

关注微信公众号【OpenCV学堂】

-应知空谷幽人意，只在春风淡荡中

图像处理教学视频

数字图像处理-基础入门

数字图像处理-空间域卷积

数字图像处理-二值图像分析

OpenCV图像处理教程

OpenCV特征提取与对象检测教程

OpenCV For Android 基础入门

个人资料

gloomyfish

博客专家

原创257

粉丝3877

喜欢1114

评论1401

等级：博客7

访问量：287万+

积分：2万+

排名：329

博客专栏

Java数字图像处理与特效

文章：68篇

阅读：1071796

HTML5 Canvas编程

文章：14篇

阅读：276592

文章分类

图像处理 (123)

OpenCV4Android教程 (4)

Android开发 (14)

Android UI编程 (3)

HTML5 (18)

J2EE (22)

杂项技巧 (19)

Swing桌面开发 (30)

Java Scket编程 (6)

JavaScript开发 (6)

7

JAVA Socket 实现HTTP与HTTPS客户端发送POST与GET方式请求

标签：Java http协议

2013年12月03日

31122人阅读

评论(5)

收藏

举报

分类：Java Scket编程 (5)

版权声明：本文为博主原创文章，未经博主允许不得转载。 //blog.csdn.net/jia20003/article/details/17104791

JAVA Socket 实现HTTP与HTTPS客户端发送POST与GET方式请求

哇,一看标题怎么这么长啊,其实意思很简单,哥讨厌用HTTP Client做POST与GET提交觉得那个毕竟是别人写得API库,所以我就自己实现了一个简单的HTTP客户端,支持POST方式提交数据,GET方式查询数据,是测试Restful API比较方便点,然后支持form与JSON两种方式提交数据,当然也支持返回数据为JSON格式.当然这些东西都是基于Java Socket直接完成的,不借助任何第三方的库,主要是JDK的API其实已经够用啦. 当然我也没有用URLConnection这个东西,毕竟它在Socket基础上又包装了一下,有违我写这篇文章的目的.

好啦,讲完理由,下面就说说要怎么样才能实现啊,光说不练假把式啊!大致分了几个步骤

一:当然是要知道HTTP协议,知道常用的HTTP请求头,比如Host, Accept, Content-Type 知道HTTP协议支持的方法,常用有GET/POST/PUT/DELETE等

如果不知道,也不用担心,我保证你读完这篇文章,你就知道一些啦,当然有个最好的参考文档就是HTTP相关的RFC文档,认真读一下肯定解决你自己心中的HTTP那些疑惑

二: 知道发送HTTP GET与POST格式很重要, 固定的格式如下:

```
[REQUEST]<SP><URL><SP>[HTTP VERSION]<CLRF>
[REQUEST HEADER: ]<SP>[VALUE]<CLRF>
```

可以有多个请求头

最后<CLRF>

发送完HTTP请求头部以后, 针对不同请求如POST要发送内容部分,发送完成以后同样以<CLRF>结尾.

解释: <SP>表示空格, <CLRF>表示回车换行Java中表示为“\r\n”

REQUEST表示HTTP请求命令,可以为POST, GET, PUT, DELETE等之一

HTTP VERSION的常见可能值为HTTP/1.1或者HTTP/1.0

三: 如果1与2的知识你都具备了,下面就来介绍一下Java Socket的相关知识

如何创建一个Java客户端套接字Socket s = new Socket()如此即可,简单吧!

如何连接到远程的主机与端口, 当提供URL字符串时候,可以这么做

```
URL url = new URL("http://blog.csdn.net/jia20003");
String host = url.getHost();
int port = url.getDefaultPort();
SocketAddress dest = new InetSocketAddress(this.host, this.port);
s.connect(dest);
```

即可连接到远程主机,下面就可以请求文章内容了, 也很容易的

```
[java]
1. String path = "/jia20003/article/details/16908661";
2. SocketAddress dest = new InetSocketAddress(this.host, this.port);
3. socket.connect(dest);
4. OutputStreamWriter streamWriter = new OutputStreamWriter(socket.getOutputStream());
5. bufferedWriter = new BufferedWriter(streamWriter);
6. bufferedWriter.write("GET " + path + " HTTP/1.1\r\n");
7. bufferedWriter.write("Host: " + this.host + "\r\n");
8. bufferedWriter.write("\r\n");
9. bufferedWriter.flush();
```

http://blog.csdn.net/jia20003/article/details/17104791

1/6

网络协议研究 (4)

数据挖掘学习 (4)

ExtJS学习 (2)

VC++图形与图像 (1)

OSGi+Virgo (1)

文章存档

2018年1月 (1)

2017年11月 (1)

2017年9月 (1)

2017年8月 (1)

2017年6月 (4)

展开

阅读排行

图像处理-理解卷积

(93831)

图像处理之霍夫变换 (直线检...

