XEngine Authorize Service Docment

目录

[XEngine Authorize Service Docment 1](#_Toc15145)

[Preface 4](#_Toc3437)

[Reader 4](#_Toc25637)

[Overview 4](#_Toc14624)

[Associate Module 4](#_Toc10680)

[一 Technical structure 4](#_Toc22658)

[1.1 Directory Structure 4](#_Toc7390)

[1.2 Program Directory 4](#_Toc9804)

[1.3 User Leave 4](#_Toc27761)

[1.4 Detailed Protocol 5](#_Toc230)

[二 Configure Env 5](#_Toc9284)

[2.1 WINDOWS 5](#_Toc6054)

[2.1.1 Configure Environment 5](#_Toc26110)

[2.1.2 complie and run 5](#_Toc32431)

[2.2 LINUX 5](#_Toc29157)

[2.2.1 Evnironment Configure 5](#_Toc18195)

[2.2.2 Complie and Run 5](#_Toc18178)

[2.3 MacOS 6](#_Toc25849)

[2.4 Version Requirements 6](#_Toc7480)

[2.4.1 System Version 6](#_Toc15297)

[2.4.2 Software Version 6](#_Toc20807)

[三 TCP Interface Protocol 6](#_Toc10369)

[3.1 Login Protocol 7](#_Toc23006)

[3.1.1 Request 7](#_Toc25947)

[3.1.2 Reply 7](#_Toc22041)

[3.2 Notification Timeout 8](#_Toc13424)

[四 Websocket Interface Protocol 8](#_Toc2343)

[4.1 user authorize 9](#_Toc30591)

[4.2.1 Request 9](#_Toc29677)

[4.2.2 Reply 9](#_Toc16756)

[4.2 timeout protocol 9](#_Toc1240)

[五 HTTP Management Interface 9](#_Toc6513)

[5.1 Client Management Interface 10](#_Toc22343)

[5.1.1 Get User 10](#_Toc8981)

[5.1.2 User List 11](#_Toc1836)

[5.1.3 Close Client 12](#_Toc15889)

[5.1.4 Modify User 12](#_Toc730)

[5.2 Serial Management API 13](#_Toc32583)

[5.2.1 List Serial 13](#_Toc21528)

[5.2.2 Insert Serial 14](#_Toc14346)

[5.2.3 Delete Serial 15](#_Toc25328)

[5.3 User Interface 15](#_Toc9275)

[5.3.1 User Delete 15](#_Toc2003)

[5.3.2 User Register 16](#_Toc24141)

[5.3.3 User Pay 17](#_Toc12300)

[5.3.4 Get Back Password 17](#_Toc28795)

[5.3.5 Get Left Time 18](#_Toc11926)

[5.3.6 Fast Verification 19](#_Toc21694)

[5.4 Token Protocol 19](#_Toc25806)

[5.4.1 Login 19](#_Toc310)

[5.4.2 UPDate 20](#_Toc2465)

[5.4.3 Close 21](#_Toc19756)

[5.5 Function Switch 21](#_Toc19224)

[5.5.1 Get Functions List 21](#_Toc6872)

[5.5.2 Set Functions Switch 22](#_Toc9598)

[六 Third-Verification 22](#_Toc21639)

[6.1 User Login 23](#_Toc25711)

[6.1.1 Request 23](#_Toc32734)

[6.1.2 Response 23](#_Toc9609)

[6.2 User Logout 23](#_Toc77)

[6.2.1 Request 23](#_Toc12708)

[6.2.2 Response 23](#_Toc10729)

[6.3 User Timeout 23](#_Toc26800)

[6.3.1 Request 23](#_Toc4166)

[6.3.2 Response 23](#_Toc7603)

[七 Advanced Features 24](#_Toc6417)

[7.1 Distributed Authorize 24](#_Toc27512)

[八 Configure Description 24](#_Toc22641)

[8.1 Basic Configure File 24](#_Toc4075)

