1 Request 8](#_Toc23962)

[3.1.2 Reply 8](#_Toc10694)

[3.2 Register Protocol 9](#_Toc14019)

[3.2.1 Request 9](#_Toc29242)

[3.2.2 Reply 9](#_Toc13246)

[3.3 Login Protocol 10](#_Toc486)

[3.3.1 Request 10](#_Toc21502)

[3.3.2 Reply 10](#_Toc2552)

[3.4 Pay Protocol 11](#_Toc32120)

[3.4.1 Request 11](#_Toc21535)

[3.4.2 Reply 12](#_Toc10340)

[3.5 Get Password 12](#_Toc8152)

[3.5.1 Request 12](#_Toc9705)

[3.5.2 Reply 13](#_Toc13017)

[3.6 Get Time 13](#_Toc16716)

[3.6.1 Request 13](#_Toc32288)

[3.6.2 Reply 14](#_Toc14158)

[3.7 Network Verification 14](#_Toc1075)

[3.7.1 Request 14](#_Toc7657)

[3.7.2 Reply 15](#_Toc21129)

[3.10 Notification Protocol 15](#_Toc26253)

[3.11 Notification Timeout 16](#_Toc12102)

[四 Websocket Interface Protocol 16](#_Toc14205)

[4.1 user info protocol 16](#_Toc26017)

[4.1.1 Request 16](#_Toc32013)

[4.1.2 Reply 17](#_Toc11950)

[4.2 user authorize 18](#_Toc22986)

[4.2.1 Request 18](#_Toc24942)

[4.2.2 Reply 18](#_Toc5323)

[4.3 user pay protocol 19](#_Toc11335)

[4.3.1 Request 19](#_Toc24521)

[4.3.2 Reply 19](#_Toc27476)

[4.4 user time protocol 19](#_Toc23211)

[4.4.1 Request 19](#_Toc2612)

[4.4.2 Reply 19](#_Toc10399)

[4.5 fast authorize 20](#_Toc5432)

[4.5.1 Request 20](#_Toc19212)

[4.5.2 Reply 20](#_Toc32038)

[4.6 note protocol 20](#_Toc22733)

[4.6.1 Request 21](#_Toc27110)

[4.6.2 Reply 21](#_Toc7326)

[4.7 timeout protocol 21](#_Toc7130)

[4.7.1 Request 21](#_Toc15271)

[4.7.2 Reply 21](#_Toc6320)

[五 HTTP Management Interface 21](#_Toc19832)

[5.1 User Management Interface 22](#_Toc2020)

[5.1.1 Get User 22](#_Toc22791)

[5.1.2 User List 23](#_Toc1112)

[5.1.3 Close Client 23](#_Toc1637)

[5.1.4 Modify User 24](#_Toc5036)

[5.1.5 Delete User 25](#_Toc4930)

[5.2 Serial Management API 25](#_Toc14641)

[5.2.1 List Serial 25](#_Toc18559)

[5.2.2 Insert Serial 26](#_Toc2486)

[5.2.3 Delete Serial 27](#_Toc24923)

[六 Configure Description 27](#_Toc3712)

[6.1 Basic Configure 27](#_Toc6400)

[6.2 MAX Configure 27](#_Toc31014)

[6.3 Verication Configure 28](#_Toc31095)

[6.4 Encrypto Configure 28](#_Toc30504)

[6.5 Database Configure 28](#_Toc8020)

[6.6 Log Configure 28](#_Toc11139)

[appendix 29](#_Toc32657)

[Appendix 1 Type Define 29](#_Toc17799)

[Appendix 2 Protocol Define 29](#_Toc26995)

[Appendix 3 Transformation Definition 29](#_Toc10171)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Authorize Service Docment | |
| Be A Version： | V3.0.0.1001 | |
| Released： | 2022-08-16 | |
| Writer： qyt | | |

# Preface

## Reader

Developer ,tester

## **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

# 一 Technical structure

Used to tcp protocol though basic protoco

Should be bind three port when Start service

## Directory Structure

* XEngine\_Apps:example code path
* XEngine\_Docment:docment path
* XEngine\_Release:complie release path
* XEngine\_Source:code path

## Program Directory

We provide server and management tools

* XEngine\_AuthorizeApp:WINDOWS Management Tools program.depend XEngine\_AuthorizeService
* XEngine\_AuthorizeService:Network Authorize Mainly Server

## User Leave

Users have a total of 0-5 levels, -1 is banned.

