XEngine Storage Service Docment

目录

[XEngine Storage Service Docment 1](#_Toc8510)

[Preface 4](#_Toc23805)

[Reader 4](#_Toc18956)

[Overview 4](#_Toc31661)

[Associate Module 4](#_Toc12106)

[一 Technical structure 4](#_Toc25124)

[1.1 Upload interface 4](#_Toc10998)

[1.2 Download interface 5](#_Toc3343)

[1.3 Mangement Interface 5](#_Toc20232)

[二 Configure Env 5](#_Toc13952)

[2.1 WINDOWS 5](#_Toc1370)

[2.1.1 Configure Environment 5](#_Toc8375)

[2.1.2 complie and run 6](#_Toc12244)

[2.2 LINUX 6](#_Toc13330)

[2.2.1 Evnironment Configure 6](#_Toc21257)

[2.2.2 Complie and Run 6](#_Toc29123)

[2.3 Version Requirements 7](#_Toc26277)

[2.3.1 System Version 7](#_Toc2303)

[2.3.2 Software Version 7](#_Toc603)

[三 Interface Protocol 7](#_Toc2746)

[3.1 query protocol 8](#_Toc4917)

[3.1.1 file list 8](#_Toc22574)

[3.2 third interface 8](#_Toc25716)

[3.2.1 user auth 8](#_Toc2754)

[3.2.2 upload proxy 9](#_Toc24675)

[3.2.3 complete notify 9](#_Toc18449)

[四 Configure Description 9](#_Toc20678)

[4.1 Serice Configure 9](#_Toc19220)

[4.1.1 basic configure 9](#_Toc25557)

[4.1.2 Max Configure 10](#_Toc29635)

[4.1.3 Time Configure 10](#_Toc8663)

[4.1.4 Log Configure 10](#_Toc30894)

[4.1.5 Database Configure 10](#_Toc8417)

[4.1.6 Storage Configure 10](#_Toc29268)

[4.1.7 Proxy Configure 10](#_Toc28485)

[4.1.8 Limit Configure 11](#_Toc30279)

[4.1.9 Version Configure 11](#_Toc12974)

[4.2 LoadBalance Configure 11](#_Toc26721)

[4.2.1 Basic Configure 11](#_Toc10503)

[4.2.2 Loadbalance Configure 11](#_Toc6792)

[4.2.3 Load Attributes 12](#_Toc24982)

[五 Advanced configuration 12](#_Toc22503)

[5.1 Distributed 12](#_Toc30709)

[5.1.1 Network Distributed 12](#_Toc7801)

[5.1.2 Storage Distributed 13](#_Toc10191)

[appendix 13](#_Toc8792)

[Appendix 1 Type Define 13](#_Toc1931)

[Appendix 2 Protocol Define 13](#_Toc20538)

[Appendix 3 Transformation Definition 13](#_Toc26955)

[Appendix 4 update log 13](#_Toc28911)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Storage Service Docment | |
| Be A Version： | V1.5 | |
| Released： | 2021-07-16 | |
| 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