(87335)

图像处理之常见二值化方法汇总

(80648)

将HTML5 Canvas的内容保存...

(70864)

HTML5 Canvas实现360度全...

(62901)

Java数字图像处理基础知识 - ...

(53202)

HighChart学习-更新数据data.

(47944)

HTML5 Canvas鼠标与键盘事件

(41692)

图像处理之高斯金字塔

(41392)

Spring3 MVC中使用Swagge...

(40965)

最新评论

OpenCV实现手写体数字训练与识别

这世界点点滴滴全部都是你 : 有没完整Demo给一份 参考参考 @qq628927197.com

图像处理之Canny边缘检测

decouples : 写的很不错,不知道有没有对应的C++版本的

工作11年 人生第二次跳槽的感悟

王靖凡--Sam : 觉得痛苦的话就找机会转管理

关河无尽处,风雪有行人 - 我的2...

改变自己chenyu : 加油,大神2

关河无尽处,风雪有行人 - 我的2...

改变自己chenyu : 你就是我的偶像,你现在做过路,我以后一要走成你这个样子,人生才不会后悔,我已经把你的博客链接地址...

图像处理之图像内插值与外插值

yy237247 : 楼主,我请教个问题,这样外插值,图像的像素值会溢出吗 其中原理?

OpenCV On Android...

TomSweet : [reply]liugt34[/reply] 以解决,按照@henry跃的做法,只复制libs目录...

OpenCV On Android...

TomSweet : 博主你好,我这边按您的配置结果运行起来会报错, No implementation found f o...

Tesseract OCR集成An...

danfengw : 博主tesseract是否可以识别图片中文字方向?

Java使用OpenCV3.2实现...

qq_16086191 : opencv的人脸检测不太准,有的头像检测不到,有的明显非头像却被检测到

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

最后也是最重要的一点,字符编码,尽量都用同一种字符编码来发

送请求数据,推荐使用utf-8测试程序, 我写两个简单的PHP文件放在wamp server上,完成对全部代码的测试

四:关于HTTP与HTTPS

HTTP协议是位于第四层协议TCP之上完成的应用层协议, 端到端都是明文传送,别人一旦网络抓包以后都可以看到你的提交与请求,这个好像不太安全. HTTP协议的默认端口是80这个是RFC文档声明的,属于官方标准,没什么道理可以讲. HTTPS是基于SSL加密传输的,这样别人截获的数据包破解的概率要小一点,比HTTP安全一点,其默认端口是443, 好像QQ邮箱与163的WEB Mail邮箱都是基于HTTPS. 但是HTTPS通信方式只是传输数据加密,都客户说是透明的,它还是一样的要遵守HTTP协议规范来发送POST与GET请求等. 完整的测试程序如下:

```
[java]
1. package com.gloomyfish.http.client;
2.
3. import java.io.BufferedReader;
4. import java.io.BufferedWriter;
5. import java.io.IOException;
6. import java.io.InputStreamReader;
7. import java.io.OutputStreamWriter;
8. import java.net.InetSocketAddress;
9. import java.net.Socket;
10. import java.net.SocketAddress;
11. import java.net.URLEncoder;
12.
13.
14. public class TestDemo {
15.     private int port;
16.     private String host;
17.     private Socket socket;
18.     private BufferedReader bufferedReader;
19.     private BufferedWriter bufferedWriter;
20.     public TestDemo(String host, int port) {
21.         socket = new Socket();
22.         this.host = host;
23.         this.port = port;
24.     }
25.
26.     public void sendGet() throws IOException
27.     {
28.         String path = "/zhigang/getDemo.php";
29.         SocketAddress dest = new InetSocketAddress(this.host, this.port);
30.         socket.connect(dest);
31.         OutputStreamWriter streamWriter = new OutputStreamWriter(socket.getOutputStream());
32.         bufferedWriter = new BufferedWriter(streamWriter);
33.
34.         bufferedWriter.write("GET " + path + " HTTP/1.1\r\n");
35.         bufferedWriter.write("Host: " + this.host + "\r\n");
36.         bufferedWriter.write("\r\n");
37.         bufferedWriter.flush();
38.
39.         BufferedInputStream streamReader = new BufferedInputStream(socket.getInputStream());
40.         bufferedReader = new BufferedReader(new InputStreamReader(streamReader, "utf-8"));
41.         String line = null;
42.         while((line = bufferedReader.readLine()) != null)
43.         {
44.             System.out.println(line);
45.         }
46.         bufferedReader.close();
47.         bufferedWriter.close();
48.         socket.close();
49.     }
50.
51.     public void sendPost() throws IOException
52.     {
53.         String path = "/zhigang/postDemo.php";
54.         String data = URLEncoder.encode("name", "utf-8") + "=" + URLEncoder.encode("gloomyfish", "utf-8") + "&" +
55.             URLEncoder.encode("age", "utf-8") + "=" + URLEncoder.encode("32", "utf-8")
56.         );
57.         // String data = "name=zhigang_jia";
58.         SocketAddress dest = new InetSocketAddress(this.host, this.port);
59.         socket.connect(dest);
60.         OutputStreamWriter streamWriter = new OutputStreamWriter(socket.getOutputStream(), "utf-8");
61.         bufferedWriter = new BufferedWriter(streamWriter);
62.         bufferedWriter.write("POST " + path + " HTTP/1.1\r\n");
```

