XEngine Storage Service Docment

Index

[XEngine Storage Service Docment 1](#_Toc16917)

[Preface 4](#_Toc10927)

[Reader 4](#_Toc17683)

[Overview 4](#_Toc32767)

[Associate Module 4](#_Toc23774)

[一 Technical structure 4](#_Toc22926)

[1.1 Upload interface 4](#_Toc13578)

[1.2 Download interface 5](#_Toc19361)

[1.3 Mangement Interface 5](#_Toc29604)

[1.4 WEBDAV Interface 5](#_Toc17265)

[二 Configure Env 5](#_Toc6909)

[2.1 WINDOWS 5](#_Toc9268)

[2.2 LINUX 5](#_Toc27373)

[2.3 MacOS 6](#_Toc31194)

[三 Interface Protocol 6](#_Toc17717)

[3.1 Management protocol 6](#_Toc22123)

[3.1.1 file list 6](#_Toc30181)

[3.1.2 Insert File 7](#_Toc22353)

[3.1.3 Delete File 8](#_Toc2030)

[3.1.4 dir protocol 8](#_Toc31269)

[3.1.5 TaskList Query Protocol 9](#_Toc20297)

[3.2 third interface 10](#_Toc25781)

[3.2.1 user auth 11](#_Toc23899)

[3.2.2 complete notify 11](#_Toc8792)

[3.3 UP and Down 12](#_Toc31107)

[3.3.1 UPLoad 12](#_Toc10448)

[3.3.2 Download 12](#_Toc18155)

[3.4 Record Action 12](#_Toc18759)

[3.4.1 Download 12](#_Toc18967)

[3.4.2 Upload 13](#_Toc5677)

[四 Configure Description 13](#_Toc21451)

[4.1 Service Configure 13](#_Toc19708)

[4.1.1 basic configure 13](#_Toc17469)

[4.1.2 Max Configure 13](#_Toc2862)

[4.1.3 Time Configure 14](#_Toc28844)

[4.1.4 Log Configure 14](#_Toc4082)

[4.1.5 Database Configure 14](#_Toc1743)

[4.1.6 Storage Configure 14](#_Toc6233)

[4.1.7 Auth Configure 14](#_Toc25798)

[4.1.8 Proxy Configure 14](#_Toc4626)

[4.1.9 Limit Configure 15](#_Toc20838)

[4.1.10 P2XP Configure 15](#_Toc243)

[4.1.11 Cert Configure 15](#_Toc8221)

[4.1.12 Information Report Configure 15](#_Toc5600)

[4.2 LoadBalance Configure 16](#_Toc25793)

[4.2.1 Basic Configure 16](#_Toc10704)

[4.2.2 Loadbalance Configure 16](#_Toc16131)

[4.2.3 Local Loads 16](#_Toc9719)

[4.2.4 Load Attributes 16](#_Toc16998)

[五 Advanced configuration 17](#_Toc27241)

[5.1 Distributed 17](#_Toc26334)

[5.1.1 Network Distributed 17](#_Toc3763)

[5.1.2 Storage Distributed 17](#_Toc3979)

[5.2 HTTPS 17](#_Toc29743)

[5.3 Information Collection 18](#_Toc24619)

[FAQ 18](#_Toc8237)

[appendix 18](#_Toc7990)

[Appendix 1 Type Define 18](#_Toc17439)

[Appendix 2 Protocol Define 18](#_Toc4547)

[Appendix 3 Transformation Definition 18](#_Toc19564)

[Appendix 4 update log 18](#_Toc26532)

|  |  |  |  |
| --- | --- | --- | --- |
| File Status：  [ ] Draft  [√] Release | File Name： | XEngine Storage Service Docment | |
| Be A Version： | V3.16.0.1001 | |
| Released： | 2024-08-13 | |
| Writer： qyt | | |

# Preface

## Reader

Development ,test

## **Overview**

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

## Associate Module

This service used XEngine as the development package. To use this code, the XEngine development environment must be configured and installed.

# 一 Technical structure

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

P2P uses broadcast search, so it needs in the LAN

Should be bind four port when Start service Distinguish upload, download,webdav and management

cannot be used across ports

## Upload interface

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

The upload interface is not used form-data field.

The upload path need to create by user.

Such as:POST /api?filename=3.txt&storeagekey=storagekey1,storagekey can be Ignore,filename,if url have chinese etc...need to url encodec,url is utf8 codec.

http://127.0.0.1:5102/api?filename=storage/中文.txt&storeagekey=storagekey1

http://127.0.0.1:5102/api?filename=storage/%E4%B8%AD%E6%96%87.txt&storeagekey=storagekey1

## Download interface

Download interface is implement through get.