0 is the highest level, 5 is the lowest. The default registration level is 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 execute the script to configure the environment.

### 2.1.2 complie and run

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

If environment not have error.complie is succesed.contrain 5 module and 2 exe program

And you need copy 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\_AuthorizeService 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 TCP Interface Protocol

TCP interface protocol is used for application verification such as APP. It can be used for login, registration, logout, deletion, etc.

Please note: the verification protocol use to reserved field wReserve of the protocol header to reply the client request, telling the client operation is succeeded or failed, 0 means success, other means failure.

wCrypto means encryption and decryption, currently only supports type 4, XCrypto algorithm. If encryption is required, please fill in 4, otherwise please fill in 0

## 3.1 Delete Protocol

Use to delete user.

### 3.1.1 Request

Request a delete user,you need to use the header protocol + register protocol to achieve.

The content of this agreement is used to judge and verify whether the user has the right to delete himself. Of course, with the use of TOKEN, you can also realize the distribution of permissions. Use TOKEN to verify whether this user has the right to delete another user

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOL\_USERINFO)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct tag\_XEngine\_Protocol\_UserReg

{

*TCHAR* tszUserName[64];

*TCHAR* tszUserPass[64];

*TCHAR* tszEMailAddr[64];

*TCHAR* tszLoginTime[64];

*TCHAR* tszCreateTime[64];

\_\_int64x nPhoneNumber;

\_\_int64x nIDNumber;

int nUserLevel;

*int nUserState*;

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

### 3.1.2 Reply

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

## 3.2 Register Protocol

### 3.2.1 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\_REQREGISTER

unPacketSize = sizeof(XENGINE\_PROTOCOL\_USERINFO)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct tag\_XEngine\_Protocol\_UserReg

{

*TCHAR* tszUserName[64];

*TCHAR* tszUserPass[64];

*TCHAR* tszEMailAddr[64];

*TCHAR* tszLoginTime[64];

*TCHAR* tszCreateTime[64];

\_\_int64x nPhoneNumber;

\_\_int64x nIDNumber;

int nUserLevel;

*int nUserState*;

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

### 3.2.2 Reply

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

## 3.3 Login Protocol

The login protocol is the first step that the client needs to do when requesting the server to exchange data

### 3.3.1 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];

*CHAR* tszUserPass[64];

ENUM\_PROTOCOLCLIENT\_TYPE enClientType;

ENUM\_PROTOCOLDEVICE\_TYPE enDeviceType;

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

### 3.3.2 Reply

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 = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 3.3.2.1 Reply Type

The Server reply success or failure can be custom by developer,we are make sure through wReserver,the value on the login protocol,return value means:

* 0:success
* 251:user not found
* 252:password is error
* 253:User does not have permission
* 254:user timeout
* 255:Server internal error

## 3.4 Pay Protocol

The user recharge agreement is recharged through a recharge card, and the recharge card needs to be generated through our serial number module

### 3.4.1 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\_REQPAY

unPacketSize = sizeof(AUTHREG\_PROTOCOL\_USERPAY)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct

{

*CHAR* tszUserName[XENGINE\_AUTHREG\_SERVICE\_SQL\_MAX\_USERNAME];

*CHAR* tszSerialNumber[128];

}AUTHREG\_PROTOCOL\_USERPAY, \*LPAUTHREG\_PROTOCOL\_USERPAY;

### 3.4.2 Reply

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

## 3.5 Get Password

### 3.5.1 Request

Protocol Header:User And IDNumber and phone number can be empty

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOL\_USERINFO)

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct tag\_XEngine\_Protocol\_UserReg

{

*TCHAR* tszUserName[64];

*TCHAR* tszUserPass[64];

*TCHAR* tszEMailAddr[64];

*TCHAR* tszLoginTime[64];

*TCHAR* tszCreateTime[64];

\_\_int64x nPhoneNumber;

\_\_int64x nIDNumber;

int nUserLevel;

*int nUserState*;

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

### 3.5.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = sizeof(XENGINE\_PROTOCOL\_USERAUTH)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct tag\_XEngine\_Protocol\_Auth

{

*CHAR* tszUserName[64];

*CHAR* tszUserPass[64];

ENUM\_PROTOCOLCLIENT\_TYPE enClientType;

ENUM\_PROTOCOLDEVICE\_TYPE enDeviceType;

