Tom's Blog -

ZeroMQ - Fun Part 1

MARKDOWN # ZeroMQ - Mindgames and Samples ## Router bound, 2 Dealers connected to Router ![Image](images/zmq_rdd.png) In this sample: * Send data from **dealer1 to router** * Create message to **push data from router to dealer1 and dealer2** ```java package org.tt.zmq.tests; import java.nio.charset.Charset; import org.junit.Test; import org.zeromq.ZContext; import org.zeromq.ZMQ; import org.zeromq.ZMsg; public class BRouterToCDealer { @Test public void testmain() { ZContext ctx = new ZContext(); ZMQ.Socket router = ctx.createSocket(ZMQ.ROUTER); // give the socket an identity. this name will be used as address if // this socket connects to a router. // this is important if you want to route to a socket that didn't // contact you before router.setIdentity("router".getBytes()); router.bind("tcp://127.0.0.1:9000"); ZMQ.Socket dealer1 = ctx.createSocket(ZMQ.DEALER); // give the dealer an identity. here it is actually important dealer1.setIdentity("dealer1".getBytes()); dealer1.connect("tcp://127.0.0.1:9000"); ZMQ.Socket dealer2 = ctx.createSocket(ZMQ.DEALER); // give the dealer an identity. here it is actually important dealer2.setIdentity("dealer2".getBytes()); dealer2.connect("tcp://127.0.0.1:9000"); // send data to router dealer1.send("d1-data"); // receive data as msg from router. // the router adds one frame with the address of the caller to be able // to send it back again ZMsg msg = ZMsg.recvMsg(router); // get that address that is added by the router String addressFromCaller = msg.popString(); assert (addressFromCaller.equals("dealer1")); // get the actual data sent by dealer1 String dataSentByCaller = msg.popString(); assert (dataSentByCaller.equals("d1-data")); System.out.println("router got msg:" + msg); // send data from router to dealer1 // stack togather: // first the address router.sendMore("dealer1"); // followed by the data that should be sent to dealer1 // the router gets the whole msg and expects the first frame to be a // routing-address // if this address is in a internal table it pops the address again and // sends the rest to this connection router.send("r1 - hi dealer1"); // you can send as many msgs as you want (and as the router-queue can // handle) // now let's send a message to dealer2 using ZMsg ZMsg msgToD2 = new ZMsg(); // here we have to stack upwards down. // first: the data msgToD2.push("r1 - hi dealer2"); // second: the routing-address msgToD2.push("dealer2"); msgToD2.send(router); // now let's receive the msgs the server sent msg = ZMsg.recvMsg(dealer1); String incomingMsgD1 = msg.popString(); assert (incomingMsgD1.equals("r1 - hi dealer1")); System.out.println("dealer1:" + incomingMsgD1); // get the data directly from the socket String incomingDataD2 = dealer2.recvStr(Charset.defaultCharset()); assert (incomingDataD2.equals("r1 - hi dealer2")); System.out.println("dealer2:" + incomingDataD2); // check if there are more parts left in the msgs we are now reading // line by line (in our case,no) boolean moreData = dealer2.hasReceiveMore(); assert (!moreData); // close the context ctx.close(); } } ```

Published by Thomas Trocha
Fri Jan 06 13:02:00 CET 2017
http://thomas.trocha.com:80/pebble/zeromq-fun-part-1