XEngine Storage Service Docment

目录

[XEngine网络通信引擎协议文档 1](#_Toc5143)

[前言 5](#_Toc25646)

[阅读者 5](#_Toc2032)

[概述 5](#_Toc12165)

[协议定义 5](#_Toc21349)

[相关模块 5](#_Toc24153)

[一 协议规范 5](#_Toc18035)

[1.1 协议头说明 6](#_Toc27956)

[1.2 协议头格式 6](#_Toc11765)

[1.2.1 普通协议头 6](#_Toc5655)

[1.2.2 扩展协议头 6](#_Toc16946)

[1.3 协议头字段 7](#_Toc9551)

[1.3.1 普通协议头字段 7](#_Toc18140)

[1.3.2 扩展协议头字段 7](#_Toc15865)

[1.4 协议头表示 8](#_Toc20487)

[1.5 协议展示 8](#_Toc21400)

[1.5.1 普通协议 8](#_Toc31718)

[1.5.2 扩展协议 8](#_Toc2410)

[二 使用协议 9](#_Toc12103)

[2.1 主协议 9](#_Toc28001)

[2.2 子协议 9](#_Toc4909)

[2.3 协议分类 9](#_Toc29677)

[三 系统协议 10](#_Toc17397)

[3.1 日志协议 10](#_Toc754)

[3.2 验证协议 11](#_Toc26346)

[3.2.1 删除协议 11](#_Toc3494)

[3.2.2 注册协议 12](#_Toc16694)

[3.2.3 登录协议 13](#_Toc4139)

[3.2.4 充值协议 14](#_Toc908)

[3.2.5 找回密码 15](#_Toc28956)

[3.2.6 获取时间 16](#_Toc22533)

[3.2.7 网络验证 18](#_Toc16491)

[3.2.8 公告协议 18](#_Toc27655)

[3.2.9 超时通知 19](#_Toc17981)

[3.3 流媒体协议 19](#_Toc15912)

[3.3.1 推送协议 19](#_Toc10672)

[3.3.2 拉流协议 21](#_Toc30831)

[3.3.3 控制协议 23](#_Toc30371)

[3.3.4 通知协议 24](#_Toc8547)

[3.4 存储服务协议 24](#_Toc17151)

[3.4.1 文件传输协议 24](#_Toc709)

[3.4.2 上传文件协议 25](#_Toc30485)

[3.4.3 查询文件协议 26](#_Toc23275)

[3.4.4 删除文件协议 27](#_Toc6662)

[3.4.5 下载文件协议 28](#_Toc22033)

[3.4.10 文件夹查询协议 30](#_Toc19025)

[3.4.11 文件夹创建协议 31](#_Toc30493)

[3.4.12 文件夹删除协议 32](#_Toc3072)

[3.4.13 用户信息查询 33](#_Toc5040)

[3.5 后台服务协议 34](#_Toc8456)

[3.5.1 协议处理 34](#_Toc25968)

[3.5.2 后台管理协议 34](#_Toc1497)

[3.6 P2XP协议 36](#_Toc752)

[3.6.1 登录协议 37](#_Toc15147)

[3.6.2 内网地址同步 38](#_Toc25769)

[3.6.3 外网地址同步 39](#_Toc3986)

[3.6.4 用户查询 41](#_Toc32474)

[3.6.5 请求连接 42](#_Toc28261)

[3.7 消息队列 43](#_Toc32590)

[3.7.1 投递包协议 44](#_Toc9960)

[3.7.2 获取包协议 45](#_Toc30287)

[3.7.3 删除包协议 46](#_Toc30185)

[3.8 消息分发 47](#_Toc11356)

[3.8.1 请求应答协议 47](#_Toc11568)

[3.8.2 通知协议 49](#_Toc12887)

[3.9 UDX协议 51](#_Toc28525)

[3.9.1 数据传输协议 51](#_Toc26684)

[3.9.2 登录协议 51](#_Toc27782)

[3.9.3 数据重传 52](#_Toc22197)

[3.9.4 通知协议 53](#_Toc10614)

[3.10 XRPC协议 54](#_Toc23028)