[8.1.1 Basic Configure 24](#_Toc11643)

[8.1.2 MAX Configure 24](#_Toc2880)

[8.1.3 Verication Configure 24](#_Toc1854)

[8.1.4 Login Configure 25](#_Toc26959)

[8.1.5 Encrypto Configure 25](#_Toc16174)

[8.1.6 Database Configure 25](#_Toc3786)

[8.1.7 Log Configure 25](#_Toc7165)

[8.2 Functions Switch Configure 26](#_Toc28989)

[appendix 26](#_Toc26658)

[Appendix 1 Type Define 26](#_Toc6135)

[Appendix 2 Protocol Define 26](#_Toc8599)

[Appendix 3 Transformation Definition 26](#_Toc30029)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Authorize Service Docment | |
| Be A Version： | V3.3.0.1001 | |
| Released： | 2022-11-01 | |
| 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.Network authorization verification can be performed through this interface. For other management interfaces, please refer to the HTTP interface.

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 XCrypto algorithm. If encryption is required, please fill, otherwise please fill in 0

## 3.1 Login Protocol

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

### 3.1.1 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; //can be 0

ENUM\_PROTOCOLDEVICE\_TYPE enDeviceType; //must set

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

### 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\_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
* 250:enDeviceType does not set
* 251:user not found
* 252:password is error
* 253:loginned
* 254:User does not have permission
* 255:user timeout
* 256:Server internal error

## 3.2 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 authorize

Use to login

### 4.2.1 Request

{

"unOperatorType":2,

"unOperatorCode":8197,

"byIsReply":1,

"wCrypto":0,

"st\_UserAuth":{

"tszUserName":"123123aa",

"tszUserPass":"123123",

"enClientType":10,

"enDeviceType":25

}

}

### 4.2.2 Reply

{  
    **"wHeader"**:**17**,  
    **"wTail"**:**255**,  
    **"unOperatorType"**:**2**,  
    **"unOperatorCode"**:**8198**,  
    **"wCrypto"**:**0**,  
    **"wReserve"**:**0**  
}

## 4.2 timeout protocol

The code value is :XENGINE\_COMMUNICATION\_PROTOCOL\_OPERATOR\_CODE\_AUTH\_TIMEDOUT

# 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 Client Management Interface

management requires Token verification. You need to log in to get the Token value and attach it to the Root of Json before it can be used

To get the value of Token, refer to the protocol in Section 5.4

{

"xhToken":100011213

}

### 5.1.1 Get User

#### 5.1.1.1 Request

Method:POST

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

Payload:

{

"xhToken":100011213,

"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:

{

"xhToken":100011213,

"Online":false,

"PosStart":0,

"PosEnd":100

}

#### 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:

{

"xhToken":100011213,

"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:

{

"xhToken":100011213,

"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"

}

## 5.2 Serial Management API

Requires Token Value,refer 5.1

### 5.2.1 List Serial

#### 5.2.1.1 Request

Method:POST

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

Payload:

{

"xhToken":100011213,

"PosStart":0,

"PosEnd":100

}

#### 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:

{

"xhToken":100011213,

"st\_SerialInfo":{

"enSerialType":1,

"nNumberCount":5,

"nSerialCount":2,

"tszHasTime":"20"

}

}

enSerialType:serial type

nNumberCount:serial field number

nSerialCount:serial number

tszHasTime:have time

#### 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:

{

"xhToken":100011213,

"Array":[

{

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

}

]

}

#### 5.2.3.2 Reply

{

"code":0,

"msg":"success"

}

## 5.3 User Interface

User api can be used without login

### 5.3.1 User Delete

#### 5.3.1.1 Request

Method:POST

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

Payload:

{

"st\_UserInfo" :

{

"nIDNumber" : 511025111111111111,

"nPhoneNumber" : 1366666666,

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

"tszUserName" : "123123aa",

"tszUserPass" : "123123"

}

}

#### 5.3.1.2 Reply

{

"code" : 0,

"msg" : "success"

