

消息表
id 自增id
mid 唯一主键，分布式
sequence 服务端id,
chatId 会话id
senderId 发送方uid
receiverId 接收方uid
status 消息状态

数据库

客户端 发送方

发送

生成全局唯一
messageId

生成规则

生成chatId

生成规则

创建消息，写入相关数据，
content, senderId 等

将消息status 设置为
sending

重发

保存消息到数据库

事件响应刷新UI

发送消息到服务器

服务端响应

将消息status设置为
success

成功

失败

将消息status设置为failed

更新消息到数据库

事件响应刷新UI

服务器

客户端 接收方

数据库

对比本地消息列表中最大的sequence

小于本地sequence

不做处理

大于本地sequence

是否在处理过程中
判断isPulling

是否存在
continue

使用键值对方式存储
continue，值为
sequence

是

sequence > continue

否

结束

更新continue

通知

更改状态isPulling =
true

根据sequence获取扩散
列表

服务端响应是否成功

是

将服务端返回的扩散
表数据去重，根据
sequence进行排序

通过DB过滤(查询 in)
接收到的扩散数据

过滤后数据是否为空

是

isPulling = false

不为空，请求消息详情

响应是否成功

是

保存消息详情到数据库

isPulling = false

消息详情返回

消息保存成功后，通知
订阅更新UI