XEngine API服务文档

目录

[XEngine API服务文档 1](#_Toc28509)

[前言 4](#_Toc11518)

[阅读者 4](#_Toc898)

[概述 4](#_Toc2353)

[相关模块 4](#_Toc30243)

[一 技术结构 4](#_Toc10300)

[1.1 HTTP 4](#_Toc3666)

[二 配置环境 4](#_Toc29023)

[2.1 WINDOWS 4](#_Toc10486)

[2.1.1 配置环境 5](#_Toc15846)

[2.1.2 编译运行 5](#_Toc29229)

[2.2 LINUX 5](#_Toc18825)

[2.2.1 环境配置 5](#_Toc16013)

[2.2.2 编译运行 5](#_Toc10022)

[2.3 版本要求 5](#_Toc11169)

[2.3.1 系统版本 5](#_Toc25551)

[2.3.2 软件需求 6](#_Toc10516)

[三 接口协议 6](#_Toc5060)

[3.1 身份证信息查询 6](#_Toc22631)

[3.1.1 请求 6](#_Toc8153)

[3.1.2 回复 6](#_Toc24356)

[3.2 银行卡校验 7](#_Toc8011)

[3.2.1 请求 7](#_Toc26610)

[3.2.2 回复 7](#_Toc10523)

[3.3 语言转换 7](#_Toc12530)

[3.3.1 请求 7](#_Toc8551)

[3.3.2 回复 7](#_Toc14920)

[3.4 翻译 8](#_Toc27223)

[3.4.1 请求 8](#_Toc32199)

[3.4.2 回复 8](#_Toc23954)

[3.5 P2XP协议 8](#_Toc8465)

[3.5.1 加入网络 8](#_Toc23139)

[3.5.2 同步列表 9](#_Toc23984)

[3.6 分布式锁 10](#_Toc21999)

[3.6.1 请求 10](#_Toc30445)

[3.6.2 回复 10](#_Toc1580)

[3.7 行政邮编 10](#_Toc4982)

[3.7.1 请求 10](#_Toc17782)

[3.7.2 回复 11](#_Toc16651)

[3.8 日志系统 11](#_Toc31739)

[3.8.1 日志创建 11](#_Toc23248)

[3.8.2 日志插入 12](#_Toc1362)

[3.8.3 日志查询 12](#_Toc3832)

[3.8.4 日志删除 13](#_Toc31735)

[3.9 二维码 14](#_Toc26328)

[3.9.1 生成二维码 14](#_Toc24967)

[3.9.2 解析二维码 14](#_Toc3252)

[3.10 网络测试 15](#_Toc4588)

[3.10.1 创建 15](#_Toc9404)

[3.10.2 销毁 16](#_Toc8408)

[3.10.3 报告 16](#_Toc31499)

[3. A 模块插件系统 17](#_Toc31818)

[3.A.1 密码生成 17](#_Toc5814)

[3.A.2 生肖星座 17](#_Toc17384)

[3.A.3 标准身材 18](#_Toc546)

[3.A.4 时区转换 18](#_Toc1137)

[3.A.5 计量转换 20](#_Toc23358)

[3.A.6 电话信息查询 21](#_Toc16287)

[3.A.7 IP地址信息查询 21](#_Toc4689)

[3.B Lua插件系统 22](#_Toc11307)

[3.B.1 数学计算 22](#_Toc13438)

[3.B.2 时间获取 23](#_Toc14031)

[四 配置说明 23](#_Toc18224)

[4.1 服务器配置 23](#_Toc21626)

[4.1.1 基本配置 23](#_Toc9222)

[4.1.2 最大配置 23](#_Toc30488)

[4.1.3 时间配置 24](#_Toc245)

[4.1.4 日志配置 24](#_Toc16789)

[4.1.5 数据库接口配置 24](#_Toc28353)

[4.1.6 信息接口配置 24](#_Toc15625)

[4.1.7 插件配置 24](#_Toc10237)

[4.2 语言配置 24](#_Toc8237)