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
63.         bufferedWriter.write("Host: " + this.host + "\r\n");
64.         bufferedWriter.write("Content-Length: " + data.length() + "\r\n");
65.         bufferedWriter.write("Content-Type: application/x-www-form-urlencoded\r\n");
66.         bufferedWriter.write("\r\n");
67.         bufferedWriter.write(data);
68.         bufferedWriter.flush();
69.         bufferedWriter.write("\r\n");
70.         bufferedWriter.flush();
71.
72.         BufferedInputStream streamReader = bufferedInputStream(socket.getInputStream());
73.         bufferedReader = new BufferedReader(new InputStreamReader(streamReader, "utf-8"));
74.         String line = null;
75.         while((line = bufferedReader.readLine()) != null)
76.         {
77.             System.out.println(line);
78.         }
79.         bufferedReader.close();
80.         bufferedWriter.close();
81.         socket.close();
82.     }
83.
84.     public static void main(String[] args)
85.     {
86.         TestDemo td = new TestDemo("127.0.0.1",8099);
87.         try {
88.             // td.sendGet(); //send HTTP GET Request
89.
90.             td.sendPost(); // send HTTP POST Request
91.         } catch (IOException e) {
92.             // TODO Auto-generated catch block
93.             e.printStackTrace();
94.         }
95.     }
96.
97.     <span style="font-size:18px;"></span>
```

想要成为支持HTTPS客户端, 只要在创建Socket的时候如下:

