**客户端与服务器通信协议（草案）Ver 0.1**

# 〇、说明

* 1. 服务器和客户端采用长链接TCP socket进行通信；
  2. 客户端与服务器建立连接后，会定期（3秒一次）进行心跳检查，检查失败时（3次未响应）会主动切断连接，并尝试重连服务器；
  3. 服务器或客户端收到协议定义外的消息（BUG）时主动切断连接，客户端会重连服务器；
  4. 客户端之间通过服务器中转进行通信；
  5. 客户端发送的密码均采用RSA公钥加密并做Base64转换后发送。

# 数据包定义

数据包为包头16Bytes + 可选的数据部分。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Magic | Resv1 | DataLen | MsgType | Resv2 | Data |

Magic:【3Bytes】包开始标识，固定为0x5A,0x48,0x48

Resv1:【1Byte】预留

DataLen:【4Bytes】可选数据区长度，无符号整型，网络字节序传输

MsgType:【2Bytes】本数据包消息类型，无符号整型，网络字节序传输

Resv2:【6Bytes】预留

Data:可选数据区，长度任意

C语言描述的包头结构如下：

struct ST\_DATA\_HEAD{

unsigned char magic[3];

unsigned char resv1;

long dataLen;

unsigned short msgType;

unsigned char resv2[6];

}

# 消息类型定义

消息类型取值和说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 客户端->服务器 | | | |
| 序号 | MsgType取值 | 说明 | 备注 |
| 1 | 0x00 00 | 账户注册请求 | register\_request.json |
| 2 | 0x00 01 | 账户登录请求 | login\_request.json |
| 3 | 0x00 02 | 账户信息修改请求 | update\_register\_info\_request.json |
| 4 | 0x00 10 | 客户端发送消息请求 | message\_request.json |
| ~~5~~ | ~~0x00 11~~ | ~~客户端接收消息应答~~ | ~~数据区协议待定~~ |
| ~~6~~ | ~~0x00 20~~ | ~~查找用户请求~~ | ~~数据区协议待定~~ |
| 7 | 0x00 21 | 添加好友请求 | add\_friend\_request.json |
| 8 | 0x00 22 | 好友列表更新请求 | update\_friend\_list\_request.json |
| 9 | 0x00 23 | 好友信息更新请求 | update\_friend\_request.json |
| 10 | 0x00 24 | 删除好友请求 | del\_friend\_request.json |
| … | … |  |  |
| N | 0x00 F0 | 心跳请求 | heartbeat\_response.json |
| 服务器->客户端 | | | |
| 序号 | MsgType | 说明 | 备注 |
| 1 | 0x10 00 | 账户注册应答 | register\_response.json |
| 2 | 0x10 01 | 账户登录应答 | login\_response.json |
| 3 | 0x10 02 | 账户信息修改应答 | update\_register\_info\_response.json |
| 4 | 0x10 10 | 客户端发送消息应答 | message\_response.json |
| ~~5~~ | ~~0x10 11~~ | ~~服务器发送消息通知~~ | ~~数据区协议待定~~ |
| ~~6~~ | ~~0x10 20~~ | ~~查找用户应答~~ | ~~数据区协议待定~~ |
| 7 | 0x10 21 | 添加好友应答 | add\_friend\_response.json |
| 8 | 0x10 22 | 好友列表更新应答 | update\_friend\_list\_response.json |
| 9 | 0x10 23 | 好友信息更新应答 | update\_friend\_response.json |
| 10 | 0x10 24 | 好友添加推送 | update\_friend\_server\_push.json |
| 11 | 0x10 25 | 删除好友应答 | del\_friend\_response.json |
| 12 | 0x10 30 | 好友在线信息推送 | online\_state\_server\_push.json |
| 13 | 0x1031 | 好友信息更新推送 | update\_register\_info\_push.json |
| 14 | 0x1032 | 被删除好友推送 | del\_friend\_server\_push.json |
| ... | ... |  |  |
| N | 0x10 F0 | 心跳应答 | heartebeat\_request.json |

## 2.1 账户注册请求

客户端注册账户时使用。上报【邮箱号】、【加密密码】、【昵称】、【性别】的信息。

## 2.2 账户登录请求

客户端登录账户时使用。上报【本用户ID或邮箱号】、【加密密码】信息。

## 2.3 账户信息修改请求

客户端修改账户信息时使用。上报修正后的【邮箱】、【加密密码】、【昵称】信息。

## 2.4 发送消息请求

客户端发送文本、文件信息时使用。上报【本用户ID】、【通信对象ID】、【文本】。

## ~~2.5 客户端接收消息应答~~

~~客户端接收服务器转发消息时使用。上报【接收成功否】。~~

## ~~2.6 查找用户请求~~

~~客户端查找用户/群组时使用。上报过滤条件：【本用户ID】、【查找用户ID】或空。~~

## 2.7 添加好友请求

客户端添加好友时使用。上报【查找用户ID或邮箱】。

## 2.8 好友列表更新请求

客户端登录更新好友列表时使用。

## 2.9 好友信息更新请求

客户端更新特定好友信息时使用。上报【好友ID】。

## 2.10 心跳请求

客户端定期发送心跳，并接收服务器返回的心跳应答。

## 2.11 账户注册应答

服务器收到客户端【账户注册请求】时使用。下发【成功/失败】、【用户ID】。

## 2.12 账户登录应答

服务器收到客户端【账户登录请求】时使用。下发【成功/失败】。

## 2.13 账户信息修改应答

服务器收到客户端【账户信息修改请求】时使用。下发【成功/失败】。

## 2.14 客户端发送消息应答

服务器收到客户端【发送消息请求】时使用。下发【成功/失败】。

## ~~2.15 服务器发送消息通知~~

~~服务器收到客户端【发送消息请求】时转发消息使用。下发【发送用户ID】、【加密文本】、【加密文件】。~~

## ~~2.16 查找用户应答~~

~~服务器接收到【查找用户请求】时使用。下发过滤后的【用户ID】、【用户昵称】、【群组名】。~~

## 2.15 添加好友应答

服务器接收到客户端【添加好友请求】时使用。下发【成功/失败】。

## 2.16 好友列表更新应答

服务器接收到客户端【好友列表更新请求】时使用。下发【用户ID】、【邮箱】、【昵称】、【性别】。

## 2.17 好友信息更新应答

服务器接收到客户端【好友信息更新请求】时使用。下发【用户ID】、【邮箱】、【昵称】、【性别】。

## 2.18 好友添加推送

服务器接收到客户端【好友添加请求】时使用。推送【用户ID】、【邮箱】、【昵称】、【性别】。

## 2.19 好友添加推送

服务器定期推送【好友在线/好友离线】。

## 2.20 心跳应答

服务器接收到客户端【心跳请求】时使用。