[4.3 插件配置 25](#_Toc7463)

[五 高级模式 25](#_Toc903)

[5.1 Lib插件编写 25](#_Toc20136)

[5.2 Lua插件编写 25](#_Toc28660)

[附录 26](#_Toc19381)

[附录1 类型定义 26](#_Toc25203)

[附录2 协议定义 26](#_Toc9578)

[附录3 转换定义 26](#_Toc30560)

[附录4 更新历史 26](#_Toc24474)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布 | 文件名称： | XEngin API服务文档 | |
| 适用版本： | V2.8.0.1001 | |
| 发布日期： | 2023-02-07 | |
| 拟 制： qyt | | |

# 前言

## 阅读者

开发人员,测试人员,QA

## **概述**

此文档包含了存储服务的相关技术说明和接口定义!

## 相关模块

此服务使用了XEngine作为开发包.要使用此服务代码必须配置安装好XEngine开发环境.

# 一 技术结构

服务器基于C/C++实现,协议基于HTTP实现

## HTTP

消息通过Get或者Post方法

# 二 配置环境

## 2.1 WINDOWS

需要下载XEngine.

通过VS来编写代码,编译,调试.

### 2.1.1 配置环境

按照XEngine Readme文件的说明配置环境

### 2.1.2 编译运行

在完成环境配置后.你可以进入代码目录.使用VS打开XEngine.sln,然后选择x86 debug 编译.

如果环境没有错误,编译会直接成功

成功后,你需要拷贝XEngine\_Release下的文件到你编译的目录下.运行XEngine\_\*App.exe即可.

注意：XEngine环境可以通过VSCopy脚本拷贝到你的编译目录下,前提是你配置好你的XEngine环境.

## 2.2 LINUX

### 2.2.1 环境配置

如果使用LINUX来开发运行,那么你需要在UBUNTU或者CENTOS下面才可以使用,UBUNTU需要20.04 以上系统.CENTOS需要8.x版本(兼容所有类centos版本)

### 2.2.2 编译运行

配置完环境后,你可以编译它,打开终端,进入源码目录执行以下命令

编译:make

安装:make FLAGS=InstallAll

清理:make FLAGS=CleanAll

如果没有错误,你可以在XEngine\_Release目录下看到编译的XEngine\_\*App

## 2.3 版本要求

### 2.3.1 系统版本

最低版本要求:

WINDOWS: win7 sp1

Ubuntu:20.04

Centos:8.x

MacOS:12

### 2.3.2 软件需求

最低版本要求:

XEngine:V7.38

# 三 接口协议

## 3.1 身份证信息查询

身份证查询会验证身份证号码是否正确

### 3.1.1 请求

方法:GET

参数:params1 = IP地址

地址:http://127.0.0.1:5501/api?function=id&params1=511025191101018792

回复:nSex:1代表男0代表女.

nCheck:表示校验码

nPoliceID:表示派出所ID

### 3.1.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"nBirthDay"**:**1**,  
        **"nBirthMonth"**:**7**,  
        **"nBirthYear"**:**1988**,  
        **"nCheck"**:**2**,  
        **"nPoliceID"**:**87**,  
        **"nSex"**:**1**,  
        **"tszCity"**:**"内江市"**,  
        **"tszCounty"**:**"资中县"**,  
        **"tszIDNumber"**:**"511025191101018792"**,  
        **"tszProvincer"**:**"四川省"**  
    },  
    **"msg"**:**"success"**  
}

## 3.2 银行卡校验

### 3.2.1 请求

方法:GET

参数:params1 = 银行卡号

地址:http://127.0.0.1:5501/api?function=bank&params1=6214000000

### 3.2.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"enBankType"**:**2**,  
        **"tszBankAbridge"**:**"CMB"**,  
        **"tszBankName"**:**"招商银行"**,  
        **"tszBankNumber"**:**"621400000"**  
    },  
    **"msg"**:**"success"**  
}