}

### 5.3.2 User Register

#### 5.3.2.1 Request

Method:POST

API:http://IP:PORT/auth/user/register

Payload:

{

"st\_UserTable" :

{

"st\_UserInfo" :

{

"nIDNumber" : 511025111111111111,

"nPhoneNumber" : 1366666666,

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

"tszUserName" : "123123aa",

"tszUserPass" : "123123",

"nUserLevel":5

},

"tszHardCode" : "2FDWAD02JD2091"

}

}

#### 5.3.2.2 Reply

{

"code":0,

"msg":"success"

}

### 5.3.3 User Pay

#### 5.3.3.1 Request

Method:POST

API:http://IP:PORT/auth/user/pay

Payload:

{

"st\_UserPay" :

{

"tszSerialNumber" : "XAUTH-XYRYM-ZAA10-59NKW-KOFLP-35838-ZXC36-ZOVTH",

"tszUserName" : "123123aa"

}

}

#### 5.3.3.2 Reply

{

"code":0,

"msg":"success"

}

### 5.3.4 Get Back Password

#### 5.3.4.1 Request

Method:POST

API:http://IP:PORT/auth/user/pass

Payload:

{

"st\_UserInfo" :

{

"nIDNumber" : 511025111111111111,

"nPhoneNumber" : 1366666666,

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

"tszUserName" : "123123aa"

}

}

#### 5.3.4.2 Reply

{

"code":0,

"msg":"success",

"st\_UserAuth":{

"enClientType":0,

"enDeviceType":0,

"tszUserName":"123123aa",

"tszUserPass":"123123"

}

}

### 5.3.5 Get Left Time

#### 5.3.5.1 Request

Method:POST

API:http://IP:PORT/auth/user/time

Payload:

{

"st\_UserAuth" :

{

"tszUserName" : "123123aa",

"tszUserPass" : "123123"

}

}

#### 5.3.5.2 Reply

{

"st\_UserTime":{

"enDeviceType":0,

"enSerialType":1,

"nTimeLeft":2,

"nTimeONLine":3,

"tszLeftTime":"2022-08-25 13:56:36",

"tszUserAddr":"127.0.0.1:50168",

"tszUserName":"123123aa"

}

}

### 5.3.6 Fast 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

#### 5.3.6.1 Request

Method:POST

API:http://IP:PORT/auth/user/try

Payload:

{

"st\_UserTry":{

"tszSerial":"cpuid:112233"

}

}

#### 5.3.6.2 Reply

{

"code":0,

"msg":"success"

}

## 5.4 Token Protocol

Used to manage API permission verification management

### 5.4.1 Login

Used to get the Token value, this login support management and authorize verifcation login

#### 5.4.1.1 Request

Method:GET

API:http://IP:PORT/api?function=login&user=123123aa&pass=123123

Payload:none

#### 5.4.1.2 Reply

{

"code":0,

"msg":"success",

"nTimeout":3600,

"tszTimeEnd":"2022-08-25 15:19:12",

"tszTimeStart":"2022-08-25 14:19:12",

"xhToken":1000030115

}

Token has an expiration time. After expiration, you cannot use the management AP

Note:if you login for normal user.you have to update token and you have to get user timeout thought 5.3.5.only when set third-verification, otherwise server does not notify to you

### 5.4.2 UPDate

If you want to continue to use this Token, you can use the renewal agreement to update,provided that token exist.