Get Download file through url,like this :GET /bucketkey/dir/name HTTP/1.1

BucketKey is a unique key value encoded by each storage path of distributed storage.

## 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.

## WEBDAV Interface

The WEBDAV protocol is a file management protocol based on HTTP/1.1, and it is currently supported as a file management protocol on many platforms. In version 3.16, we have also begun to preliminarily support the WEBDAV protocol.

We strictly adhere to the standards in our implementation of the WEBDAV protocol. Currently, the supported WEBDAV methods include: OPTIONS, POST, GET, PUT, PROPFIND, PROPPATCH, DELETE, LOCK, and UNLOCK.

# 二 Configure Env

## 2.1 WINDOWS

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

If environment not have error.complie is succesed.contrain 9 module and 1 exe program

And you need copy file under XEngine\_Release to your complied dir

Note: you can copy VSCopy.bat to your program dir.This script will automatically copy dependent modules file

## 2.2 LINUX

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.

## 2.3 MacOS

MacOS requires 13 or above versions, and the compilation and operation mode can refer to linux

# 三 Interface Protocol

Our interface are all HTTP protocol.

## 3.1 Management protocol

### 3.1.1 file list

#### 3.1.1.1 Requestion

Request interface:/Api/Manage/File

Request Method:POST

Request content:empty or follow below:

nMode:0 reply directly,1 reyly p2p

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

**"nMode"**:0  
}

#### 3.1.1.2 Reply

{  
    **"Code"**:**0**,  
    **"Count"**:**1**,  
    **"List"**:[  
        {  
            **"nFileSize"**:**23897183**,

**"tszBuckKey"**:**"Bucket Name"**,  
            **"tszFileHash"**:**"85E62C9D28FCE775A68DD126E0519F25"**,  
            **"tszFileName"**:**"1.docx"**,  
            **"tszFilePath"**:**"./XEngine\_File"**,  
            **"tszFileTime"**:**"2021-07-12 05:59:17"**,  
            **"tszFileUser"**:**""**  
        }  
    ],  
    **"Msg"**:**"ok"**,  
    **"lpszTimeEnd"**:**""**,  
    **"lpszTimeStart"**:**""**  
}

### 3.1.2 Insert File

If want p2p network fid the file,you must add the file to the storage service.

Base on P2P FILE Transfer need to record in the database.file download,File download needs to be implemented by the user,so when you download complete a file.you wan the file add p2p network,Then you need to submit a record to the local storage service(Please note: It is not submitted to the server, but the local storage network server).

#### 3.1.2.1 Request

with tszFileName and tszFilePath field Cant be empty,it is file path.other can be empty

Request interface:/Api/Manage/Insert

Request Method:POST

Request content:Only one of bucket and filepath can exist

{  
    **"Count"**:**1**,  
    **"List"**:[  
        {  
            **"nFileSize"**:**23897183**,

**"tszBuckKey"**:**"Bucket Name"**,  
            **"tszFileHash"**:**"85E62C9D28FCE775A68DD126E0519F25"**,  
            **"tszFileName"**:**"1.docx"**,  
            **"tszFilePath"**:**"./XEngine\_File"**,  
            **"tszFileTime"**:**"2021-07-12 05:59:17"**,  
            **"tszFileUser"**:**""**  
        }  
    ]  
}

#### 3.1.2.2 Reply

HTTP CODE 200 = Success

### 3.1.3 Delete File

Delete Request can be delete file,you can submit a delete request to service

#### 3.1.3.1 Request

Can be use file address or just use hash

Request interface:/Api/Manage/Delete

Request Method:POST

Request content:Only one of bucket and filepath can exist

{  
    **"Count"**:**1**,  
    **"List"**:[  
        {  
            **"tszFileHash"**:**"85E62C9D28FCE775A68DD126E0519F25"**,  
            **"tszFileName"**:**"1.docx"**,

**"tszBuckKey"**:**"Bucket Key Name",**  
            **"tszFilePath"**:**"./XEngine\_File"**  
        }  
    ]  
}

#### 3.1.3.2 Reply

HTTP CODE

### 3.1.4 dir protocol

The protocol can be provide folder query, create, delete operations

#### 3.1.4.1 Requestion

Request Interface:/Api/Manage/Diry

Request Method:POST

Request content:

