# 接口文档(第二次)

Git链接：<https://github.com/BrunoQin/Open-Reuse.git>

组员：1352958 金敏 1352911 曹琦 1354366 潘舜达

1352961 秦博 1352839 饶伊文

执笔人：饶伊文

**目录**

[接口文档(第二次) 1](#_Toc16487)

[Client接口 com.openresure.client 4](#_Toc29059)

[com.openresure.client 4](#_Toc32678)

[class ClientAgent API 4](#_Toc26053)

[com.openresure.client.config 4](#_Toc25348)

[class ConfigManager 4](#_Toc30946)

[com.openresure.client.handler 5](#_Toc31015)

[class IncomingMessageHandler extends ChannelInboundHandlerAdapter 5](#_Toc12970)

[com.openresure.client.listener; 5](#_Toc5977)

[interface MessageListener 5](#_Toc13585)

[com.openresure.client.queue 6](#_Toc14924)

[class MessageQueue 6](#_Toc2662)

[com.openresure.client.sender 6](#_Toc20061)

[class MessageSender implements Runnable 6](#_Toc16575)

[com.openresure.client.service 6](#_Toc20552)

[class ConnectionMgmtService 6](#_Toc31698)

[class MessageNotifyService 6](#_Toc50)

[class MessageSendingService 7](#_Toc24013)

[Common接口 com.openresure.common 7](#_Toc17849)

[com.openreuse.common.message.builder 7](#_Toc2284)

[class MessageBuilder 7](#_Toc5854)

[com.openreuse.common.message 7](#_Toc19738)

[class Message 7](#_Toc5663)

[enum MessageType 信息类型枚举 8](#_Toc5140)

[class Reserved(作为保留字为后续开发留空间) 8](#_Toc31383)

[com.openreuse.common.persist 8](#_Toc9107)

[class LocalPersistHelper 8](#_Toc23041)

[Server接口 com.openreuse.server 9](#_Toc3526)

[class ServerApplication 9](#_Toc27584)

[com.openreuse.server.handler 9](#_Toc10917)

[class RawJsonHandler extends ChannelInboundHandlerAdapter 9](#_Toc21867)

[com.openreuse.server.misc 9](#_Toc2282)

[class Constants 9](#_Toc21608)

[com.openreuse.server.misc.worker 9](#_Toc31660)

[interface Worker extends Runnable 9](#_Toc30100)

[interface WorkerPool<Textends Worker> 9](#_Toc20976)

[com.openreuse.server.pojo.client; 9](#_Toc21268)

[class AbstactClient 9](#_Toc28591)

[com.openreuse.server.pojo.notify 10](#_Toc20926)

[class DelayedNotify implements Delayed 10](#_Toc31775)

[com.openreuse.server.registry 10](#_Toc25278)

[class RegistryManager 10](#_Toc14154)

[com.openreuse.server.registry.validateRegistry 11](#_Toc13150)

[class UserInfoDao 11](#_Toc22826)

[class UserInfoEntity 11](#_Toc17437)

[com.openreuse.server.request.dispatcher 12](#_Toc12011)

[interface Dispatcher void dispatch(Message message); 12](#_Toc27005)

[class RouteDispatcher implements Dispatcher 12](#_Toc3845)

[com.openreuse.server.request.json 12](#_Toc3111)

[class ParseJsonService 12](#_Toc17218)

[class RawBytesWorker implements Worker 12](#_Toc27486)

[class RawBytesWorkerPool implements WorkerPool<RawBytesWorker> 12](#_Toc16374)

[com.openreuse.server.request.route 12](#_Toc31701)

[interface Route boolean route(Message message); 12](#_Toc31218)

[class LoginRoute implements Route 12](#_Toc3681)

[class RegisterRoute implements Route 13](#_Toc30959)

[class LogoutRoute implements Route 13](#_Toc1318)

[class TextRoute implements Route 13](#_Toc29325)

[com.openreuse.server.request.session 13](#_Toc21287)

[class CacheMapRefresher extends TimerTask(未使用~~~~~~) 13](#_Toc23067)

[class SessionManager 13](#_Toc8550)

[com.openreuse.server.response 14](#_Toc29631)

[abstract class ResponseTask implements Runnable 14](#_Toc3575)

[class ResponseHelper 14](#_Toc11703)

[class ResponseService 14](#_Toc23728)

[class UnicastResponseTask extends ResponseTask 15](#_Toc22729)

[class MulticastResponseTask extends ResponseTask 15](#_Toc28256)

[com.openreuse.server.response.listener 15](#_Toc24867)

[class ThrottleStatsListener implements Runnable 15](#_Toc15026)