## 3.3 语言转换

### 3.3.1 请求

方法:GET

参数:params1 = 银行卡号

params2 = 转换类型

地址:http://127.0.0.1:5501/api?function=language&params1=繁体到简体&params2=1

### 3.3.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"enType"**:**1**,  
        **"tszDestStr"**:**"繁體到簡體"**,  
        **"tszSourceStr"**:**"繁体到简体"**  
    },  
    **"msg"**:**"success"**  
}

## 3.4 翻译

### 3.4.1 请求

方法:GET

参数:params1 = 要翻译的内容,UTF8

Params2 = 翻译类型,参考ENUM\_XENGINE\_APISERVICE\_TRANSLATION\_TYPE

地址:http://127.0.0.1:5501/api?function=translation&params1=中国&params2=0

### 3.4.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"enType"**:0,  
        **"tszDestStr"**:**"china"**,  
        **"tszSourceStr"**:**"中国"**  
    },  
    **"msg"**:**"success"**  
}

## 3.5 P2XP协议

P2XP同样使用HTTP协议实现,P2XP需要用到用户信息,要使用P2XP协议,调用者可能需要单独开发一套验证接口.用于验证用户名信息.

当用户通过P2XP登录后,每隔5秒还需要进行心跳保持信息,只有超过指定时间没收到,那么才会认为不在线.

### 3.5.1 加入网络

只有发送这个协议请求,服务器才会记录这个客户端,客户端才能加入P2XP网络,否则将无法使用P2XP网络.

此功能主要为了跨网段使用分布式存储网络,如果你没有这方便的需求,可以不使用

#### 3.5.1.1 请求

方法:POST

参数:params1 = 操作码的十进制,参考协议定义

地址:http://192.168.1.12:5501/api?function=p2p&params1=24577

协议体:

{  
    **"tszUserName"**:**"用户名,由使用者管理,服务器不做验证"**,  
    **"tszPrivateAddr"**:**"私有地址,本机内部IP地址"**,  
    **"tszPublicAddr"**:**"公有地址"**,  
    **"dwConnectType"**:**0**,  
    **"dwPeerType"**:**0**  
}

#### 3.5.1.2 回复

{  
    **"code"**:**0**,  
    **"msg"**:**"success"**  
}

### 3.5.2 同步列表

内网地址只同步内网IP,外网地址同步可以同步在这个外网IP地址下面的所有内网IP地址,对于一些大型网络公司和环境复杂的内网,使用此功能可以方便判断出在同一网络下的用户.

#### 3.5.2.1 请求

方法:POST

参数:params1 = 操作码的十进制,参考协议定义

地址:http://192.168.1.12:5501/api?function=p2p&params1=24579

协议体:

{  
    **"tszUserName"**:**"用户名,由使用者管理,服务器不做验证"**,  
    **"tszPrivateAddr"**:**"私有地址,本机内部IP地址"**,  
    **"tszPublicAddr"**:**"公有地址"**  
}

#### 3.5.2.2 回复

{  
    **"code"**:**0**,  
    **"msg"**:**"sucess"**,  
    **"ClientCount"**:**2**,  
    **"ClientArray"**:[  
        {  
            **"ClientAddr"**:**"192.168.1.101"**,  
            **"ClientUser"**:**"123123aa"**  
        },  
        {  
            **"ClientAddr"**:**"192.168.1.102"**,  
            **"ClientUser"**:**"123123bb"**  
        }  
    ]  
}

## 3.6 分布式锁

### 3.6.1 请求

方法:GET

参数:无

地址:http://127.0.0.1:5501/api?function=lock&params1=token&params2=0

Params1为token值,可以创建也可以自己指定.