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

#### 2.5.2.1 Error Code

* 291:User not exist
* 292:ver infomation is not success

## 3.6 Get Time

Get Time mean is get user left time,login is requested

### 3.6.1 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\_REQGETTIME

unPacketSize = 0

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

### 3.6.2 Reply

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = sizeof(AUTHREG\_PROTOCOL\_TIME)

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:

typedef struct

{

CHAR tszUserAddr[XENGINE\_AUTHREG\_SERVICE\_SQL\_MAX\_USERNAME];

CHAR tszUserName[XENGINE\_AUTHREG\_SERVICE\_SQL\_MAX\_USERNAME];

CHAR tszLeftTime[64];

\_\_int64x nTimeLeft;

\_\_int64x nTimeONLine;

ENUM\_AUTHREG\_GENERATESERIALTYPE enSerialType;

}AUTHREG\_PROTOCOL\_TIME, \* LPAUTHREG\_PROTOCOL\_TIME;

## 3.7 Network Verification

Temporary Network Authentication Protocol,Can be used without registering and logging in,Register through a unique identification code (CPUID, motherboard ID, hard disk ID).This registration will be sent to the server, the server will automatically save the record, and identify whether it has expired, and it is convenient for users to use.

This protocol is more secure than local authentication. It can be convenient for some users who don't want to register..

### 3.7.1 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\_REQTRYVER

unPacketSize = IDLEN

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:填写唯一标识码,可以是CPUID等

### 3.7.2 Reply

Protocol Header:Success will return the following content, indicating that the time has not expired

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = 0

byVersion = 1

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

#### 3.7.2.1 Error Code

* 0x2D1:Try to timeout
* 0x2D2:Server not enable
* 0x2D3:Server internal occur

## 3.10 Notification Protocol

This protocol is only used to send text content. It is used to notify users of messages.

Protocol Header:

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = MSGLEN

byVersion = 1

byIsReply = TRUE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

Protocol Body:The body of the agreement is the content to be sent.

## 3.11 Notification Timeout

This protocol is a notification protocol, and the server actively issues it. When the client receives this protocol, it needs to actively disconnect, otherwise the server will continue to issue this notification.

This agreement indicates that the client's time has expired and there is no time left.

wHeader = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_HEADER

xhToken = 0

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

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

unPacketSize = 0

byVersion = 0

byIsReply = FALSE

wReserve = 0

wPacketSerial = 0

wTail = XENGIEN\_COMMUNICATION\_PACKET\_PROTOCOL\_TAIL

# 四 Websocket Interface Protocol

WEBSOCKET Protocol referred to as ws protocol.

In addition to the above TCP protocol, the WS protocol can support HTML or web processing authorization verification, so in addition to APP programs, we can also support authorization verification for websites

The authorization verification of the web is realized by WEBSOCKET protocol load JSON.

The unOperatorType in JSON refers to the unOperatorType of TCP, and the unOperatorCode refers to the unOperatorCode of TCP, which is filled in decimal

## 4.1 user info protocol

Use to delete and register and get pass and user set

### 4.1.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8193**,  
    **"byIsReply"**:**1**,

**"wCrypto"**:**0**,  
    **"st\_UserInfo"**:{  
        **"tszUserName"**:**"123123aa"**,  
        **"tszUserPass"**:**"123123"**,  
        **"tszEMailAddr"**:**"486179@qq.com"**,  
        **"nPhoneNumber"**:**1369999999**,  
        **"nIDNumber"**:**5000000000**  
    }  
}

### 4.1.2 Reply

#### 4.1.2.1 user delete,register set

{  
    **"wHeader"**:**17**,  
    **"wTail"**:**255**,  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8197**,  
    **"byIsReply"**:**0**,  
    **"wCrypto"**:**0**,  
    **"wReserve"**:**0**  
}

#### 4.1.2.2 get password

{  
    **"wHeader"**:**17**,  
    **"wTail"**:**255**,  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8197**,  
    **"byIsReply"**:**0**,  
    **"wCrypto"**:**0**,  
    **"wReserve"**:**0**,  
    **"st\_UserAuth"**:{  
        **"tszUserName"**:**"123123aa"**,  
        **"tszUserPass"**:**"123123"**,  
        **"enClientType"**:**10**,  
        **"enDeviceType"**:**25**  
    }  
}

## 4.2 user authorize

Use to login,get user protocol

### 4.2.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8208**,  
    **"byIsReply"**:**1**,

**"wCrypto"**:**0**,  
    **"st\_UserAuth"**:{  
        **"tszUserName"**:**"123123aa"**,  
        **"tszUserPass"**:**"123123"**,  
        **"enClientType"**:**10**,  
        **"enDeviceType"**:**25**  
    }  
}