[com.openreuse.server.throttle 15](#_Toc29489)

[class ThrottleManager 15](#_Toc10784)

[class ThrottleStatsManager 15](#_Toc5177)

[com.openreuse.server.validate 16](#_Toc22704)

[class Validators 16](#_Toc13496)

**Client接口 com.openresure.client**

**com.openresure.client**

class ClientAgent API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| registerListener(type:MessageType, listener:MessageListener):void | 注册监听表 | type:MessageType, listener:MessageListener |  |  |
| configureServer(ipAddr:String):void | 根据ip连接服务器 | ipAddr:String |  |  |
| registerValidate(username:String, password:String):boolean | 注册 | username:String, password:String |  |  |
| loginValidate(ipAddr:String, username:String, password:String):boolean | 登录 | ipAddr:String, username:String, password:String | 是否成功 |  |
| logout(username:String):void | 登出 |  |  |  |
| sendTextMessage(username:String, content:String):void | 发消息 |  |  |  |

**com.openresure.client.config**

class ConfigManager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ConfigManager | 新建并获得类实例 |  |  |  |
| refreshStatsToFile():void | 统计登录注册消息数据写入文件 |  |  |  |
| getCurrentServerAddr():String | 获得当前连接服务器地址 |  | 当前连接服务器ip地址 |  |
| setCurrentServerAddr(ipAddr:String):void | 设置当前要连接的服务器ip | Ip地址 |  |  |
| getChannel():Channel | 获得通道 |  | 当前通道 |  |
| setChannel(channel:Channel):void | 设置通道 | 通道 |  |  |
| getChannelLock():Lock | 获得通道锁 |  | 锁 |  |
| setChannelLock(channelLock:Lock):void | 设置通道锁 | 通道锁  channelLock:Lock |  |  |
| getUsrFrom():String | 获得用户 |  |  |  |
| setUsrFrom(usrFrom:String):void | 设置用户 | usrFrom:String |  |  |
| isLogined(username:String):boolean | 获得登录状态 | 用户名  username:String | 是否登录 | 返回对应用户登录状态关键字 |
| setLogined(username:String):void | 设置登录状态 | 用户名  username:String |  | 设置对应用户登录状态关键字为真 |
| unsetLogined(username:String):void | 设置登录状态 | 用户名  username:String |  | 设置对应用户登录状态关键字为假 |
| setChannelFuture(future:ChannelFuture):void | 设置通道异步操作 | future:ChannelFuture |  |  |

**com.openresure.client.handler**

class IncomingMessageHandler extends ChannelInboundHandlerAdapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| channelActive(ctx:ChannelHandlerContext):void | 设置并启动通道 | ctx:ChannelHandlerContext |  |  |
| channelRead(ctx:ChannelHandlerContext, msg:Object):void | 设置通道  读入消息 | ctx:ChannelHandlerContext, msg:Object |  |  |
| exceptionCaught(ctx:ChannelHandlerContext, e:Throwable):void | 处理通道处理异常 | ctx:ChannelHandlerContext, e:Throwable |  | 关闭通道 |

**com.openresure.client.listener**

interface MessageListener

class ValidateLoginListener implements MessageListener

class ValidateRegisterListener implements MessageListener

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| onMessageArrive(body:String):void | 收到消息监听并改变manager状态 | 消息内容  Body:String |  | 在login实现中设置body返回用户名，将manager状态设为登入；  在register实现中判断用户名是否为消息内容并改变注册状态 |
| isValid():boolean | 确认收到 |  | 真 |  |

**com.openresure.client.queue**

class MessageQueue

所有client使用同一个队列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getMessage():Message | 取得消息队列消息 |  |  | 队列对消息数量进行限制 |
| putMessage(message:Message):void | 放消息进入消息队列 | message:Message |  | 对一秒内发消息进行限制 |

**com.openresure.client.sender**

class MessageSender implements Runnable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| run():void | 将消息写入channel |  |  | 无消息通道解锁，有消息写入通道  writeAndFlush |

**com.openresure.client.service**

class ConnectionMgmtService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ConnectionMgmtService | 新建并获得类实例 |  |  |  |
| doConnect(ipAddr:String):boolean | 连接服务器 | 服务器地址  ipAddr:String | 连接成功为真 |  |

class MessageNotifyService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():MessageNotifyService | 新建并获得类实例 |  |  |  |
| registerListener(type:MessageType, listener:MessageListener):void | 根据消息类型注册消息监听器 | type:MessageType  listener:MessageListener |  | 将监听注册为静态，且对应监听类型和监听事件的map也为静态 |
| publish(message:Message):void | 消息处理，根据输入的消息进行相应操作，调用监听器 | message:Message |  | 消息处理的核心函数，根据map调用监听器 |

class MessageSendingService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():MessageSendingService | 新建并获得类实例 |  |  | 获得通道，启动发消息的线程 |
| provideMessage(message:Message):void | 将消息放入队列 | message:Message |  |  |
| consumeMessage():Message | 取得队列中消息 |  | 队列中消息 |  |