Params2为类型,参考用户协议定义ENUM\_XENGINE\_APISERVICE\_LOCKER\_TYPE

负载:无

### 3.6.2 回复

{

"code":0,

"data":{

"xhToken":1000106561

},

"msg":"success"

}

## 3.7 行政邮编

### 3.7.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=zipcode&params1=0

参数:params1,0通过ZIPCode查询,1通过地址名称查询

负载:

0:

{

"nZipCode":100010

}

1:

{

"tszCounty":"资中县"

}

### 3.7.2 回复

{

"code":0,

"data":{

"dlLat":39.93157,

"dlLng":116.41005,

"nAreaCode":10,

"nLevel":3,

"nZipCode":100010,

"tszCity":"北京市",

"tszCounty":"东城",

"tszPinYin":"dongcheng",

"tszProvincer":"北京"

},

"msg":"success"

}

## 3.8 日志系统

日志系统可以做日志查询服务,用于永久日志保存

### 3.8.1 日志创建

#### 3.8.1.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=log&params1=0

参数:params1=0 表示创建,1表示插入,2表示查询,3表示删除

负载:

{

"tszTableName":"xengine"

}

#### 3.8.1.2 回复

{

"code":0

}

### 3.8.2 日志插入

#### 3.8.2.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=log&params1=1

负载:

{

"tszTableName":"xengine",

"tszLogBuffer":"d12d",

"nLogSize":4,

"tszFileName":"file.cpp",

"tszFuncName":"xengine\_file\_insert",

"tszLogTimer":"2023-01-13 22:10:01",

"nLogLine":102,

"nLogLevel":4

}

#### 3.8.2.2 回复

{

"code":0

}

### 3.8.3 日志查询

#### 3.8.3.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=log&params1=2

负载:

{

"tszTableName":"xengine",

"tszTimeStart":"2023-01-13 22:10:00",

"tszTimeEnd":"2023-01-13 22:10:02"

}

#### 3.8.3.2 回复

{

"code":0,

"data":[

{

"nLogLevel":4,

"nLogLine":102,

"nLogSize":0,

"tszFileName":"file.cpp",

"tszFuncName":"xengine\_file\_insert",

"tszLogBuffer":"adwdad12d21d123d132rd213d32f23df23rf",

"tszLogTimer":"2023-01-13 22:10:01",

"tszTableName":"xengine"

},

{

"nLogLevel":4,

"nLogLine":102,

"nLogSize":0,

"tszFileName":"file.cpp",

"tszFuncName":"xengine\_file\_insert",

"tszLogBuffer":"adwdad12d21d123d132rd213d32f23df23rf",

"tszLogTimer":"2023-01-13 22:10:01",

"tszTableName":"xengine"

}

],

"msg":"success"

}

### 3.8.4 日志删除

#### 3.8.4.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=log&params1=3

负载:

{

"tszTableName":"xengine"

}

#### 3.8.4.2 回复

{

"code":0

}

## 3.9 二维码

二维码服务支持解析和生成

### 3.9.1 生成二维码

#### 3.9.1.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=qrcode&params1=0

负载:

{

"tszMsgBuffer":"要编码的二维码数据",

"tszFmtBuffer":"要返回的图片格式,比如输入:.png"

}

#### 3.9.1.2 回复

Content-Type: image/png;

返回图片格式,可以直接写到文件或者读取到内存

### 3.9.2 解析二维码

#### 3.9.2.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=qrcode&params1=1

类型:Content-Type: image/png

负载:二进制图片

#### 3.9.2.2 回复

返回解析完毕的二维码字符串信息

## 3.10 网络测试

可以对一个服务进行测试,可以用于生成报告或者对服务进行存活性判断

### 3.10.1 创建

#### 3.10.1.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=test&params1=0

负载:

{

"tszAPIUrl":"报告的POST URL地址",

"bTCP":true,

"bConn":true,

"tszAddr":"测试IP地址",

"nPort":3555,

"nCloseWaitContTime":2000,

"nConnectCount":10,

"nConnectTest":1,

"nContWaitTime":1000,

"tszSDBuffer":"发送的数据",

"tszRVBuffer":"接受数据,如果需要验证的话",

"nRVLen":1024,

"nSDLen":1024

}

#### 3.10.1.2 回复

{

"code":0

}

### 3.10.2 销毁

#### 3.10.2.1 请求

方法:POST

地址:http://127.0.0.1:5501/api?function=test&params1=1

负载:

