
Protokoll: echo

By rac

Published: 15.01.2008 - 09:32

Java Implementation eines Echo Servers und eines Echo clients

- [echoclient.java](#)
- [echoserverThreaded.java](#)

[< Internet Technologien und Daten Protokolle](#) [up](#) [echoclient.java](#) [>](#)

echoclient.java

By rac

Published: 15.01.2008 - 09:36

1.
package hsz.ITDP.rac.uebungen.echoClientSrv;

2.
import java.io.BufferedReader;

3.
import java.io.IOException;

4.
import java.io.InputStreamReader;

5.
import java.io.PrintWriter;

6.
import java.net.Socket;

7.

8.

9.
public class echoclient {

```
10. public static void main( String [] args) throws IOException {  
  
11.     BufferedReader in = null;  
  
12.     PrintWriter out = null;  
  
13.     BufferedReader user = null;  
  
14.     Socket sock = null;  
  
15.     String srvIP = "localhost"; //args[0];  
  
16.     int srvPort = 9998; // Integer.parseInt(args[1]);  
  
17.  
  
18.     try {  
  
19.         sock = new Socket (srvIP, srvPort);  
  
20.         in = new BufferedReader (new InputStreamReader (sock.getInputStream()));  
  
21.         user = new BufferedReader (new InputStreamReader ( System .in));  
  
22.
```

```
    out = new PrintWriter (sock.getOutputStream(),true);

23.    boolean exit = false;

24.    while(!exit) {

25.        String userwrote = user.readLine();

26.        if(userwrote.equals("exit")) {

27.            exit = true;

28.        } else {

29.            out.println(userwrote);

30.            System .out.println(in.readLine());

31.        }

32.    }

33.

34. } catch ( Exception e) {
```

```
35.         // TODO: handle exception

36.     } finally {

37.         if (in != null)

38.             in.close();

39.         if (user != null)

40.             user.close();

41.         if (out != null)

42.             out.close();

43.         if (sock != null)

44.             sock.close();

45.     }

46. }

47.
```

}

[< Protokoll: echo up echoserverThreaded.java >](#)

echoserverThreaded.java

By rac

Published: 15.01.2008 - 09:40

1.
package hsz.ITDP.rac.uebungen.echoClientSrv;

2.
import java.io.BufferedReader;

3.
import java.io.IOException;

4.
import java.io.InputStreamReader;

5.
import java.io.PrintWriter;

6.
import java.net.ServerSocket;

7.
import java.net.Socket;

8.

9.

```
10. public class echoserverThreaded implements Runnable {
11.     private Socket sock = null;
12.     public echoserverThreaded( Socket sockin) {
13.         System .out.println("create thread for "+sockin.getRemoteSocketAddress());
14.         this.sock = sockin;
15.     }
16.     public void run() {
17.         System .out.println("run()");
18.         BufferedReader in = null;
19.         PrintWriter out = null;
20.         while ( true ) {
21.
22.
```

```
try {  
23.     in = new BufferedReader (new InputStreamReader (this.sock.getInputStream()));  
24.  
25.     out = new PrintWriter (this.sock.getOutputStream(),true);  
26.  
27.     boolean exit = false;  
28.  
29.     while(!exit || in != null) {  
30.  
31.         String userwrote = in.readLine();  
32.  
33.         if(userwrote.equals("exit")) {  
34.  
35.             exit = true;  
36.  
37.         } else {  
38.  
39.             System .out.println("User "+ sock.getInetAddress() +" on  
socket "+sock.getLocalSocketAddress()+"wrote: "+userwrote);  
40.  
41.             out.println(userwrote);  
42.  
43.         }  
44.     }  
45.
```

```
    }  
35.     } catch ( Exception e) {  
36.         // TODO: handle exception  
37.     } finally {  
38.         if (in != null)  
39.             try {  
40.                 in.close();  
41.             } catch ( Exception e) {  
42.                 // TODO: handle exception  
43.             }  
44.         if (out != null)  
45.             try {  
46.                 out.close();
```

```
47.         } catch ( Exception e) {
48.             // TODO: handle exception
49.         }
50.
51.     if (sock != null)
52.     {
53.         sock.close();
54.     } catch ( Exception e) {
55.         // TODO: handle exception
56.     }
57.
58. }
59.
```

```
    }  
  
60. }  
  
61.  
  
62. public static void main( String [] args) throws IOException {  
  
63.     try {  
  
64.         ServerSocket srvsock = null;  
  
65.         int srvPort = 9998; // Integer.parseInt(args[0]);  
  
66.         srvsock = new ServerSocket (srvPort);  
  
67.         while(true) {  
  
68.             Thread newThread = new Thread ( new echoserverThreaded(srvsock.  
accept() ) );  
  
69.             newThread.start();  
  
70.         }  
  
71.
```

```
72.  
73.     } catch ( Exception e) {  
74.         // TODO: handle exception  
75.     }  
76. }  
77.  
78.  
79. }
```

[< echoclient.java](#) [up](#) [Protokoll: SNTP >](#)