## Utils包

**ClientUtil：客户端工具类**

static void ConnectServer(String ip,int port);//建立连接

void Send(Message m);//单向通信，只管发送，不管是否收到

Result SyscSend(Message m);//同步通信，有返回值

Result AyscSend(Message m);//异步通信

Message Receive();//同步接受消息

**ServerUtil：服务器工具类**

static Result init(int port);//初始化，等待连接

PersistenceUtil：持久化工具类，创建程序时初始化，退出程序时清除

Result SyscStore(MyQueue queue);//同步刷盘

Result AyscStore(MyQueue queue);//异步刷盘

MyQueue ReadMessageQueue();//读取消息队列

MyQueue ReadDeadQueue();//读取死信队列

FormatTransformUtil：格式转换工具包

Message toMessage（String s）；//转换为消息

String toString（Message s）；//转换为字符串

## Common包

**MessageType**：消息类型类（final类，0表示单向消息，1表示要回复的消息，2表示心跳包，都是常量（public static final），变量名大写）

**Message：消息类**

定义传输协议：

消息长度（字符串的4位）+消息类型+消息内容

private String message;//消息的成员变量

private int type；//消息类型

private int length；//消息长度

Public Message（String s）；//构造方法创建消息，默认消息类型为0

Public Message（String s，int type）；

void setType（int type）；//设置消息类型，注意检查类型是否存在

int getType（）；//获取消息类型

void setLength（int length）；//设置消息长度，长度限制（0-9999）

int getLength（）；//获取消息长度

void setContent（String content）；//设置消息内容，注意检查消息长度

String getContent（）；//获取消息内容

**IpNode**：//描述ip和端口号

private成员变量String ip，int port

构造方法public IpNode（String ip，int port）

Ip和port的get，set方法

**Topic：主题类**

private static List<IpNode> address；//存储地址的Map，实现为HashMap

private static String NameServerIp；

private static String NameServerPort；

public Topic（String s/\*主题名称\*/，int queueNum，String ip，int port）；//构造方法，定义主题内容，消息队列数，NameServer地址

List<IpNode> getAddress() ;//IpNode格式为 ip，port，给NameServer发送一个数据包（通过ClientUtil类），包含主题内容，请求队列的数量，特殊值判断：返回值为null/返回值为空List

**final HeartBeat** extends Message：心跳包//默认content为空

Private static final String message；//常量Message

Private static final int type；//固定为2

Private static final int length；//常数

public HeartBeat();

int getType（）；//获取消息类型

int getLength（）；//获取消息长度

**Result:结果类**//0表示失败，1表示成功，-1表示抛出异常

Private int state；//状态变量

public Result（int state）；

public int getResult（）；

## NameServer包

**QueueData**：//队列所在地址，队列编号

private成员变量IpNode ipnode，String queueName；

构造方法如上，get，set方法

**MQManager**:注册中心

//主要功能：

//1. broker的注册服务

//2.每5秒接收broker的心跳包，维护

//3. 对producer提供broker和队列分配服务

成员变量：

Private HashMap<String/\*BrokerName\*/，IpNode> BrokerMap；

Private HashMap<String/\*TopicName\*/，List<QueueData>> QueueMap；

void init（）；//初始化

new Client(String ip,int port);//作为客户端对Broker发送消息

new Server(int port);// 在某个端口上监听，作为服务端接收其他节点的消息

Void addBroker(String BrokerName，IpNode ipnode);//向brokermap中添加Broker

List<IpNode> getBrokerList（）;//将BrokerMap中的value元素生成一个LinkedList返回

List<QueueData> getQueue(String TopicName);//根据Topic找到对应的QueueData