### 4.2.2 Reply

#### 4.2.2.1 Login

Reference 4.1.2.1

#### 4.2.2.2 get user

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8203**,  
    **"byIsReply"**:**1**,  
    **"wCrypto"**:**0**,  
    **"st\_UserInfo"**:{  
        **"tszUserName"**:**"user"**,  
        **"tszUserPass"**:**"pass"**,  
        **"tszCreateTime"**:**"register time"**,  
        **"tszEMailAddr"**:**"email"**,  
        **"tszLoginTime"**:**"last login time"**,  
        **"nIDNumber"**:**5151515151**,  
        **"nPhoneNumber"**:**13131313**,  
        **"nUserLevel"**:**1**,  
        **"nUserState"**:**0**  
    }  
}

## 4.3 user pay protocol

### 4.3.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8199**,  
    **"byIsReply"**:**1**,

**"wCrypto"**:**0**,  
    **"st\_UserPay"**:{  
        **"tszUserName"**:**"123123aa"**,  
        **"tszSerialNumber"**:**"123123"**  
    }  
}

### 4.3.2 Reply

Reference 4.1.2.1

## 4.4 user time protocol

Use to get user time

### 4.4.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8203**,  
    **"byIsReply"**:**1,**

**"wCrypto"**:**0**  
}

### 4.4.2 Reply

{  
    **"wHeader"**:**17**,  
    **"wTail"**:**255**,  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8204**,  
    **"byIsReply"**:**0**,  
    **"wCrypto"**:**0**,  
    **"wReserve"**:**0**,  
    **"st\_UserTime"**:{  
        **"tszUserName"**:**"123123aa"**,  
        **"tszUserAddr"**:**"123123"**,  
        **"tszLeftTime"**:**"2021-12-22 10:11:23"**,  
        **"nTimeLeft"**:**25**,  
        **"nTimeONLine"**:**10**,  
        **"enDeviceType"**:**1**,  
        **"enSerialType"**:**1**  
    }  
}

## 4.5 fast authorize

Reference 3.6

### 4.5.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8203**,  
    **"byIsReply"**:**1**,  
    **"wCrypto"**:**0**,  
    **"lpszPayload"**:**"hard code or other"**  
}

### 4.5.2 Reply

Reference 4.1.2.1

## 4.6 note protocol

Reference 3.10

### 4.6.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**12272**,  
    **"byIsReply"**:**1**,  
    **"wCrypto"**:**0**,  
    **"lpszPayload"**:**"send message"**  
}

### 4.6.2 Reply

无

## 4.7 timeout protocol

Reference 3.11

### 4.7.1 Request

{  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**12287**,  
    **"byIsReply"**:**0**,  
    **"wCrypto"**:**0**  
}

### 4.7.2 Reply

Null

# 五 HTTP Management Interface

This interface is used for HTTP RESTFUL, and the server can be managed through this interface in the future

The HTTP interface will only return 200 success, the error code needs to be obtained through the json of the payload. code and msg

## 5.1 User Management Interface

### 5.1.1 Get User

#### 5.1.1.1 Request

Method:POST

API:http://IP:PORT/auth/client/get

Payload:

{

"st\_UserInfo":{

"tszUserName":"用户名"

}

}

#### 5.1.1.2 Reply

{

"code":0,

"msg":"success",

"st\_UserTable":{

"tszLeftTime":"还剩下多少时间",

"tszHardCode":"硬件吗",

"enSerialType":1,

"enDeviceType":0,

"st\_UserInfo":{

"tszUserName":"用户名",

"tszUserPass":"密码",

"tszEMailAddr":"电子邮件",

"tszLoginTime":"登录时间",

"tszCreateTime":"注册时间",

"nPhoneNumber":1369999999,

"nIDNumber":5111010101011,

"nUserLevel":5,

"nUserState":0

}

}

}

### 5.1.2 User List

#### 5.1.2.1 Request

Method:POST

API:http://IP:PORT/auth/client/list

Payload:

#### 5.1.2.2 Reply

{

"Array":[

{

"enDeviceType":0,

"enSerialType":1,

"st\_UserInfo":{

"nIDNumber":511025111111111100,

"nPhoneNumber":1366666666,

"nUserLevel":5,

"nUserState":0,

"tszCreateTime":"2022-05-27 10:09:17",

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

"tszLoginTime":"",

"tszUserName":"123123aa",

"tszUserPass":"123123"

