Atitit.http代理的实现 代码java php c# python

# 代理服务器用途

代理服务器看成是一种扩展浏览器功能的途径。例如，在把数据发送给浏览器之前，可以用代理服务器压缩数据

调试器

数据搜集器

木马病毒

作者::  ★(attilax)>>>  绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

**package** aaa0proxy;

**import** java.io.\*;

**import** java.net.\*;

**import** log.Log;

**import** com.attilax.exception.ExUtil;

**import** com.attilax.json.AtiJson;

**import** com.attilax.net.SocketUtil;

**import** com.attilax.web.UrlX;

**public** **class** MyHttpProxy **extends** Thread {

**static** **public** **int** *CONNECT\_RETRIES* = 5; // 尝试与目标主机连接次数

**static** **public** **int** *CONNECT\_PAUSE* = 5; // 每次建立连接的间隔时间

**static** **public** **int** *TIMEOUT* = 20000; // 每次尝试连接的最大时间

**static** **public** **boolean** *logging* = **false**; // 是否记录日志

// 与客户端相连的Socket

**protected** Socket csocket;

**public** MyHttpProxy(Socket cs) {

csocket = cs;

start();//run ()

}

**public** **void** run() {

String reqHead\_firstLine = ""; // 读取请求头

String URL = ""; // 读取请求URL

String host = ""; // 读取目标主机host

**int** port = 80; // 默认端口80

Socket ssocket = **null**;

SocketUtil client\_SocketUtil=**null**;

SocketUtil server\_SocketUtil\_remote = **null**;

Log logger=**new** Log();

System.***out***.println("--log file:"+logger.LOGFILENAME\_C );

**try** {

client\_SocketUtil=**new** SocketUtil(csocket).setSoTimeout(*TIMEOUT*).setLogger(logger);

UrlX urlUtil=**new** UrlX();

**try** {

reqHead\_firstLine = client\_SocketUtil. readFirstLine().firstLine;

} **catch** (Exception e) {

System.***out***.println("---read first line ex,msg:"+e.getMessage());

// 、、+ "csocket inso:"+ AtiJson.toJson(csocket) );

ExUtil.*throwEx*(e);

}

**if**(reqHead\_firstLine.contains(".jpg"))

System.***out***.println("dbg");

// 抽取URL(<a href="http://www.baidu.com/">http://www.baidu.com/</a>)

//GET http://s.cimg.163.com/catchpic/E/E0/E007CE953D48E9E1EC2319B343B5940D.jpg.670x270.jpg HTTP/1.1

URL = urlUtil.getRequestURL(reqHead\_firstLine);

System.***out***.println("--url："+URL);

**if**(URL.contains("1366134690"))

System.***out***.println("dbg");

host = urlUtil.hostNoport(URL);

port=urlUtil.port(URL);

// s.cimg.163.com

server\_SocketUtil\_remote= **new** SocketUtil()

.createConn2remote(host, port, *CONNECT\_RETRIES*,

*CONNECT\_PAUSE*).setSoTimeout(*TIMEOUT*).setLogger(logger)

.write(reqHead\_firstLine+"").write(client\_SocketUtil.charFirstLineBreak);

server\_SocketUtil\_remote.pipe(client\_SocketUtil.is, client\_SocketUtil.os); // 建立通信管道;

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**new** SocketUtil( client\_SocketUtil).close();

**new** SocketUtil( server\_SocketUtil\_remote) .close();

logger.flush();logger.close();

System.***out***.println("--thd finish");

}

}

**public** **static** **void** startProxy(**int** port, Class clobj) {

**try** {

ServerSocket ssock = **new** ServerSocket(port); //if port use ,throw ex

/\*

\* java.net.BindException: Address already in use: JVM\_Bind

\* \*/

System.***out***.println("---start ok port:"+port);

**while** (**true**) {

Class[] sarg = **new** Class[1];

Object[] arg = **new** Object[1];

sarg[0] = Socket.**class**;

**try** {

java.lang.reflect.Constructor cons = clobj

.getDeclaredConstructor(sarg);

arg[0] = ssock.accept();

cons.newInstance(arg); // 创建HttpProxy或其派生类的实例

} **catch** (Exception e) {

Socket esock = (Socket) arg[0];

**try** {

esock.close();

} **catch** (Exception ec) {

}

}

}

} **catch** (IOException e) {

System.***out***.println("\nStartProxy Exception:");

e.printStackTrace();

}

}

// 测试用的简单main方法

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

System.***out***.println("在端口808启动代理服务器\n");

// MyHttpProxy.log\_S = file\_S;

// MyHttpProxy.log\_C = file\_C;

// MyHttpProxy.logging = true;

MyHttpProxy.*startProxy*(808, MyHttpProxy.**class**);

}

}