```
[java]
1.     <span style="font-size:18px;">socket = (SSLSocket)((SSLConnectionFactory)SSLSocketFactory.getDefault
    ().createSocket(this.host, port);</span>
```

所以有时候离开apache的HTTP Client等第三方Jar,程序员也是一样活!

- [上一篇](#) 图像处理之角点检测算法(Harris Corner Detection)
- [下一篇](#) HTML5 Canvas实现360度全景图

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

查看评论



湖心亭看雪2楼 2017-08-20 18:28发表

作者的代码好像并没有处理http响应读取超时什么的? 每次都是等待一分钟之后socket连接被服务器主动关闭了, 才从循环中跳出来的。。。搞不懂, 还是去看看okhttp源码吧




gloomyfishRe: 2017-08-20 21:34发表

回复湖心亭看雪: 我这个只是简单的代码演示, 什么okhttp源码我真的没兴趣看, 对我来说是浮云! 想用就用, 离开它我也一样做项目!



silema11楼 2016-11-18 09:49发表

CR(carrage return) LF (line feed)

一言不合就写socket的post和get请求  bupt073114 2016年08月02日 11:14 2309
rt。一言不合就写socket的post和get请求。写个桌面程序，利用java写get和post请求。测试成功； SocketReq.java package com.test.Cipher...

通过Socket实现Http中Get请求  lmy86263 2016年11月23日 20:28 353
对于网络通信这部分，一直搞不清楚网络模型中传输层 应用层协议是怎么进行交互的，于是想通过直接使用TCP协议实现http请求的发送，所以这里通过Socket编程来完成这项工作，在Java中还可以使用...

练习使用 socket 发送简单的 HTTP GET 请求
最近项目中有这样的需求，需要在游戏中内嵌手机验证模块，而手机绑定信息保存在运营后台，游戏服务器无法直接访问，所以需要游戏服务器向运营后台发送一些 HTTP 请求来获取这类的信息。 因为这部分功能写...

 zaffix 2013年02月06日 10:11 3781

使用JavaSocket编写发送HTTP_POST请求的工具类


完整版见<https://jadyer.github.io/>
 jadyer 2013年04月11日 14:52 3255

Java Socket 实现HTTP与HTTPS协议发送POST/GET请求


JAVA Socket 实现HTTP与HTTPS客户端发送POST与GET方式请求 哇,一看标题怎么这么长啊,其实意思很简单,哥讨厌用HTTP Client做POST与GET提交觉...
 zmx729618 2016年05月11日 15:00 6713


JAVA编写的一个简单的Socket实现的HTTP响应服务器进阶版

1、首先建立ServerSocket监听8000端口，等待浏览器的连接。 public class HttpServer { //WEB_ROOT该服务器的根目录，这个目录可以自己定义，主要是服务...
 u012945986 2016年09月20日 17:16 2005

socket模拟http发送请求  dodan 2016年08月01日 23:48 3477
import java.io.IOException; import java.io.InputStream; import java.io.OutputStreamWriter; import ja...

如何使用SOCKET 发送HTTP1.1 GET POST请求包

如何使用SOCKET 发送HTTP1.1 GET POST请求包HTTP报文是面向文本的，报文中的每一个字段都是一些ASCII码串，各个字段的长度是不确定的。HTTP有两类报文：请求报文和响应报文。请...
 yc0188 2009年10月29日 10:58 31751

使用Socket发送Http请求  sddxqlrjxr 2016年05月05日 13:45 3922
1、建立TCP连接。 SocketAddress addr = new InetSocketAddress("www.baidu.com",80); Socket sock = new Sock...

tcp socket客户端发送请求连接http服务

首先模拟一个http服务端，用于接收请求package bhttp.rserver.test;import java.io.BufferedReader; import java.io.IOExcep...

 cuiyaoqiang 2016年07月07日 13:30  1970

VC_Socket_GET_POST发送HTTP请求示例

2013年05月24日 16:32 24KB [下载](#)


7



一个简单的HTTP请求与应答基于socket

 MOVIE14 2017年08月03日 18:47  452

tomcat是一个web容器，网络请求基于HTTP，HTTP底层基于socket抽象层，直接上图（盗的图） 我就简单实现了一个socket来模拟HTTP请求与应答 package Sock...

C socket 发送HTTP请求

 u011680118 2015年10月20日 23:46  3397

HTTP请求头部样例： GET http://www.baidu.com/ HTTP/1.1 Accept: html/text Host: 220.181.6.175:80 Connec ti...

通过一个Socket例子理解Http协议（实战Java Web服务器 - - 使用HTTP协...

Web服务器与客户端的通信使用HTTP协议（超文本传输协议），所以也叫做HTTP服务器。用Java构造Web服务器主要用二个类，java.net.Socket和java.net.ServerSocket...

 rongwenbin 2015年08月31日 17:42  4010

java爬虫的2种爬取方式（HTTP||Socket）简单Demo(一)

最近在找java的小项目自己写着玩，但是找不到合适的，于是写开始学一点爬虫，自己也是感觉爬虫比较有趣。这里自己找了一个教程，这一次写的是基于Socket和http方式爬取。（1）SystemCo...

 qq_24489717 2016年10月16日 11:44  2463

Java Socket发送与接收HTTP消息简单实现

在上次Java Socket现实简单的HTTP服务我们实现了简单的HTTP服务，它可以用来模拟HTTP服务，用它可以截获HTT P请求的原始码流，让我们很清楚的了解到我们向服务发的HTTP消息的结构，对...

 a9529lty 2012年01月04日 10:52  38216

JAVA使用Socket实现的简易HTTP服务器

非阻塞IO的方式解析浏览器请求,封装为请求报文类 import java.io.IOException; import java.io.InputStream; imp ort java.util.Ar...

 ColdCoding 2016年11月16日 22:25  1451

JAVA编写的一个简单的Socket实现的HTTP响应服务器

JAVA编写的一个简单的Socket实现的HTTP响应服务器，看后就很容易理解Web服务器的原理了。package test.io; import java.net.*; import ...

 u012945986 2016年09月20日 08:29  3321

java socket http连接

import java.io.BufferedInputStream; import java.io.  redOutputStream; import java.io.ByteArrayInp...

7

 Mr_fang 2012年04月20日 17:44  666

微信开发中使用curl实现http与https...jst级get请求实例教程（附工程代码）

mark

 ljh504429906 2016年08月25日 17:06  3194