## 聊天

**注**:主要用webscoket来实现聊天及移动端推送功能(chat

模块),目前只实现了消息推送(ChatServer),AudioServer为语音聊天的服务,暂未进行与移动端的测试,聊天记录也没有彻底进行测试,可能存在bug

实现思路:

**Join(加入聊天)**

1. 应用启动时注册webscoket服务端
2. 客动端连接服务端时对客户端连上来的用户进行初始化,并保存到google cache,以供查询是否已经登录过了,如果没有连接过,则加入cache,否则更新缓存(值的注意的是因为重新连接会导致原本的session可能close,所以必须更新)
3. 读取离线时未读取的消息(未读消息保存在mongodb)
4. 读取系统推送消息(根据msg的类型来读取)
5. 上线通知,通知好友(根据msg的类型来读取)

**quit(退出聊天)**

1. 判定是刷新还是退出(此处好像没处理好)
2. 下线通知,通知好友(根据msg的类型来读取)
3. 客户端从缓存移除
4. 关闭会话

**Incoming(发送消息及接收)**

1.对消息发送的大小进行设置,(不能设置太大,否则内存飙升导致内存移除,程序)

2.解析消息(已跟移动端做好规定,什么类型的数据怎样处理)根据msg的类型

点对点发送,群发,还是上下线通知,系统通知(未处理)

**onError(错误处理)**

发生错误当作下线处理

### 文件及图片的发送思路

以发送图片为例.如移动端需要发送一张图片:

1. 会先把图片单独的进行上传(减少对服务端 的压力)
2. 服务器返回图片的url信息
3. 移动端获取显示,可通过消息的类型进行判定,如是图片类型的消息,移动端会在另外的接口获取文件的缩略图获取原图进行显示

关于消息的保存,服务端仅仅对url进行保存,而不是文件字节

**涉及的表结构:**

Mongodb:

Msg(消息表)

Mysql:

chat\_group(群),chat\_group\_member(群成员),chat\_friend\_group(朋友分组),chat\_friend(我的好友)