

# Goim 客户端通讯协议文档

comet支持两种协议和客户端通讯 websocket, tcp。

Git 地址： <https://github.com/Terry-Mao/goim>

## websocket

略

## tcp

请求URL

tcp://DOMAIN

协议格式

二进制，请求和返回协议一致

请求&返回参数

参数名	必选	类型	说明
package length	TRUE	int32 bigendian	包长度
header Length	TRUE	int16 bigendian	包头长度
ver	TRUE	int16 bigendian	协议版本
operation	TRUE	int32 bigendian	协议指令
seq	TRUE	int32 bigendian	序列号
body	FALS	binary	\$(package lenth) - \$(header length)

## 指令

指令	说明
2	客户端发起心跳(服务器判断是否在线, 150秒超时将断掉连接)
3	服务端心跳答复
4	私信消息上行(request)
5	私信消息回复(request reply)
6	服务端通知客户端断掉连接(disconnect)
7	auth认证
8	auth认证返回
15	客户端上线拉取未读消息(pull)
16	ACK , 客户端通知服务器已收到该消息, 消息体待定
17	服务器私信消息下行通知客户端

Package Length	4 bytes
Header Length	2 bytes
Protocol Version	2 bytes
Operation	4 bytes
Sequence Id	4 bytes
Body	PackLen - HeaderLen

## 消息体格式:

私信消息体:

```
{
    "Id": 99999,          /* 可选, 发送消息时不带, 由服务器生成, 给接收消息的一方做去重 */
    "From_id": 25933, /* 发送消息uid , 必选 */
    "To_id": 25934,    /* 接收消息uid , 必选 */
    "Action": "user", /* 消息 action , user 为用户私信, 必选 */
    "Type": "text",    /* 消息type, text为文本, 未来兼容音频, 图片, 必选。暂定512字节 */
    "Content": "你好啊", /* 消息内容, 必选 */
    "Platform": 0,      /* 0 : IOS , 1 : Android , 2 : web , 必选 */
    "Version": 1        /* 消息客户端版本号, 发送端在协议里, 接收端在消息体内也会展现 */
}
```

```
}
```

拉取离线未读消息体：

```
{
  "Type": "unread", /* 拉取未读消息, 未来兼容消息漫游 */
  "StartId": 99999, /* 拉取起始消息ID */
  "Limit": 99 /* 倒序, 分页大小, 是否每次全拉待定 */
}
```

ACK 消息体：

```
{
  "Type": "ack", /* */
  "AckIds": [99997, 99998, 99999] /* ACK ID list */
}
```

## IM业务需求疑点：

1. 消息是否需要音频/图片等，甚至以后更多的类型？文本的长度限制，建议1024
2. 用户上线下线维护
3. 离线消息是否维护，维护的话是拉还是推的方式，离线消息是否走推送？还是直接抛弃？走推送的话需要推送相关做一个微服务
4. 客户端是否需要ACK
5. 消息体是否存数据库
6. AES 验证/关系链收敛到用户微服务，需要用户微服务提供API
7. 客户端需保证消息去重，可以考虑以消息体“Id”做标识
8. 是否需要支持漫游消息？需要的话就需要存储收发消息用户的timeline

9. 用户什么样的关系允许发消息？
10. 是否支持多点登陆？
11. 消息体是否加密？

游戏部兄弟的消息系统代码，应用到我们系统主要有这几个难点：

1. 最前端需要实现一层消息网关层；
2. 需要将单进程单线程模型改为协程模型，运维部署管理难度较高；
3. 消息数据积压时全在内存，一旦重启很容易丢消息，需要实现消息队列  
publish / subscribe ；
4. 需要扩展将语音，图片等大文件上传OSS，消息内部只传输URL