超高可用实时聊天程序

内存型数据库详细设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 内容 | 前一版本 | 当前版本 | 修改/创建人 | 时间 |
| 1 | 创建文档，从mysql详细设计移出。调整表序号。 | V0.1.8a | V0.2.0a | Alban | 2017/4/5 |
| 2 | 版本变更。此版本为开发前稳定版。 | V0.2.0a | V0.4.0b | Alban | 2017/4/5 |

**表1** 好友关系表 user\_friendship（redis存储）

HashMap<String key , List<Friend > value > friendships;

其中key为用户账号。

Value为好友的账号的list。

注：好友关系不是相互的。非好友使用id聊天。

public class Friend{

String id; //好友的ID

int group;//好友所在的组。1组为默认组“我的好友”，-1为默认组“黑名单”。0组为默认组“我的电脑/手机”，2组为默认组“陌生人”。

int friendShip;//好友的关系。1：特别关注好友；0：正常好友；2：黑名单。

int remark;//好友的备注。

}

HashMap<String key ,List<GroupInfo> group\_info>

其中key的格式为：userGroup\_+“UserID”。

其中GroupInfo类对象内容如下。

Public class GroupInfo{

Int groupId;//系统定义的组ID，不能重复，且不能为1、0、2

String groupName;//存储用户的组名

}

HashMap<String key,List<String> community\_id>;

其中key的格式为：userCommunity\_+“UserID”

//用户所包含的群表（当群表中无法找到community\_id时，从list中移除此ID）

**表2** 群成员表 community\_member（redis存储）

HashMap<String key , Hash<String member\_key，Member> value >

Key 的格式为群ID。

//Member\_key 为群成员ID。 其中Member类对象内容如下

Public Class Member{

String memberId;//成员ID

Bool admin;//是否为管理员（初始化为FALSE）

Date joinTime;//加群日期

String communityNickName;//群成员的群昵称（初始化成为用户昵称）//当用户尝试修改此值时，交由rabbitmq进行分发到Java服务端进行操作。

}