{  
    **"nOPerator"**:**0**,

**"lpszUserDir"**:**"Operator sub dir",**  
    **"lpszBuckKey"**:**"bucket name"**  
}

* 0:Query
* 1:Create
* 2:Delete

#### 3.1.4.2 Reply

Create and delete the standard HTTP message of the reply, query the following content of the reply

{  
    **"Code"**:**0**,  
    **"Count"**:**3**,  
    **"List"**:[  
        {  
            **"tszFilePath"**:**"./XEngine\_File/Scene/scene1"**  
        },  
        {  
            **"tszFilePath"**:**"./XEngine\_File/Scene/scene1/tmp"**  
        },  
        {  
            **"tszFilePath"**:**"./XEngine\_File/Scene/scene2"**  
        }  
    ],  
    **"Msg"**:**"ok"**  
}

### 3.1.5 TaskList Query Protocol

The task query protocol can query all task with uploaded and downloaded by the current storage service, which is convenient for users to make management interfaces.

#### 3.1.5.1 Request

Request Interface:/Api/Manage/Task

Request Method:POST

Request content:nothing

#### 3.1.5.2 Reply

* ullCount:total File size
* ullRWCount:The total size that the client needs to read and write
* ullRWLen:now read and write size
* ullPosStart:The start offset required by the client
* ullPosEnd:The tail offset required by the client

{  
    **"Code"**:**0**,  
    **"Msg"**:**"ok"**,  
    **"DLCount"**:**1**,  
    **"UPCount"**:**1**,  
    **"DLList"**:[  
        {  
            **"tszClientAddr"**:**"client ip address"**,  
            **"tszFilePath"**:**"file path"**,  
            **"ullCount"**:**100**,  
            **"ullRWCount"**:**90**,  
            **"ullRWLen"**:**50**,  
            **"ullPosStart"**:**10**,  
            **"ullPosEnd"**:**100**  
        }  
    ],  
    **"UPList"**:[  
        {  
            **"tszClientAddr"**:**"client ip address"**,  
            **"tszFilePath"**:**"file path"**,  
            **"ullCount"**:**100**,  
            **"ullRWCount"**:**90**,  
            **"ullRWLen"**:**50**,  
            **"ullPosStart"**:**10**,  
            **"ullPosEnd"**:**100**  
        }  
    ]  
}

## 3.2 third interface

Third-interface is used to access the third-party server

### 3.2.1 user auth

User auth just support the http of basic now. Just support use implement auth by http pass proxy.

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:

{

"lpszMethod":"use method",

"lpszPostUrl":"client post url",

"lpszClientAddr":"client ip address",

"lpszUser":"username",

"lpszPass":"password"

}

If sucess.return 200...and payload limite data,nLimitSpeed 0 mean is unlimited

{

"code":0,

"nLimitSpeed":1024000

}

### 3.2.2 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.

bDown:1 mean is download,0 mean is upload

{

**"bDown"**:1,

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

## 3.3 UP and Down

### 3.3.1 UPLoad

Upload can be used post method,upload interface need to bucket name by system configure.upload is interface must contain Content-Length field,if you want use resume trans,then need range field.

System return 200 after upload is complete,and return 3.2.2 json payload

### 3.3.2 Download

Download interface use get method.download resume trans is supported

System will be add filehash filed in http header after Request download,it mean is this file is hash code.

## 3.4 Record Action

Transcription action can support the user to send a specified message to the server , so that the server to upload local files to a specified server or download a specified file to the server .

Support for various protocols, currently supports HTTP, FTP upload and download.

### 3.4.1 Download

API:/Api/Action/download

Method:POST

Payload:

{

"byType": 0,

"tszBucketStr": "storagekey1",

"tszFileName": "1.exe",

"tszFileUrl": "https://sw.pcmgr.qq.com/ff516c11bd16011bf946a4c31bbe0a6f/663dbf6d/spcmgr/download/QQ9.7.23.240423.exe"

}

### 3.4.2 Upload

API:/Api/Action/upload

Method:POST

Payload:

{

"byType": 0,

"tszBucketStr": "storagekey1",

"tszFileName": "1.exe",

"tszFileUrl": "ftp://10.0.1.12/QQ9.7.23.240423.exe"

}

# 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
* nWebdavPort:WEBDAV protocol port

### 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
* nWebdavThread:WEBDAV 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
* nWebdavTimeOut:webdav timeout