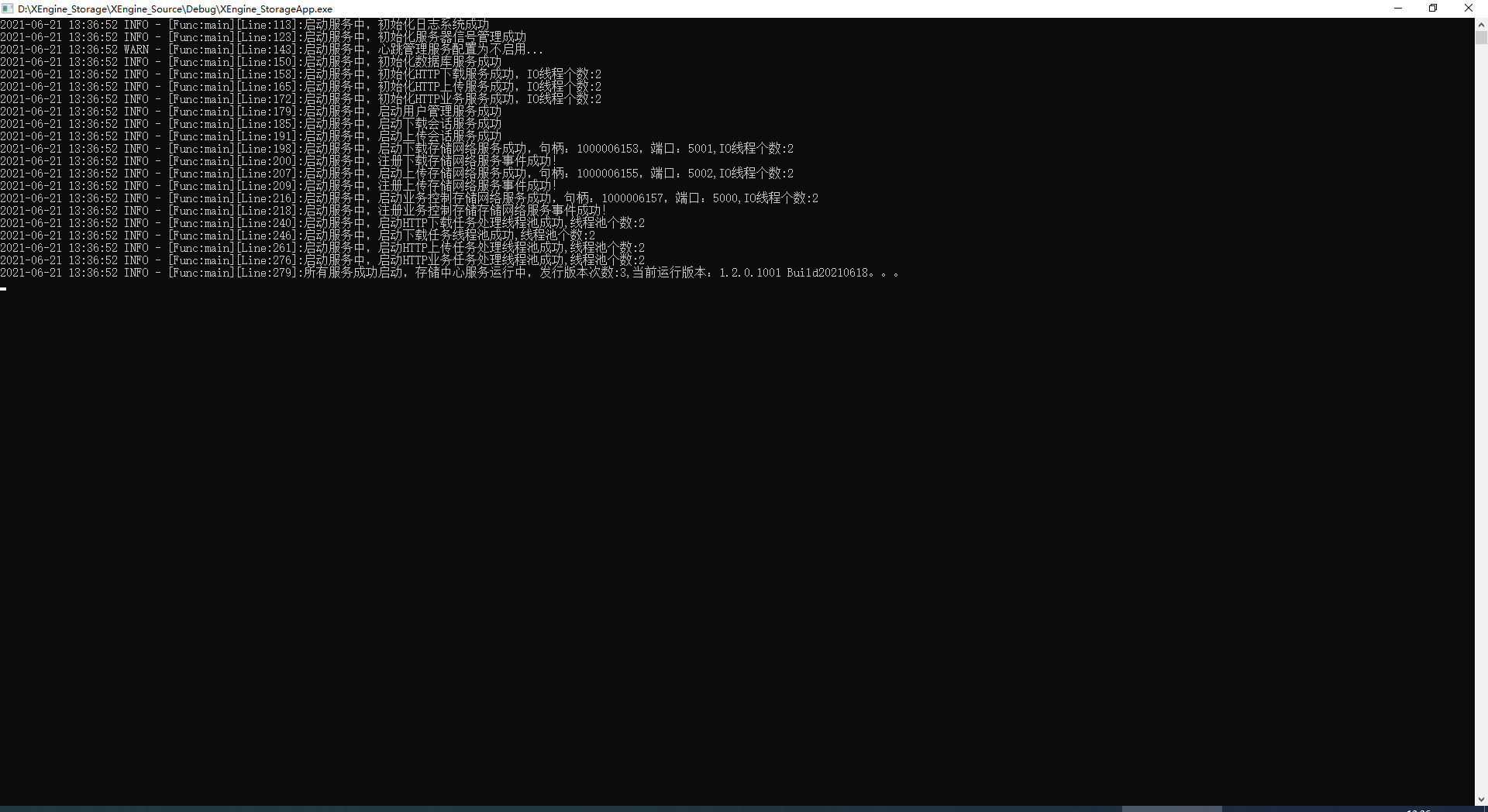
### 2.1.2 complie and run

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 Evnironment Configure

If you use linux.you must running on ubuntu(20.04) or centos(8.x)...

After download complete.you can install xengine it by the shell file.

Execute command:sudo XEngine\_RunEnv.sh -i 3

of course,and need install jsoncpp environment to your system.

Ubuntu:sudo apt install libjsoncpp-devel -y

Centos:sudo dnf install jsoncpp-devel -y

### 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\_StorageApp file in XEngine\_Release

You can running at terminal.if there is no error,you can infomation:



## 2.3 Version Requirements

### 2.3.1 System Version

Minimum version requirements:

WINDOWS: win7 sp1

Ubuntu:20.04

Centos:8.x

### 2.3.2 Software Version

Minimum version requirements:

XEngine:V7.14

JsonCpp:V1.9.2

# 三 Interface Protocol

Our interface are all post protocol.and post port.

## 3.1 query protocol

### 3.1.1 file list

Query interface:/api/query/file

Payload:empty or follow below:

{  
    **"lpszTimeStart"**:**"Start Time,can be NULL"**,  
    **"lpszTimeEnd"**:**"end time,Can be NULL"**,  
    **"lpszFileName"**:**"file name,can be NULL"**,  
    **"lpszFileHash"**:**"file HASH,can be NULL"**  
}

## 3.2 third interface

Third-interface is used to access the third-party server.now supported nginx is nginx upload module and nginx download proxy

### 3.2.1 user auth

User auth just support the http of basic now. You can configure implement user auth through userlist.txt file of mine.and use implement auth by http pass proxy.

#### 3.2.1.1 local auth

Use userlist.txt implement in XEngine\_Config

Each row represents a user.The middle is separated by a space, the front indicates the user name, and the back indicates the password.you only need configure this file to implement http basic.

#### 3.2.1.2 proxy auth

Proxy auth is send http post message implement through service,return 200 is success.other is failure.

