XEngine网络存储服务文档

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)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布 | 文件名称： | XEngine网络存储服务文档 | |
| 适用版本： | V1.0 | |
| 发布日期： | 2021-06-16 | |
| 拟 制： qyt | | |

# 前言

## 阅读者

开发人员,测试人员

## **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.

此服务还使用了第三方模块jsoncpp

# 一 技术结构

采用的是HTTP协议来作为基础通信协议.上传,下载,管理接口都是HTTP

服务器启动会绑定三个端口,来作为区分上传,下载,管理.

三个端口不可以跨端口使用,每个端口做的事情是固定的

支持第三方服务接口,支持NGINX作为上传和下载引擎.

## 上传接口

上传接口采用HTTP实现,上传接口需要使用HTTP的PUT方法实现.

上传接口不需要使用form-data头字段.

上传的路径需要由用户通过接口创建或者指定.

上传的路径为PUT /dir/name HTTP/1.1 取中间的URL地址

## 下载接口

下载接口通过GET实现.可以通过查询得到NGINX的下载地址

下载的文件通过 GET /dir/name HTTP/1.1 中间的URL地址获得

## 管理接口

管理接口通过POST实现,POST可以为空的负载信息,可以带JSON作为负载信息.根据每个接口的不同,负载的内容也不同,但是都是通过JSON来实现的

# 二 配置环境

## 2.1 WINDOWS

需要下载XEngine.

通过VS2019来编写代码,编译,调试.

下载地址:https://gitee.com/xengine/libxengine

### 2.1.1 配置环境

下载XEngine后,如果你是压缩包下载,需要解压,解压后,添加用户环境变量

你需要在你的系统当中,添加下面两个用户环境变量

* XEngine\_Include  XEngine的头文件目录地址
* XEngine\_Library  XEngine的库目录文件地址

比如:



你还需要JSONCPP环境,你可以通过VCPKG来安装JSONCPP.

也可以自己下载:https://github.com/open-source-parsers/jsoncpp/

如果是自己下载,你需要自己安装和编译.并且在我们的项目中-属性-VC++目录 配置你的JSONCPP目录才可使用.

# 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 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.

# 四 配置说明

# 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

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