### 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,The server database uses MYSQL.

* SQLType:database enable type.0 not use.1 mysql,2 sqlite

### 4.1.6 Storage Configure

XStorage Configure

* nHashMode:HASH algorithm,1 MD5,2 HASH1,you can see openssl define
* bResumable:send mode,1:Use global speed limit mode,2:Use full speed mode
* bUPHash:Whether to force the uploaded file to bring the file hash field

### 4.1.7 Auth Configure

XAuth 配置

* bUPAuth:Whether enable upload auth
* bDLAuth:Whether enable Download auth
* bCHAuth:Whether enable Management auth
* tszUserList: user list address

### 4.1.8 Proxy Configure

XProxy Configure

* bUPPass:Whether enable upload complete notify
* bDLPass:whether enable download complete notify
* bAuthPass:whether enable Auth Verficaiton Proxy pass
* tszUPPass:upload complete notify address
* tszDLPass:download complete notify address
* tszAuthPass:use remote auth,auth for http.please read 3.2.1.2

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

### 4.1.9 Limit Configure

XLimit Configure

* bLimitMode:enable to speed limit
* nMaxUPLoad:Max upload speed..0 unlimited.,unit:(BYTE)
* nMaxDNLoad:Max download speed...
* nMaxUPConnect:upload of ip connect limit number
* nMaxDNConnect:download of ip connect limit number

### 4.1.10 P2XP Configure

P2XP Configure:XP2xp

* bEnable:enable or disable
* nTime:max process network time,unit second.
* nSDPort:send port,use broadcast
* nRVPort:recv port,use broadcast

### 4.1.11 Cert Configure

Cert Configure:XCert

* bDLEnable:true Enable Download SSL
* bUPEnable:true Enable UPLoad SSL
* bCHEnable:true Enable Center SSL
* bWDEnable:true Enable WEBDAV SSL
* tszCertChain:cert chain file
* tszCertServer:server cert file
* tszCertKey:cert key

### 4.1.12 Information Report Configure

Info Report Configure:XReport

* bEnable:ture enable report information
* tszAPIUrl:report address
* tszServiceName:serivce name

## 4.2 LoadBalance Configure

Configure File:XEngine\_LBConfig.json

### 4.2.1 Basic Configure

null

### 4.2.2 Loadbalance Configure

LBDistributed,configuration for redirection

* nUPLoadMode:upload distributed service,0 no enable,1 random select,2 order select,3 reverse select
* nDownldMode:download distributed service
* nStorageMode:storage distributed service

### 4.2.3 Local Loads

LBLocation Configure

* nUPLoadMode:storage key selection ,1 random select,2 order select,3 reverse select
* nDownldMode:not used
* nStorageMode:not used

### 4.2.4 Load Attributes

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

* DownloadAddr:download address pool
* UPLoaderAddr:upload address pool
* StorageAddr:distributed storage

bEnable:Whether to enable

nLevel:storage priority level, the smaller the higher the priority

Size:Allowed size, KB, MB, GB

XEngine\_Key:Bucket key,used to hide the path. Corresponding path

XEngine\_Path:Saved path

CreateDir:whether limit upload file exist dir

Rewrite:whether allow upload file rewrite

UPLimit:whether limit size and switch on upload mode 4

UPReady:whether preview create file

# 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

Storage distributed requires deploying this storage server on your client machine and enabling the SQLITE database.

Now you can query and download files through the lan network. You can request the local storage service first, and the storage service will automatically broadcast and query the files in the lan network. If it exists, it will return you the content of section 3.1.1.2, and will add a field tszTableName to indicate Server IP and port. It will return a list of all queried available local area network nodes, you can choose one download or connect multiple through the HTTP Range field to do distributed block download

## 5.2 HTTPS

After version 3.1, the service supports HTTPS access. To use HTTPS, you need to have a certificate. You can create a certificate by yourself or apply for a free one or buy one.

The certificate use is a certificate chain certificate and a certificate key (like NGINX), without verifying the client certificate. If you have advanced requirements, you can modify the code to use other certificate types or require the client to have a certificate.

We have provided a default certificate for users. Now you can directly use the certificate under XEngine\_Cert, you just need to enable SSL. This certificate is self-generated, so it's not trusted. Alternatively, you can also purchase a certificate.

## 5.3 Information Collection

Information gathering is our way to collect data on who is using our open-source software. You can control the collection of information yourselves. If you do not wish for your information to be collected by us, you can simply turn it off. You can view all the gathered information content.

# FAQ

1. We are not recommend using upload speed limit. It is best to set it to 0.

# 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