客户端回复消息方案文档

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订版本** | **修改描述** | **作者** |
| 2021-3-10 | V1.0 | 编写 | 零度C/中午 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 范围

本规范适用于回复功能的实现。

# 概念名词

| 缩略语 | 英文全名 | 说明 |
| --- | --- | --- |
| 可见权限 |  | 某个用户对某条消息的可见权限是指该消息最开始是否对该用户可见，如果可见就说明对该消息有可见权限，不可见就没有可见权限  如：某个群，a用户发送消息1的时候，b不在该群聊里，c在群里。所以b对消息1没有可见权限，c有可见权限 |
| 消息删除状态 |  | 在用户有可见权限的情况下，如果原消息在被清空、为自己删除、为所有人删除、已注销用户消息4种情况下，被回复消息显示消息已删除 |

# 目标

记录回复消息功能产品需求和客户端技术实现方案

# 需求描述

## 针对回复消息，如果用户没有原消息的可见权限，则该消息的UI展示和普通文本消息一样

## 针对回复消息，如果用户有原消息的可见权限，如果原消息没有被清空、没有被删除（包括为自己删除和为所有人删除），也不是注销用户发送的消息，则按照正常的回复消息样式展示

## 针对回复消息，如果用户有原消息的可见权限，如果原消息满足被清空、没有被删除（包括为自己删除和为所有人删除）或者是注销用户发送的消息其中一种，则直接展示原消息已被删除且去掉用户昵称

注：为自己删除，只有自己显示原消息已被删除，其他人显示正常

# 客户端技术方案描述

## 实现方案

### 消息可见权限判断

后台会话信息表中有msgFilter字段，该字段是一个json数据，记录了该用户进出群聊的所有记录，客户端同步该字段并入库，收取某条回复消息后，需要遍历msgFilter字段，检查该消息是否在可见范围内，并将该记录入库

### UI展示

UI展示某条回复消息时，需要现判断该回复消息的原消息是否有可见权限，如果没有可见权限，则按照正常的文本消息展示；如果有可见权限，则第一步判断原消息是否小于会话的minMsgID（即是否被清空），如果小于直接显示消息已被删除。如果大于minMsgID 则进行第二步判断原消息是否是已删除用户所发送消息，如果是则直接显示消息已被删除，如果不是则进行第三步判断原消息是否已被删除（原有异步查询方案，见文档结尾），如果删除则显示消息已被删除，如果没有删除则正常按回复消息逻辑展示展示

### msgFilter同步方案

1. 会话信息表新增msgFilter字段，用于持久化后台msgFilter字段信息
2. 离线转在线，同步会话列表，并将msgFilter字段信息存入数据库
3. 在线会话变更通知，需要将通知中的msgFilter字段信息更新到数据库中

### 在线通知处理方案

在线收到消息删除、撤回、msgFilter变更以及账号注销4种通知时，需要重新检测相关回复消息，并记录原消息的删除状态，刷新页面显示

注：

1. 如果用户停留在聊天页，重新进行tcp登录拉取完离线消息后需要重新刷新页面，保证回复消息原消息状态能正常展示
2. msgFilter字段示例说明：

[{"endMsgId":6735399135312412690,"beginTime":0,"endTime":1605844293308,"beginMsgId":0},{"beginTime":1605844299860,"beginMsgId":6735399135312412690}]

每个msgFilter子Item最多包括beginTime、beginMsgId、endTime、endMsgId 4个字段，表示（beginMsgId，endMsgId）区间内的消息有可见权限，该区间是左开右闭区间，如果没有endMsgId，endTime则表示该beginMsgId以后的所有消息都有可见权限

3）

```Java

{

"myselfContent":"huifu",

"replyContent":"[图片]",

"replyName":"user10",

"replyType":1

“url”:“url”

“isAit”:true,

"msgId":0

}

```

## 原消息异步查询方案

因为群聊消息没有全部拉取到本地，所以回复消息的原消息的删除通知理论上也有可能没有同步到本地，导致回复消息的原消息状态展示不正确，为了能正确展示回复消息，在回复消息展示的时候会实时异步向服务器查询回复消息的原消息状态，如果被删除，则先更新数据库数据并异步回调通知UI进行刷新

（原消息异步查询方案优化）为了减少每次展示都需要请求后台查询消息状态的次数，做了以下优化：

1）如果某条消息的原消息本地有记录且已被删除，则直接显示消息已删除，不再查询消息状态

2）每次tcp登录后记录向后台请求查询的消息ID，在一次长链接登录生命周期内，同一条消息只请求一次