**Common接口 com.openresure.common**

**com.openreuse.common.message.builder**

class MessageBuilder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| messageBuilder():MessageBuilder | 新建并获得类实例 |  |  | 信息的建立 |
| build():Message | 获得信息 |  | Message |  |
| setBody(body:String):MessageBuilder | 设置信息内容 | body:String | MessageBuilder |  |
| setFrom(from:String):MessageBuilder | 设置信息来源地 | from:String | MessageBuilder |  |
| setType(type:MessageType):MessageBuilder | 设置信息类型 | type:MessageType | MessageBuilder |  |
| setTo(to:String):MessageBuilder | 设置信息目的地 | to:String | MessageBuilder |  |
| setReserved(reserved:Reserved):MessageBuilder | 设置保留字 | reserved:Reserved | MessageBuilder |  |

**com.openreuse.common.message**

class Message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getTo():String | 获得信息目的地 |  | 当前信息目的地 |  |
| setTo(to:String):void | 设置信息目的地 | 信息目的地  to:String |  |  |
| getBody():String | 获得信息内容 |  | 当前信息内容 |  |
| setBody(body:String):void | 设置信息内容 | 信息内容  body:String |  |  |
| getReserved():Reserved | 获得信息存储状态 |  | 当前信息存储状态 |  |
| setReserved(reserved:Reserved):void | 设置信息存储状态 | 信息存储状态  reserved:Reserved |  |  |
| getType():MessageType | 获得信息类型 |  | 当前信息类型 |  |
| setType(type:MessageType):void | 设置信息类型 | 信息类型  type:MessageType |  |  |
| getFrom():String | 获得信息来源地 |  | 信息来源地 |  |
| setFrom(from:String):void | 设置信息来源地 | 信息来源地  from:String |  |  |

enum MessageType 信息类型枚举

LOGIN\_MESSAGE("login"),  
LOGOUT\_MESSAGE("logout"),  
REGISTER\_MESSAGE("register"),  
TEXT\_MESSAGE("text"),  
HEARTBEAT\_MESSAGE("heartbeat"),  
RESP\_OK("response\_ok"),  
RESP\_INTERNAL\_ERROR("response\_error"),  
RESP\_REFUSE("response\_refuse"),  
REDO\_LOGIN\_MESSAGE("redo\_login");

class Reserved(**作为保留字为后续开发留空间**)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getContent():String | 获得内容 |  |  |  |
| setContent(content:String):void | 设置内容 | content:String |  |  |

**com.openreuse.common.persist**

class LocalPersistHelper

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| persistMapToLocal(map:Map, file:File):String | 输出性能管理信息到文件 | map:Map, file:File | String | Map将所需信息转换成json形式并存储到文件，进程阻塞 |
| loadLocalToMap(file:File):Map | 解析文件信息载入 | file:File | Map | ??????????  没有用过 |

**Server接口 com.openreuse.server**

**先worker将信息提出，然后dispatch，然后将任务放入taskqueue**

class ServerApplication

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| main(args:String[]):void | Server应用启动函数 |  |  | 建立Netty服务器 |

**com.openreuse.server.handler**

class RawJsonHandler extends ChannelInboundHandlerAdapter

Netty的handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| channelActive(ctx:ChannelHandlerContext):void | 通道开启 | ctx:ChannelHandlerContext |  |  |
| channelRead(ctx:ChannelHandlerContext, msg:Object):void | 通道信息读入 | ctx:ChannelHandlerContext, msg:Object |  | 对SessionManager处理 |
| channelReadComplete(ctx:ChannelHandlerContext):void | 通道读入完成 | ctx:ChannelHandlerContext |  |  |
| exceptionCaught(ctx:ChannelHandlerContext, cause:Throwable):void | 异常处理 | ctx:ChannelHandlerContext, cause:Throwable |  |  |

**com.openreuse.server.misc**

class Constants

设置静态变量 对各种限制约束的初始化

**com.openreuse.server.misc.worker**

interface Worker extends Runnable

interface WorkerPool<Textends Worker>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| startWorkers():void |  |  |  |  |