{

"xhToken":123123,

"bTCP":true,

"bConn":true

}

#### 3.10.2.2 回复

{

"code":0

}

### 3.10.3 报告

#### 3.10.3.1 请求

方法:POST

地址:创建的tszAPIUrl参数

负载:

{

"lpszIPAddr":"测试的地址",

"nIPPort":3555,

"xhToken":3133131,

"nNumber":10,

"nFailed":0,

"nSuccess":10,

"nStatus":0

}

#### 3.10.3.2 回复

无

## A 模块插件系统

### 3.A.1 密码生成

#### 3.A.1.1 请求

方法:GET

参数:params1 = 生成的类型,1为随机,2为数字,3为字母

params2 = 密码长度

地址:http://192.168.1.12:5501/api?function=pass&params1=1&params2=4

#### 3.A.1.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"tszParamLength"**:**"4"**,  
        **"tszParamType"**:**"1"**,  
        **"tszPassword"**:**"9xKm"**  
    },  
    **"msg"**:**"success"**  
}

### 3.A.2 生肖星座

#### 3.A.2.1 请求

方法:GET

参数:params1 = 要处理日期,比如1988年1月21号为:19880121

params2 = 密码长度

地址:http://192.168.1.12:5501/api?function=zodiac&params1=19880121

#### 3.A.2.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"tszChineseZodiac"**:**"龙"**,  
        **"tszDate"**:**"19880121"**,  
        **"tszEnglishZodiac"**:**"水瓶座"**  
    },  
    **"msg"**:**"success"**  
}

### 3.A.3 标准身材

#### 3.A.3.1 请求

方法:GET

参数:params1 = 身高,单位米

params2 = 体重,单位公斤

地址:http://127.0.0.1:5501/api?function=bmindex&params1=1.78&params2=62

#### 3.A.3.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"Result"**:**"标准"**,  
        **"flHigh"**:**1.78**,  
        **"flValue"**:**19.568236333796236**,  
        **"flWeight"**:**62**  
    },  
    **"msg"**:**"success"**  
}

### 3.A.4 时区转换

#### 3.A.4.1 总数查询

##### 3.A.4.1.1 请求

方法:GET

参数:params1 = 类型,0为查询时区个数

地址:http://127.0.0.1:5501/api?function=timezone&params1=0

##### 3.A.4.1.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"Count"**:**91**  
    },  
    **"msg"**:**"success"**  
}

#### 3.A.4.2 列表查询

##### 3.A.4.2.1 请求

方法:GET

参数:params1 = 类型1为列表查询

params2 = 要查询的列表范围,不能太大,否则会造成缓冲区问题

地址:http://127.0.0.1:5501/api?function=timezone&params1=1&params2=1-10

##### 3.A.4.2.2 回复

{  
    **"Array"**:[  
        {  
            **"tszTimeCountry"**:**"大西洋标准时间（加拿大）"**,  
            **"tszTimeZone"**:**"AST"**  
        },  
        {  
            **"tszTimeCountry"**:**"夏威夷-阿拉斯加标准时间"**,  
            **"tszTimeZone"**:**"AHST"**  
        },  
        {  
            **"tszTimeCountry"**:**"以色列标准时间"**,  
            **"tszTimeZone"**:**"IST"**  
        }

],  
    **"Count"**:**3**,  
    **"code"**:**0**,  
    **"msg"**:**"success"**,  
    **"nPosEnd"**:**3**,  
    **"nPosStat"**:**1**  
}

#### 3.A.4.3 时区转换

##### 3.A.4.3.1 请求

方法:GET

参数:params2 = 类型2为时区转换

params2 = 要转换的类型

Params3 = 时间,时间格式是固定的

地址:http://127.0.0.1:5501/api?function=timezone&params1=2&params2=AST&params3=2022-07-21\_12:10:01

##### 3.A.4.3.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"tszTimeCountry"**:**"大西洋标准时间（加拿大）"**,  
        **"tszTimeStr"**:**"2022-07-21 08:10:01"**,  
        **"tszTimeZone"**:**"AST"**  
    },  
    **"msg"**:**"success"**  
}