[3.10.1 函数调用 54](#_Toc30758)

[3.10.2 命令调用 57](#_Toc20671)

[3.11 心跳服务 57](#_Toc9943)

[3.11.1 心跳同步 57](#_Toc22683)

[3.12 分包协议 58](#_Toc27674)

[3.12.1 分包开始 58](#_Toc10348)

[3.12.2 分包结束 59](#_Toc31604)

[3.13 离开协议 59](#_Toc25871)

[四 用户协议 60](#_Toc20502)

[4.1 用户协议规范 60](#_Toc26543)

[五 特别说明 60](#_Toc26188)

[5.1 协议头 60](#_Toc27626)

[5.1.1 加密类型 60](#_Toc8608)

[5.1.2 负载类型 60](#_Toc3075)

[5.2 权限级别 61](#_Toc19639)

[附录 61](#_Toc25521)

[附录1 类型定义 61](#_Toc18803)

[附录2 协议定义 61](#_Toc13895)

[附录3 转换定义 61](#_Toc31467)

[附录4 更新历史 61](#_Toc6991)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Storage Service Docment | |
| Be A Version： | V1.3 | |
| Released： | 2021-07-23 | |
| Writer： qyt | | |

# Preface

## Reader

Development ,test

## **Overview**

This document contains related technical descriptions and interface definitions for storage services!

## 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

Use http protocol as the basic communication protocol.upload,download,manage interface are all http.

Should be bind three port when Start service Distinguish upload, download, and management.

Three ports cannot be used across ports

Support third-party service interface, support NGINX as upload and download engine.

## Upload interface

The upload interface is implemented by HTTP,upload interface need to be implemented using the put of the http.

The upload interface is not used form-data field.

The upload path need to create by user.

Such as: PUT /dir/name HTTP/1.1 get url path....

## Download interface

Download interface is implement through get.

Get Download file through url

## Mangement Interface

Mangement interface is implement through post,POST can be empty of body,can be contain body of the json.

Manage interface need to provided api/type/name through triplet situation.

Here is:the API version, API type, and API name.

# 二 Configure Env

## 2.1 WINDOWS

Need to download XEngine.

Complie and run and debug by vs2019.

Download address:https://gitee.com/xengine/libxengine

### 2.1.1 Configure Environment

After Download XEngine,if you download is zip file.you have to uncompress for xengine and add user environment value.

You need add to follow two user environment in your system

* XEngine\_Include  XEngine is header path
* XEngine\_Library  XEngine is library path

比如:



You need jsoncpp env.you can install through vcpkg

And you can download for youself:https://github.com/open-source-parsers/jsoncpp/

If you download for youself,you need install and complie by youself and configure project attributes vc++ path in your vs2019

