

## Using Tor in Java

This program connects to the website `whatismyipaddress.com` and filters the reply to find out what IP it appears to have. If `useTor` is true it tries to use the Tor Proxy, this requires Vidalia to be running.

Vidalia provides a SOCKS proxy. The only difference between the Tor and non-Tor connections is that the Tor socket is initially set up to use the proxy.

```
import java.io.*;
import java.net.*;
import java.util.regex.*

public class TorDemo {

    static boolean useTor = true;

    public static void main (String[] args) throws Exception {

        // Open a connection to the server
        Socket socket;
        // If using Tor go via a proxy, otherwise make a normal socket
        if (useTor) {
            SocketAddress addr = new InetSocketAddress("localhost", 9050);
            Proxy proxy = new Proxy(Proxy.Type.SOCKS, addr);
            socket = new Socket(proxy);
            InetSocketAddress dest
                = new InetSocketAddress("whatismyipaddress.com", 80);
            socket.connect(dest);
        }
        else
        {
            socket = new Socket("whatismyipaddress.com",80);
        }
        PrintWriter out = new PrintWriter (socket.getOutputStream(),true);
        BufferedReader rd = new BufferedReader(
            new InputStreamReader(socket.getInputStream()));
```

```

// Request the website using HTTP
out.write("GET / HTTP/1.0\n");
out.write("Host: whatismyipaddress.com\n");
out.write("\n");
out.flush();

// Read in the reply from the server and find the line with my IP
String line;
line=rd.readLine();
Pattern p = Pattern.compile("([\\d]+\\. [\\d]+\\. [\\d]+\\. [\\d]+)");
Matcher m;
while(line != null) {
    m = p.matcher(line);
    if (m.find()) {
        System.out.println("The server says your IP is: "+m.group(1));
        break;
    }
    line=rd.readLine();
}
socket.close();
}
}

```