**com.openreuse.server.pojo.client;**

class AbstactClient

**com.openreuse.server.pojo.notify**

class DelayedNotify implements Delayed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| now():long | 获得当前延迟时间 |  |  |  |
| compareTo(delayed:Delayed):int | 以纳秒计判断延迟情况 | delayed:Delayed | Int与传入的延迟比较，超出返回值为1 |  |
| getDelay(unit:TimeUnit):long | 获得运行时间和当前时间差 | unit:TimeUnit | 运行时间和当前时间差 |  |
| getNanoDelay():long | 获得延迟时间 |  | 延迟时间 | Default invalid delay value |
| getClientId():long | 获得延迟的客户id |  | 延迟的客户id | Default invalid client ID |

**com.openreuse.server.registry**

class RegistryManager

Validators.validateClientID（断言判断用户id）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():RegistryManager | 新建并获得类实例 |  |  | 对Map或许应该有一个上限 |
| registerLogin(clientID:long):void | 注册和登陆 | clientID:long |  | 太多的登陆可能会造成OOM  DelayedNotify  ThrottleManager |
| checkLogin(clientID:long):boolean | 查询登陆状态 | clientID:long | 登陆超过三次为假 | 使登陆违规下线 |
| countDownLogin(clientID:long):void | 减少登陆次数 | clientID:long | 为1时登陆计数删除，大于1时将其减少1 | 用于每分钟的三次登陆限制 |

**com.openreuse.server.registry.validateRegistry**

class UserInfoDao

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| addUser(username:String, password:String):long | 通过用户名和密码注册 | username:String, password:String | 用户id |  |
| userInsert(userInfoEntity:UserInfoEntity):void | 对数据库添加记录 | userInfoEntity:UserInfoEntity |  | session.save |
| existOrNot(username:String, password:String):boolean | 检查用户是否存在于数据库 | username:String, password:String | 存在为真 |  |
| getIdbyName(username:String):long | 使用用户名获得用户id | username:String | 用户id |  |
| validatePassword(username:String, password:String):boolean | 判断用户名密码吻合 | username:String, password:String | 吻合为真 |  |

class UserInfoEntity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getUserid():int | 获得用户id |  | 用户id int |  |
| setUserid(userid:int):void | 设置用户id | userid:int |  |  |
| getUsername():String | 获得用户名 |  | 用户名 |  |
| setUsername(username:String):void | 设置用户名 | username:String |  |  |
| getPassword():String | 获得密码 |  | 密码 |  |
| setPassword(password:String):void | 设置密码 | password:String |  |  |
| equals(o:Object):boolean | 判断两个记录是否相同 | o:Object | 相同为真 |  |
| hashCode():int | 获得记录的哈希码 |  | 用户id用户名，密码的哈希码 |  |

**com.openreuse.server.request.dispatcher**

interface Dispatcher void dispatch(Message message);

class RouteDispatcher implements Dispatcher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():RouteDispatcher | 新建并获得类实例 |  |  |  |
| dispatch(message:Message):void | 对信息调度 | message:Message |  | 根据信息类型对处理路径进行调度 |

**com.openreuse.server.request.json**

class ParseJsonService?????

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ParseJsonService | 新建并获得类实例 |  |  |  |
| provideRawBytes(rawBytes:byte[]):void | 提供线程队列的byte | rawBytes:byte[] |  | BlockingQueue |

class RawBytesWorker implements Worker

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| work():void | 执行者对消息的操作 |  |  |  |
| run():void | 收到消息调用work() |  |  |  |

class RawBytesWorkerPool implements WorkerPool<RawBytesWorker>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| startWorkers():void | 执行线程启动，执行者开始对不同用户工作 |  |  |  |
| stopWorkers():void | 执行线程关闭，执行者停止对不同用户工作 |  |  |  |

**com.openreuse.server.request.route**

interface Route boolean route(Message message);

class LoginRoute implements Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| route(message:Message):boolean | 登陆消息处理路径 |  | 登陆成功返回真，否假 | 调用manager计数记录  产生回复消息 |

class RegisterRoute implements Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| route(message:Message):boolean | 注册消息处理路径 |  | 注册成功返回真，否假 | 调用manager计数记录  产生回复消息 |

class LogoutRoute implements Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| route(message:Message):boolean | 登出消息处理路径 |  | 登出成功返回真，否假 | 调用manager计数记录  产生回复消息 |

class TextRoute implements Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| route(message:Message):boolean | 文本消息处理路径 |  | 发文本消息成功返回真，否假 | 调用manager计数记录  产生回复消息 |