### 3.A.5 计量转换

#### 3.A.5.1 请求

方法:GET

参数:type=类型0长度,1温度,2功率,3速度

source=原始子类型,参考导出头文件

value=要转换的值

地址:http://127.0.0.1:5501/api?function=meter&type=0&source=0&value=10

#### 3.A.5.2 回复

根据type类型不同返回的值也不同,建议自己测试

### 3.A.6 电话信息查询

#### 3.A.6.1 请求

方法:GET

参数:params1 = 电话号码

地址:http://127.0.0.1:5501/api?function=phone&params1=13699428888

#### 3.A.6.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"nAreaCode"**:**28**,  
        **"nPhoneNumber"**:**13699428888**,  
        **"nZipCode"**:**610000**,  
        **"tszCity"**:**"成都"**,  
        **"tszProvincer"**:**"四川"**  
    },  
    **"msg"**:**"success"**  
}

### 3.A.7 IP地址信息查询

#### 3.A.7.1 请求

方法:GET

参数:params1 = IP地址

地址:http://127.0.0.1:5501/api?function=ip&params1=1.29.164.255

#### 3.A.7.2 回复

{  
    **"code"**:**0**,  
    **"data"**:{  
        **"tszIPAddr"**:**"1.29.164.255"**,  
        **"tszIPAddress"**:**"内蒙古通辽市霍林郭勒市"**,  
        **"tszIPCity"**:**"通辽市"**,  
        **"tszIPCountry"**:**"中国"**,  
        **"tszIPCounty"**:**"霍林郭勒市"**,  
        **"tszIPISP"**:**"联通"**,  
        **"tszIPProvince"**:**"内蒙古"**  
    },  
    **"msg"**:**"success"**  
}

##### 3.A.7.2.1 错误回复

如果http不是200,或者返回的code不是0,那么表示错误.

msg只有json返回类型才有,数据流返回类型只有code,下同

{  
    **"code"**:**1001**,  
    **"msg"**:**"parse is incorrent"**  
}

## 3.B Lua插件系统

### 3.B.1 数学计算

#### 3.B.1.1 请求

方法:GET

参数:param1=操作的数

param2=被操作的数

type=操作类型,0相加,1相减,2相乘,3相除

地址:http://127.0.0.1:5501/api?function=cal&param1=1&param2=2&type=0

#### 3.B.1.2 回复

{

"code":0,

"msg":"success",

"data":{

"nType":0,

"nValue1":1,

"nValue2":1,

"nCal":2

}

}

### 3.B.2 时间获取

#### 3.B.2.1 请求

方法:GET

参数:type=操作的类型

0获取系统时间,1程序运行时间(秒),2系统时间戳

地址:http://127.0.0.1:5501/api?function=time&type=1

#### 4.B.2.2 回复

{

"code":0,

"msg":"success",

"data":{

"Time":"2023-01-13 14:52:05"

}

}

# 四 配置说明

## 4.1 服务器配置

基础配置文件:XEngine\_Config.json

### 4.1.1 基本配置

* tszIPAddr: 本机IP地址
* bDeamon: 1为守护进程后台运行0为前台运行
* nHttpPort:HTTP通信端口

### 4.1.2 最大配置

XMax 配置

* nMaxClient 最大允许客户端
* nMaxQueue 最大允许队列
* nIOThread:网络IO处理线程个数
* nHttpThread:HTTP处理线程个数

