

Sequencer.java

```
1  package com.distributed;
2  import java.rmi.*;
3  import java.net.*;
4  import java.io.*;
5
6  public interface Sequencer extends Remote
7  {
8      // join -- request for "sender" to join sequencer's multicasting service;
9      // returns an object specifying the multicast address and the first sequence number to expect
10     public SequencerJoinInfo join(String sender)
11         throws RemoteException, SequencerException;
12     // send -- "sender" supplies the msg to be sent, its identifier,
13     // and the sequence number of the last received message
14     public void send(String sender, byte[] msg, long msgID, long lastSequenceReceived)
15         throws RemoteException;
16
17     // Leave -- tell sequencer that "sender" will no longer need its services
18     public void leave(String sender)
19         throws RemoteException;
20
21     // getMissing -- ask sequencer for the message whose sequence number is "sequence"
22     public byte[] getMissing(String sender, long sequence)
23         throws RemoteException, SequencerException;
24
25     // heartbeat -- we have received messages up to number "lastSequenceReceived"
26     public void heartbeat(String sender, long lastSequenceReceived)
27         throws RemoteException;
28 }
29
```