**com.openreuse.server.request.session**

class CacheMapRefresher extends TimerTask(未使用~~~~~~)

class SessionManager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():SessionManager | 新建并获得类实例 |  | SessionManager |  |
| saveChannelForUsr(from:String, channel:Channel):void | 保存用户的通道 | from:String, channel:Channel |  |  |
| removeChannelForUsr(from:String):void | 删除用户的通道 | from:String |  |  |
| getChannelForUsr(from:String):Channel | 获取用户的通道 | from:String | 用户的通道Channel |  |
| getSession(clientId:long):Channel | 获取用户会话 | clientId:long | 用户会话Channel |  |
| haveSession(channel:Channel):boolean | 判断用户会话信息 | channel:Channel | 含该用户会话为真 |  |
| registerSession(clientId:long, channel:Channel):void | 注册会话信息 | clientId:long, channel:Channel |  |  |
| removeSession(clientId:long):void | 移除会话信息 | clientId:long |  |  |
| sessionIterator():Iterator<Entry<Long, Channel>> | 获取会话迭代指针 |  | 会话迭代指针  Iterator<Entry<Long, Channel>> |  |
| getUsrId(from:String):Long | 获取用户id | from:String | 用户id | 放入用户map |
| initTimer():void | 计时器 |  |  | 1秒刷新一次用户map和会话map |
| registerUsr(username:String, password:String):long | 注册用户名密码 | username:String, password:String | 用户id |  |
| validUsrAndPass(username:String, password:String):boolean | 匹配用户名密码 | username:String, password:String | 用户存在于数据库为真 |  |

**com.openreuse.server.response**

abstract class ResponseTask implements Runnable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| run():void | 运行空接口 |  |  |  |
| getMessage():Message | 获得当前信息 |  |  |  |
| setMessage(message:Message):void | 设置信息 |  |  |  |

class ResponseHelper

设置应答消息静态变量

class ResponseService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ResponseService | 新建并获得类实例 |  |  |  |
| sendMessage(message:Message, username:String):void | 单播发消息 | 消息message:Message,  用户： username:String |  | 异步回调  开启ThrottleStat监听 |
| sendMessage(message:Message):void | 广播发消息 | message:Message |  | 异步回调  开启ThrottleStat监听 |

class UnicastResponseTask extends ResponseTask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| run():void | 单播消息 |  |  | 遍历找出相同的uid进行单播 |

class MulticastResponseTask extends ResponseTask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| run():void | 广播消息 |  |  | 遍历在线用户 |

**com.openreuse.server.response.listener**

class ThrottleStatsListener implements Runnable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| run():void | ThrottleStatsManager对收到的消息进行计数 |  |  |  |

**com.openreuse.server.throttle**

class ThrottleManager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ThrottleManager | 新建并获得类实例 |  |  |  |
| notity(notify:DelayedNotify):void | 判断是否阻塞 | notify:DelayedNotify |  |  |
| startWorker():void | throttleWorker开始工作 |  |  | 即将throttleWorker加入线程 |
| stopWorker():void | Worker停止工作 |  |  |  |

class ThrottleStatsManager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| getInstance():ThrottleStatsManager | 新建并获得类实例 |  |  |  |
| refreshStatsMap():void | 输出所有消息状态信息 |  |  |  |
| incMsgCount(uid:long):void | 对登陆后的用户发消息计数 | uid:long |  |  |
| checkMsgCount(uid:long):boolean | 看一用户所发信息数量是否合格 | uid:long | 比MAX\_MSG\_CNT\_PER\_LOGIN)大返回假 | 实现对同一用户发消息次数的限制 |
| clrMsgCount(uid:long):void | 将用户发消息数目记录清零 | uid:long |  |  |
| incForwardedMsgCount():void | 发出的信息计数 |  |  |  |
| clrForwardedMsgCount():void | 发出的信息清零 |  |  |  |
| incValidLoginCount():void | 登陆生效计数 |  |  |  |
| clrValidLoginCount():void | 登陆生效清零 |  |  |  |
| incInvalidLoginCount():void | 登陆未生效计数 |  |  |  |
| clrInvalidLoginCount():void | 登陆未生效清零 |  |  |  |
| incReceivedMsgCount():void | 收到的消息计数 |  |  |  |
| clrReceivedMsgCount():void | 收到的消息清零 |  |  |  |
| incIgnoredMsgCount():void | 忽略的消息计数 |  |  |  |
| clrIgnoredMsgCount():void | 忽略的消息清零 |  |  |  |

**com.openreuse.server.validate**

class Validators

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法** | **功能** | **输入参数** | **返回值** | **备注** |
| validateNotify(notify:DelayedNotify):boolean | 判断用户是否被阻塞 | notify:DelayedNotify | 阻塞为假 |  |
| validateClientID(clientID:long):boolean | 判断用户是否有效 | clientID:long | 无效为假 |  |