

IM长连接协议

IM长连接协议

- 数据传输的定义
 - 描述
 - 公共请求头（11 byte）
 - 协议连接标识
 - 状态码
 - 命令
 - 请求消息体
- 接口定义与描述
 - 长连接登陆
 - protobuf 消息体
 - 响应
 - 客户端心跳
 - protobuf 消息体
 - 响应
 - 加入聊天室
 - protobuf 消息体
 - 响应
 - 发送消息到指定聊天室
 - protobuf 消息体
 - 响应
 - 接收聊天室消息
 - 响应

数据传输的定义

描述

- 所有消息均使用统一的 固定请求头 + 消息体 进行传输；
- 业务采用命令驱动的方式，每一个消息均有一个命令；

公共请求头（11 byte）

len (4 byte) | seq (4 byte) | command (1 byte) | connect flag (1 byte) | status (1 byte)

Len（长度）：用于记录消息内容的长度，不需要计算本身长度

Seq（序列）：客户端填充的自增ID

Command（执行命令）：具体执行的业务

Connect flag（连接标识）

Status（状态码）：默认填零，服务器响应数据时，这里是错误码

协议连接标识

- 通过该字节实现协议的标识；
- 包括数据流方向，加密方式，压缩方式；

Bit	7	6	5	4	3	2	1	0
	version				Compression		Flow	

Version（版本号）

Compression（压缩方式）：0 不压缩；1 使用 gzip；2 保留；3 保留；

Flow（流方向）：0:客户端请求服务器；1:服务器响应客户端；2:服务器通知客户端；

状态码

状态码	描述
0	成功

命令

```
enum IMCommand {
    RESERVED = 0;    /* 保留字段 */
    LOGIN = 1;       /* 登录 */
    LOGOUT = 2;      /* 退出登录 */
    PING = 3;
    PONG = 4;
    MS = 5; /* 发送消息 */
    MP = 6; /* 推送消息 */
    CCR = 7; /* 创建聊天室 */
    DCR = 8; /* 删除聊天室 */
    JOIN_CR = 9; /* 加入聊天室 */
    QUIT_CR = 10; /* 退出聊天室 */
    CMS = 11; /* 发送聊天室消息 */
    CMP = 12; /* 推送聊天室消息 */
}
```

command	方向（客户端 - 服务器）	命令描述	是否有返回值
RESERVED	x	保留字段	False
LOGIN	>	发起IM长连接的业务登陆	True
LOGOUT	>	发起IM长连接的登出	False

请求消息体

- 使用protobuf进行序列化后的字节数据

接口定义与描述

长连接登陆

- 长连接建立成功后，第一个包总是登陆包，否则服务器将直接关闭连接；
- 使用 LOGIN 发起登陆；
- 如果app是登陆状态，则发起登陆用户对应的 token ,userId 进行长连接登陆认证；
- 如果app未登陆，发起游客登陆，游客登陆时不需要发送任何包内容；

protobuf 消息体

```
/* 客户端信息 */
message ClientInfo {
    string userId = 2;      /* 用户id */
    string appVersion = 3; /* app 版本 */
    Platform platform = 4; /* 平台信息 */
    Network network = 5;    /* 客户端网络信息 */
}

/* 登陆 */
message Login {
    ClientInfo clientInfo = 1; /* 客户端基础信息 */
    string token = 2;          /* app 登陆使用token */
}
```

响应

```
message LoginResponse {
    uint64 messageHead = 1; /* 用户消息版本号 */
    uint64 userSettingHead = 2; /* 用户设置版本号 */
    uint64 userFriendHead = 3; /* 好友信息版本号 */
    uint64 friendRequestHead = 4; /* 好友请求版本号 */
}
```

客户端心跳

- 每隔30S如果没有客户端请求，服务器会关闭对应连接；
- 客户端在请求空闲时，需要每间隔14秒进行一次心跳请求；
- 服务器收到请求后，响应 PONG 包；

protobuf 消息体

- 没有包体

响应

- 响应 PONG 包，没有包体

加入聊天室

- 客户端需要加入聊天室，才能收到聊天室消息
- 使用命令 JOIN_CR 发起加入聊天室连接

protobuf 消息体

```
message ChatroomRequest {  
    string chatroomId = 1; /* 聊天室ID */  
}
```

响应

- 响应 JOIN_CR 的固定请求头，没有包体

发送消息到指定聊天室

- 给聊天室发送一条消息 CMS；
- 需要客户端采用用户登陆方式，游客无法发送聊天室消息；

protobuf 消息体

```
/* 聊天室消息 */  
message CRMessage {  
    bytes messageContent = 1; /* 消息内容 */  
    string from = 2;          /* 来源用户id */  
    string to = 3;            /* 目标聊天室id */  
    uint64 timestamp = 4;     /* 时间戳 */  
    message UserInfo {  
        uint32 level = 1;      /* 等级 */  
        uint32 roomLevel = 2;  /* 聊天室等级 */  
        string nickname = 3;   /* 昵称 */  
        repeated string tag = 4; /* 用户标签 */  
    }  
}
```

```

UserInfo userInfo = 5;
uint32 messageType = 6;    /* 聊天室消息类型 */
}

enum CRMessageType {
    NORMAL = 0; /* 普通消息 */
    NOTIFY = 1; /* 通知消息  加入直播间/封禁用户 */
    GIFT = 2;   /* 礼物消息 */
}

```

响应

- 没有响应

接收聊天室消息

- 加入聊天室后，接收其他成员发送到该聊天室的消息；

响应

```

/* 聊天室消息 */
message CRMessage {
    bytes messageContent = 1; /* 消息内容 */
    string from = 2;          /* 来源用户id */
    string to = 3;            /* 目标聊天室id */
    uint64 timestamp = 4;     /* 时间戳 */
    message UserInfo {
        uint32 level = 1;     /* 等级 */
        uint32 roomLevel = 2; /* 聊天室等级 */
        string nickname = 3;  /* 昵称 */
        repeated string tag = 4; /* 用户标签 */
    }
    UserInfo userInfo = 5;
    uint32 messageType = 6;   /* 聊天室消息类型 */
}

enum CRMessageType {
    NORMAL = 0; /* 普通消息 */
    NOTIFY = 1; /* 通知消息  加入直播间/封禁用户 */
    GIFT = 2;   /* 礼物消息 */
}

message CRMessageResponse {
    repeated CRMessage messages = 1; /* 聊天室消息集合 */
    string chatroom = 2;             /* 聊天室ID */
}

```