},

"tszHardCode":"ND2N09DN1N11",

"tszLeftTime":"81"

}

],

"code":0,

"msg":"success"

}

### 5.1.3 Close Client

#### 5.1.3.1 Request

Method:POST

API:http://IP:PORT/auth/client/get

Payload:

{

"st\_UserInfo":{

"tszUserName":"用户名"

}

}

#### 5.1.3.2 Reply

{

"code":0,

"msg":"success"

}

### 5.1.4 Modify User

#### 5.1.4.1 Request

Method:POST

API:http://IP:PORT/auth/client/modify

Payload:

{

"st\_UserTable":{

"enSerialType":1,

"st\_UserInfo":{

"nIDNumber":511025111111111100,

"nPhoneNumber":1366666666,

"nUserLevel":5,

"tszCreateTime":"2022-05-27 10:09:17",

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

"tszUserName":"123123aa",

"tszUserPass":"123123"

},

"tszHardCode":"ND2N09DN1N11",

"tszLeftTime":"81"

}

}

#### 5.1.4.2 Reply

{

"code":0,

"msg":"success"

}

### 5.1.5 Delete User

#### 5.1.5.1 Request

Method:POST

API:http://IP:PORT/auth/client/modify

Payload:

{

"st\_UserInfo":{

"tszUserName":"用户名"

}

}

#### 5.1.5.2 Reply

{

"code":0,

"msg":"success"

}

## 5.2 Serial Management API

### 5.2.1 List Serial

#### 5.2.1.1 Request

Method:POST

API:http://IP:PORT/auth/serial/list

Payload:

#### 5.2.1.2 Reply

{

"Array":[

{

"bIsUsed":0,

"enSerialType":2,

"tszCreateTime":"2022-08-11 14:45:58",

"tszMaxTime":"2",

"tszSerialNumber":"XAUTH-XYRYD-MRLJ2-YYF85-UQLKI-31115-98XVA-LQYKX-12744-KFQ38",

"tszUserName":"NOT"

}

],

"code":0,

"msg":"success"

}

### 5.2.2 Insert Serial

#### 5.2.2.1 Request

Method:POST

API:http://IP:PORT/auth/serial/list

Payload:

{

"st\_SerialInfo":{

"enSerialType":1,

"nNumberCount":5,

"nSerialCount":2,

"tszHasTime":"20"

}

}

enSerialType:代表序列号类型

nNumberCount:序列号字段个数

nSerialCount:序列号个数

tszHasTime:拥有的时间

#### 5.2.2.2 Reply

{

"code":0,

"msg":"success"

}

### 5.2.3 Delete Serial

#### 5.2.3.1 Request

Method:POST

API:http://IP:PORT/auth/serial/delete

Payload:

{

"Array":[

{

"tszSerialNumber":"XAUTH-XYRYD-MRLJ2-YYF85-UQLKI-31115-98XVA-LQYKX-12744-KFQ38"

}

]

}

#### 5.2.3.2 Reply

{

"code":0,

"msg":"success"

}

# 六 Configure Description

Configure File:XEngine\_Config.json

## 6.1 Basic Configure

* nTCPPort:TCP listen port
* nWSPort:websocket port
* nHTTPPort:http port
* bDeamon:whether deamon run
* nTimeNotify:timeout notify set,1 means always notify until the client actively closes

## 6.2 MAX Configure

Configure Information:XMax

* nMaxClient:Max Client Count
* nMaxQueue:Max Queue Count
* nIOThread:network io count
* nTCPThread:TCP threads count
* nWSThread:WEBSOCKET threads count
* nHTTPThread:HTTP threads count

## 6.3 Verication Configure

Configure Information:Verification,Quick verification does not require login, and you can get verification authorization information by directly requesting after connecting.

* nUserTimeout:user login timeout
* nVerTime:Quick Verification Time
* nVerMode:Quick Verification Mode
* nTryMode:Try Mode
* nTryTime:Try Time

## 6.4 Encrypto Configure

Configure Information:Crypto,It is recommended to use encrypted transmission, otherwise it is easy to be cracked

* bEnable:whether to enable
* nPass:password,just number

## 6.5 Database Configure

Configure Information:XSql

* tszSQLite:Database file address

## 6.6 Log Configure

Configure Information:XLog just only use to version without interface

* nMaxSize:max log file size
* nMaxCount:file back number
* nLogLeave:log level
* tszLogFile:log save address

# 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