Send http post address sent is specified by user.payload is json format.The content is as follows:

{  
    **"lpszPostUrl"**:**"client post url"**,  
    **"lpszClientAddr"**:**"client is address"**,  
    **"lpszUser"**:**"user"**,  
    **"lpszPass"**:**"pass"**  
}

If sucess.return 200..

### 3.2.2 upload proxy

Nginx upload file implement by nginx upload module.after your install and configure complete.you need configure proxy\_pass <http://192.168.1.7:5000/Api/Event/UPFile;> and Point to our service.the service can be accepted request and return .

### 3.2.3 complete notify

Completion notification means that the server has received an upload and download request after process complete.Whether it is necessary to send an HTTP POST protocol notification to the specified service. It can be configured through the configuration file

Upload and download payload of Completion notifycation is same,follow payload below send to your service..

{  
    **"lpszFileName"**:**"file name"**,  
    **"lpszFileHash"**:**"file HASH,maybe NULL"**,  
    **"lpszClientAddr"**:**"client ip address"**,  
    **"nFileSize"**:**33333**  
}

# 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
* nCenterPort:Business processing port,HTTP POST PORT
* nStorageDLPort:download port,get protocol
* nStorageUPPort:upload port,put protocol

### 4.1.2 Max Configure

XMax Configure

* MaxClient Allow Max Client Count
* MaxQueue Allow Max Queue
* IOThread:network io process threads number
* CenterThread:business handle threads number
* nStorageUPThread:upload handle threads number
* nStorageDLThread:download handle threads number

### 4.1.3 Time Configure

XTime Configure

* bHBTime,1 is enable heartbeat,0 disable
* nDBMonth:database save time,default month
* nTimeCheck:check time
* nStorageTimeOut:how time check once

### 4.1.4 Log Configure

XLog Configure

* MaxSize:Log file size
* MaxCount:Log File Number
* LogLeave:Allow save level

### 4.1.5 Database Configure

XSql Configure

### 4.1.6 Storage Configure

XStorage Configure

* nHashMode:HASH algorithm,1 MD5,2 HASH1,you can see openssl define
* bRename:Whether to automatically change the name and path, valid for files uploaded by NGINX
* tszFileDir:Save Dir

### 4.1.7 Proxy Configure

XProxy Configure

#### 4.1.7.1 Auth Proxy

XProxyAuth Configure

* bAuth:Whether enable auth
* tszUserList: user list address
* tszAuthProxy:use remote auth,auth for http.please read 3.2.1.2,if empty.use local list file.

#### 4.1.7.2 Complete Notify

XProxyPass Configure

* bUPGet:Whether enable upload complete notify
* bDLGet:whether enable download complete notify
* tszUPPass:upload complete notify address
* tszDLPass:download complete notify address

Note: When this is configured, the service will wait for your return result, and return success and failure to the client according to whether it is 200

### 4.1.8 Limit Configure

XLimit Configure

* nMaxUPLoad:Max upload speed..0 unlimited.,unit:(BYTE)
* nMaxDNLoad:Max download speed...

### 4.1.9 Version Configure

XVer Configure:Show version for user

## 4.2 LoadBalance Configure

Configure File:XEngine\_LBConfig.json

### 4.2.1 Basic Configure

* tszIPAddr:Service IP Address
* bDistributed:Whether enable distributed,if zero,not enable

### 4.2.2 Loadbalance Configure

LBConfig

* nServerMode:Distributed Service Mode,1 Random Selection

### 4.2.3 Load Attributes

Make sure content of Load balance by nUseMode

The following content is achieved through redirection,client must support 302 redirect request

* nUseMode:0,by self,1 upload.2 download,3 center
* CenterAddr:task processing load address pool
* DownloadAddr:download address pool
* UPLoaderAddr:upload address pool

# 五 Advanced configuration

## 5.1 Distributed

### 5.1.1 Network Distributed

Distributed services are implemented through HTTP redirection, and the client must support 302 to implement distributed protocols.

Processing distributed through configuration.You can configure one or more distributed processing logic through arrays

When use to distributed.distributed configure file must be close at final

#### 5.1.1.1 Configure Mulit

Distributed Configure:LoadBalance

You can add multiple address to other server with arrays.at

at the time,the server is a distributed server.it is not process your ,It does not process the nUseMode business configuration you specify, but hands it over to multiple backends for processing.

Then the back-end processing needs to set nUseMode to close.

### 5.1.2 Storage Distributed

# 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