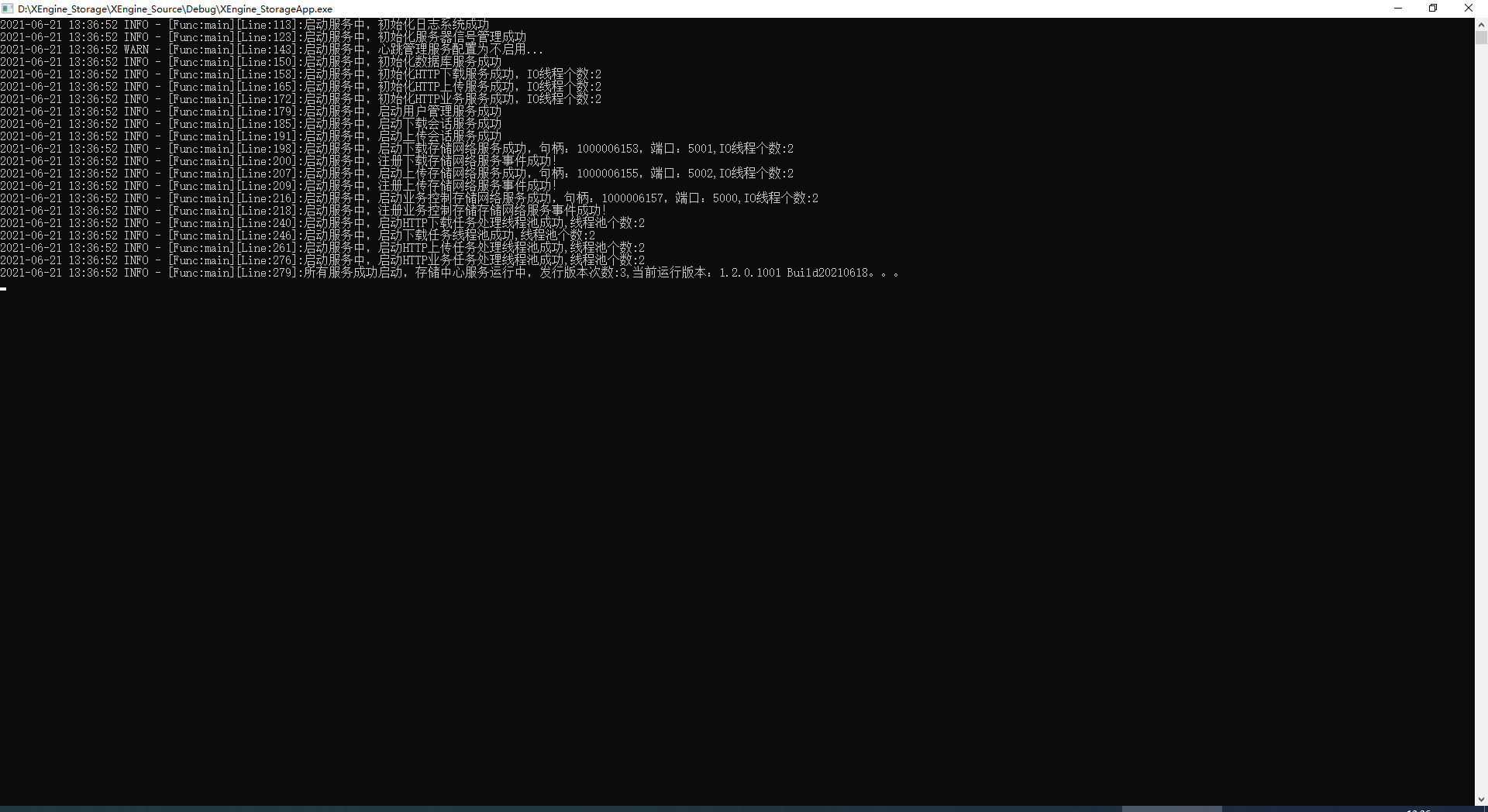
### 2.1.2 complie and run

When you complete with configuration.you can come in code path.open XEngine\_StorageApp.sln by vs2019.and choice x86(debug or release) or x86(just release) complie

If environment not have error.complie is succesed.contrain five module and one exe program

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

If not have error,you can see follow the infomation



Note: You can run the program directly, the system will prompt you what you need, you can directly enter the XEngine directory to search.

## 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STREAM版本)

在下载完毕后,你可能需要解压,解压后,在目录里面找到脚本安装文件并且执行下面的命令:

sudo XEngine\_RunEnv.sh -c 3

即可完成XEngine的环境配置.

当然,还需要JSONCPP的环境,你可以通过命令安装

Ubuntu:sudo apt install libjsoncpp-devel -y

Centos:sudo dnf install jsoncpp-devel -y

### 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\_StorageApp

然后直接在终端运行它即可.如果没有错误,你可以看到下面的信息:



## 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

### 2.3.2 软件需求

最低版本要求:

XEngine:V7.14

JsonCpp:V1.9.2

# 三 接口协议

我们的接口都是POST接口协议,也就是业务端口.

## 3.1 查询协议

### 3.1.1 文件列表

查询接口:/api/query/file

负载内容:可空或者如下:

{  
    **"lpszTimeStart"**:**"开始时间,可NULL"**,  
    **"lpszTimeEnd"**:**"结束时间,可NULL"**,  
    **"lpszFileName"**:**"文件名,可NULL"**,  
    **"lpszFileHash"**:**"文件HASH,可NULL"**  
}

## 3.2 三方接口

三方接口是用来接入第三方服务器的接口,目前支持NGINX的nginx upload module模块上传接口以及NGINX的下载代理转发接口

### 3.2.1 用户验证

用户验证目前仅仅支持BASIC的HTTP验证.你可以通过配置我们的UserList.txt文件来实现用户验证.也可以使用HTTP PASS代理来实现验证.