### 4.1.3 时间配置

XTime配置

* nTimeCheck:检测次数
* nP2PTimeOut:P2P超时时间
* nHttpTimeOut:HTTP客户端过期时间

### 4.1.4 日志配置

XLog 配置

* MaxSize:日志文件大小
* MaxCount:日志文件个数
* LogLeave:允许保存的级别

### 4.1.5 数据库接口配置

XSql 配置

* SQLAddr:数据库地址
* SQLPort:数据库端口
* SQLUser:数据库用户
* SQLPass:数据库密码

### 4.1.6 信息接口配置

XApi配置

* tszBankUrl:银行卡验证接口地址
* tszTranslationUrl:翻译接口地址

### 4.1.7 插件配置

XPlugin 配置

* bEnable:是否启用插件系统
* tszPluginLib:lib插件配置文件地址
* tszPluginLua:lua插件配置文件地址

## 4.2 语言配置

配置文件XEngine\_OPenccConfig.json

参考API服务协议头文件里面的定义

## 4.3 插件配置

配置文件XEngine\_PluginLib.json XEngine\_PluginLua.json

这是一个JSON数组.意思如下:

* PluginEnable:是否启用
* PluginMethod:要注册的方法名称,当function= 此名称的时候系统会自动交给模块处理
* PluginFile:模块文件地址

# 五 高级模式

## 5.1 Lib插件编写

模块插件是由编程语言编写而来的dll或者so或者dylib模块.模块插件需要实现PluginCore\_Init,PluginCore\_UnInit,PluginCore\_Call三个函数

可以参考XEngine\_PluginModule目录下的模块项目代码.

导出的接口都是一致的,你必须实现以下接口.

extern "C" *BOOL* PluginCore\_Init(*LPVOID* lParam = *NULL*);

extern "C" *void* PluginCore\_UnInit();

extern "C" *BOOL* PluginCore\_Call(*TCHAR* \* \*\*pppHDRList, int nListCount, int\* pInt\_HTTPCode, *TCHAR* \* ptszMsgBuffer, int\* pInt\_MsgLen, *LPCTSTR* lpszMsgBuffer = *NULL*, int nMsgLen = 0);

你还需要在插件配置文件里面启用你的插件,这样你就可以使用你的插件系统了

PluginCore\_Init:一个无类型指针参数,1个boolean返回值,在插件系统初始化的时候自动调用.

PluginCore\_UnInit:无返回值无参数,在卸载插件的时候会自动调用

PluginCore\_Call:拥有7个参数和一个boolean返回值.查找到指定插件的时候调用一次.

* pppHDRList:三级字符指针.是一个请求的URL头.比如http://192.168.1.8:5501/api?function=cal&params1=1&params2=1&param3=0后面的http://192.168.1.8:5501/api?function=calparams1=1 params2=1 param3=0,这里是三个字符串数组
* nListCount:请求的URL头的参数个数,这里是三个
* pInt\_HTTPCode:整数型指针,输出HTTP状态码
* ptszMsgBuffer:字符指针,输出负载的内容
* pInt\_MsgLen:输出负载的内容大小
* lpszMsgBuffer:输入请求的HTTP负载内容,只有nMsgLen大于0才有效
* nMsgLen:输入请求的负载内容大小.

## 5.2 Lua插件编写

你现在可以通过lua脚本来编写插件了,不需要在学习编程语言.极大方便了大家开发自己的查询系统.你可以参考XEngine\_LuaPlugin目录下的脚本

lua插件需要实现PluginCore\_Init,PluginCore\_UnInit,PluginCore\_Call三个函数

PluginCore\_Init:一个无类型指针参数,1个boolean返回值,在插件系统初始化的时候自动调用.

PluginCore\_UnInit:无返回值无参数,在卸载插件的时候会自动调用

PluginCore\_Call:拥有4个参数和一个boolean返回值.查找到指定插件的时候调用一次.

* lpszStrUrl:URL头参数,用于function=api? 后面的参数内容,比如param1=1&param2=2&param3=0
* nListCount:表示头参数个数,这里是3
* lpszMsgBuffer:负载的请求数据,如果nMsgLen大于0表示存在
* nMsgLen:负载数据的大小

# 附录

## 附录1 类型定义

参考文件 XEngine\_CommHdr.h

## 附录2 协议定义

参考文件XEngine\_ProtocolHdr.h

## 附录3 转换定义

参考文件 XEngine\_Types.h

## 附录4 更新历史

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