**消息协议**

1. 消息协议：总长度为44+length

注：websockt 只使用content那一部分

|  |  |  |
| --- | --- | --- |
| length | Int | 存放content的字节长度 |
| msgId | string | 消息id; |
| timestamp | long | 消息时间戳（ms）； |
| content | string | 消息内容json字符串，参考content |

1. content内容为json字符串

|  |  |  |
| --- | --- | --- |
| protocol | string | 消息协议类型：  Heart(1, "心跳协议"),//心跳服务端验证时间10秒，前端发要在10秒  Message(2, "消息协议"),  Response(3, "响应协议"),//待定，可能不需要了  Connect(4, "连接协议"), |
| data | Object | 每种协议对应的数据，参考后面的data |

1. Data数据模型

|  |  |  |  |
| --- | --- | --- | --- |
| ConnectData | appId | String | 应用id |
| token | string | 用户token |
| platform | string | 平台类型：  App(1, "APP"),  Web(2, "WEB"),  PC(3, "PC") |
| HeartData | status | int | 状态 |
| ResponseData | status | int | 响应状态：  200-成功  401-私聊：非好友，群聊：非群成员  403-群已禁言  500-服务内部异常 |
| msgId | string | 消息id |
| timestamp | long | 消息时间戳 |
| errorMsg | string | 失败原因 |
| MessageData | msgId | string | 消息id方便操作 |
| timestamp | Long | 消息时间戳(ms) |
| chatId | String | 会话id |
| chatType | String | 会话类型  Private(1, "私聊"),  Group(2, "普通群"),  SuperGroup(3, "超大群"); |
| fromId | String | 发送人pin |
| appId | String | 应用id |
| from | User | 发送人信息 |
| chatMsgType | String | 消息类型：  Session(1, "会话消息"),  Notice(2, "通知消息"); |
| msgFormat | String | 消息 |
| noticeType | String | 通知类型 |
| seqId | long | 消息seq |
| body | String | Notice，attact，body |

User对象

|  |  |  |  |
| --- | --- | --- | --- |
| User | pin | String | 发送人pin |
| appId | String | 应用id |
| IsBot | boolean | 是否机器人 |
| nickname | String | 发送人昵称 |
| avatar | String | 发送人头像 |

Chat 聊天对象

|  |  |  |  |
| --- | --- | --- | --- |
| Chat | id | String | 会话id |
| title | String | 会话名称 |
| type | String | 会话类型  Private(1, "私聊"),  Group(2, "普通群"),  SuperGroup(3, "超大群"); |

前端确认不需要对象；当PC端多出一个对象参数时，不会报错；

1. 会话消息类型

|  |  |
| --- | --- |
| msgFormat | TEXT("文本", false, null) |
| PICTURE("图片", false, null) |
| TIME("时间", false, null) |
| CARD("名片", false, null) |
| TRANSFER("转发", false, null) |
| VIDEO("视频", false, null) |
| AUDIO("音频", false, null) |
| LINE\_VIDEO("视频通话", false, null), |
| LINE\_AUDIO("音频通话", false, null), |
| REPLY("回复", false, null), |
| REPLY\_SPECIAL("@某人", false, null), |
| REPLY\_AT("既有回复又有@", false, null), |
| FILE("文件", false, null), |
| Emoji("表情", false, null), |
| Friend\_Agree("好友同意", true, "{0}同意了{1}的好友申请,现在可以开始聊天了") |
| Team\_Create("创建群", true, "{0}创建了群聊"), |
| Team\_Join("邀请入群", true, "{0}邀请了{1}进入了群聊"), |
| Team\_Kick("群踢人", true, "{0}将{1}移除群聊"), |
| Team\_Leave("退群", true, "{0}已退出群聊"), |
| Team\_Remove("群解散", true, "本群已解散"), |
| Team\_Mute("群禁言", true, "{0}已开启了群禁言"), |
| Team\_Mute\_Cancel("群取消禁言", true, "{0}已取消群禁言"), |
| Team\_Protect("群成员保护", true, "{0}已开启群成员保护"), |
| Team\_Protect\_Cancel("群成员保护取消", true, "{0}取消了群成员保护"), |
| Team\_Msg\_Top("群消息置顶", true, "{0}置顶了《{1}》"), |
| Team\_Manager("设置管理员", true, "{0}已将{1}设置为管理员"), |
| Team\_Manager\_Cancel("取消管理员", true, "{0}已将{1}移除管理员"); |
|  |

1. 通知消息类型

|  |  |  |
| --- | --- | --- |
|  | FRIEND\_REQUEST(3, "请求添加好友 "), |  |
| FRIEND\_AGREE(4, "同意添加好友"), |  |
| FRIEND\_DELETE(6, "删除好友"), |  |
| TEAM\_CREATE(8, "创建群"), |  |
| TEAM\_JOIN(10, "加入群聊"), |  |
| TEAM\_KICK(11, "踢出群"), |  |
| TEAM\_REMOVE(12, "解散群"), |  |
| TEAM\_UPDATE(13, "修改群资料"), |  |
| TEAM\_LEAVE(14, "退群"), |  |
| TEAM\_MUTE(106, "群全体禁言"), |  |
| TEAM\_MUTE\_CANCEL(107, "群全体取消禁言"), |  |
| TEAM\_FORBID\_FRIEND(108, "群成员禁止添加好友"), |  |
| TEAM\_FORBID\_FRIEND\_CANCEL(109, "群成员取消禁止添加好友"), |  |
| TEAM\_INVITE(110, "","普通群成员能否邀请他人入群"), |  |
| TEAM\_INVITE\_CANCEL(111, "","取消普通群成员能否邀请他人入群"), |  |
| TEAM\_MANAGER(112, "","设置管理员"), |  |
| TEAM\_MANAGER\_CANCEL(113, "","取消设置管理员"), |  |
| SESSION\_INFO\_PUSH(201, "推送会话信息"), |  |
| SESSION\_LIST\_DELETE(202, "删除会话列表"), | 私聊和群公用 一个 |
| MSG\_DELETE(205, "删除消息"), |  |
| DEL\_NOTICE(206, "","删除通知"),//前端处理完通知后，通知后端进行删除 |  |
| MSG\_READ(208, "消息已读"), |  |
| TOP\_ON(209, "群消息置顶"), |  |
| TOP\_OFF(210, "群消息取消置顶"), |  |
| USER\_UPDATE\_BASIC(700, "好友修改基本资料信息"), |  |
| CUSTOM(900, "自定义通知"); |  |