#### 5.4.2.1 Request

Method:GET

API:http://IP:PORT/api?function=update&token=1000030115

Payload:none

#### 5.4.2.2 Reply

{

"code" : 0,

"msg" : "success",

"nTimeout" : 3600,

"tszTimeEnd" : "2022-08-25 15:31:42",

"tszTimeStart" : "2022-08-25 14:31:42",

"xhToken" : 1000032558

}

### 5.4.3 Close

If you does not use this token,you can close or wait auto destory when timeout.if login user is normal permission(> 1),then server update user time

#### 5.4.3.1 Request

Method:GET

API:http://IP:PORT/api?function=close&token=1000030115

Payload:None

#### 5.4.3.2 Reply

{

"code" : 0,

"msg" : "success"

}

## 5.5 Function Switch

The function switch interface is only a temporary function and will not work after reboot

### 5.5.1 Get Functions List

#### 5.5.1.1 Request

Method:POST

Interface:http://IP:PORT/auth/switch/get

Payload:

{

"xhToken":12345679

}

#### 5.5.1.2 Reply

{

"msg":"success",

"code":0,

"bSwitchDelete":true,

"bSwitchLogin":true,

"bSwitchPass":true,

"bSwitchPay":true,

"bSwitchRegister":true

}

### 5.5.2 Set Functions Switch

#### 5.5.2.1 Request

Method:POST

Interface:http://IP:PORT/auth/switch/set

Payload:

{

"xhToken":123456789,

"st\_SwitchInfo":{

"bSwitchDelete":true,

"bSwitchLogin":true,

"bSwitchPass":true,

"bSwitchPay":true,

"bSwitchRegister":true

}

}

#### 5.5.2.2 Reply

{

"msg":"success",

"code":0

}

# 六 Third-Verification

From version 3.2 after, the authentication service can handle user login, logout, timeout, etc. through authentication interface. If you want to manage the user table yourself, then you need to use this function. If this function is enabled, the server's user database will be no longer working

The Third-Ver uses the HTTP POST method, which will be submitted to the specified URL according to the configuration file

## 6.1 User Login

### 6.1.1 Request

Method:POST

URL:tszPassLogin of Configure

Payload:Response JSon of 5.3.4

### 6.1.2 Response

HTTP 200is success,other is failed.

If sucesss,you need response a user table to our,you can refer 5.1.4 JSON,must fill all field

## 6.2 User Logout

### 6.2.1 Request

Method:POST

URL:tszPassLogout of Configure

Payload:Response JSon of 5.3.5

### 6.2.2 Response

System does not care about response

## 6.3 User Timeout

### 6.3.1 Request

Method:POST

URL:tszPassTimeout of Configure

Payload:Response Json of 5.3.5

### 6.3.2 Response

System does not care about response

# 七 Advanced Features

## 7.1 Distributed Authorize

Distributed authentication is enabled by default, if users want distributed authentication, they can submit to our server through the auth/pass/interface of post, and the server will pass the authentication and return to the authentication side.

Server can process login, logout, expiration

You can also use the 6 sections as a validation forwarding service. The new version provides more flexible authentication methods

# 八 Configure Description

## 8.1 Basic Configure File

Configure File:XEngine\_Config.json

### 8.1.1 Basic Configure

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

### 8.1.2 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

### 8.1.3 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
* nTokenTimeout:Token Timeout,second.
* nVerTime:Quick Verification Time
* nVerMode:Quick Verification Mode
* nTryMode:Try Mode
* nTryTime:Try Time

### 8.1.4 Login Configure

Configure Infomation:XLogin

* bMultiLogin:Whether Allow Multi login
* bHTTPAuth:whether enable to http authorize verifcaiton,need http heartbeat
* bPassAuth:whether enable to third-ver
* nHTTPAuthTime:http authorize verification timeout,second.
* tszPassLogin:User Login API
* tszPassLogout:User Logout api
* tszPassTimeout:user timeout api

### 8.1.5 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

### 8.1.6 Database Configure

Configure Information:XSql

* tszSQLite:Database file address

### 8.1.7 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

## 8.2 Functions Switch Configure

Configure File:XEngine\_SwitchConfig.json

* bSwitchDelete:User Delete Switch
* bSwitchRegister:User Register Switch
* bSwitchLogin:User Login Switch
* bSwitchPay:User Pay Switch
* bSwitchPass:User Get Password Swtich
* bSwtichTime:User Get LeftTime Swtich

# 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