# Icare信息推送系统

信息包分成两类：1连接维护 2 信息message.

**public** **final** **static** **byte** ***HEART\_BEATING*** =100;

**public** **final** **static** **byte** ***TEXT\_MESSAGE*** =101;

**private** **static** **final** **byte** ***MESSAGE\_ANSWER*** = 102;

**public** **final** **static** **byte** ***IMAGE\_MESSAGE*** =103;

**public** **static** **final** **byte** ***DATA\_MESSAGE*** = 104;

**public** **static** **final** **byte** ***TEST*** = 105;

**public** **static** **final** **byte** **ANSWER**\_***TEST*** = 106;  
  
**1心跳heartbeating信息包格式：**

{type (1byte)} //包类别100  
{icare\_id (8bytes)}  
**2信息message信息包格式：**

{type (1byte)} //包类别101  
{total\_bytes(8 bytes)} //总字节数，每条信息如果大于1024会被拆分成多个包发送，该数

值代表该信息（不包括头信息）总字节数  
{from\_icare\_id(8bytes)} //发送方icare\_id  
{to\_icare\_id(8bytes)} //接收方icare\_id  
{sn(4bytes)} //该信息包在message中的序号  
**3 客户端回复服务器信息**

**4 图像信息**

**5 数据信息**

**6 连接测试信息**

客户端向服务器发出的连接测试信息，等待服务器做出回复

**6 服务器回复连接测试信息**

服务器向客户端回复的连接测试信息，当服务器接收到客户端的TEST信息后服务器向客户端回复。

## 服务端

主程序：coboo.MinaServer.java

配置文件：config.xml

概述：

系统使用了mina网络通信框架

### 主要类：

**coboo.mina.CoMessageHandler**

存储IoSession的公共变量*clients*

**static** HashMap<Long,IoSession>*clients*=**new** HashMap<Long, IoSession>();

**信息进入服务器mina调用的函数**

**public** **void** **messageReceived**(IoSession session, Object message) **throws** Exception

转发或者本地存储消息

**private** **void** forwardStoreMsgArray(IoSession session, **final** **byte**[] msgArray)

当信息接收方没有在线时候，信息会以临时文件的形式存储到磁盘上，存放文件名字为：

**/icare/message/{receiver\_icare\_id}/{message\_id}\_0**

当客户端连接服务器后会检查自己目录发现文件后收取，并且删除该目录和文件。

**private** **void** storeMessage(**byte**[] msgArray, **long** content\_total\_bytes, **long** message\_id, **long** to\_icare\_id, **int** sn,**byte**[] str)

**信息发送后mina调用**

**public** **void** messageSent(IoSession session, Object message) **throws** Exception

## 客户端

客户端对应**coboo.icare.network.MessageClient**

*/\*读线程和心跳线程\*/*Thread **initSocketThread**,**readThread** , **heartBeatingThread**;  
 */\*接收信息包缓存\*/***volatile private byte**[]**buffer**=**new byte**[Config.***MESSAGE\_BUFFER\_SIZE***];  
 */\*发送和接收数据网络接口\*/* **volatile private** Socket **socket**;

#### startMessageClient ：

是推送系统客户端运行的起始点，该函数是当网络状态变化时候网络状态接收器**NetworkStatsReceiver**调用。

客户端启动后首先进行调用connect()连接，连接成功后获得**socket，然后**

receiving();  
sendHeartBeating();

启动接收信息进程和发送心跳进程。