#### 3.2.1.1 本地验证

在XEngine\_Config文件夹的UserList.txt里面有用户列表.

每一行代表一个用户,中间用空格分开,前面表示用户名,后面表示密码.你只需要配置这个文件即可实现HTTP 基本的验证.没有验证通过将无法继续操作.

#### 3.2.1.2 代理验证

代理验证是由服务器发送一条HTTP POST消息给一个指定地址,返回200才表示成功,其他值表示失败.

发送的HTTP POST代理地址由用户指定,负载内容由服务器指定,负载为JSON格式,内容如下:

{  
    **"lpszPostUrl"**:**"客户端提交的URL"**,  
    **"lpszClientAddr"**:**"客户端的IP地址"**,  
    **"lpszUser"**:**"用户"**,  
    **"lpszPass"**:**"密码"**  
}

如果成功,那么需要返回200.

### 3.2.2 上传代理

nginx上传是用的nginx upload module 实现的.在你所有安装配置成功后,需要使用

proxy\_pass <http://192.168.1.7:5000/Api/Event/UPFile;> 指向我们的服务器.服务器会接受到代理转发的结果,并且返回.

### 3.2.3 完成通知

完成通知表示服务器在接受到一个上传和下载请求处理完毕后,是否需要给指定服务发送一个HTTP POST协议的通知.可以通过配置文件配置.

#### 3.2.3.1 上传通知

#### 3.2.3.2 下载通知

# 四 配置说明

## 4.1 基本配置

* bDeamon: 1为守护进程后台运行0为前台运行
* nCenterPort:业务处理端口,HTTP POST处理端口
* nStorageDLPort:下载端口GET 协议
* nStorageUPPort:上传端口 PUT协议

## 4.2 最大配置

XMax 配置

* MaxClient 最大允许客户端
* MaxQueue 最大允许队列
* IOThread:网络IO处理线程个数
* CenterThread:业务处理线程个数
* nStorageUPThread:上传处理线程个数
* nStorageDLThread:下载处理线程个数

## 4.3 时间配置

XTime配置

* bHBTime是否启用,1为启用0不启用心跳
* nDBMonth:数据库保存时间,默认月
* nTimeCheck:检测次数
* nStorageTimeOut:间隔多久检查一次

## 4.4 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

## 4.5 数据库配置

XSql 配置.需要MYSQL的支持

## 4.6 存储配置

XStorage 配置

* nHashMode:HASH算法,1MD5,2HASH1,具体值查看OPENSSL的定义
* bRename:是否自动改名和路径,对于NGINX上传的文件有效
* tszHttpAddr:本机IP地址
* tszNginAddr:NGINX服务地址,可不启用
* tszFileDir:保存的路径

## 4.7 代理配置

XProxy 配置

### 4.7.1 验证代理

XProxyAuth 配置

* bAuth:是否启用验证
* tszUserList: 用户列表地址
* tszAuthProxy:使用远程验证,通过HTTP POST来验证,参考3.2.1.2,如果为空表示使用本地列表,否则请填写一个HTTP地址

### 4.7.2 完成通知

XProxyPass 配置

* bUPGet:是否启用上传完成通知
* bDLGet:是否启用下载完成通知
* tszUPPass:用户上传完成通知地址
* tszDLPass:用户下载完成通知地址

注意:配置了这个,服务将等待你的返回结果,并且根据是否是200来返回成功和失败给客户端

## 4.8 限制配置

XLimit 配置

* nMaxUPLoad:最大上传速率.0不限制,单位字节(BYTE)
* nMaxDNLoad:最大下载速率,同上

## 4.9 版本配置

XVer 配置:用户显示当前版本号

# 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 只有LINUX下才有!

## 附录4 更新历史