

一言不合就写socket的post和get请求

原创

2016年08月02日 11:14:47


2310

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

```
[java]
1. package com.test.CipherIndex;
2.
3. import java.io.BufferedInputStream;
4. import java.io.BufferedReader;
5. import java.io.BufferedWriter;
6. import java.io.ByteArrayOutputStream;
7. import java.io.DataOutputStream;
8. import java.io.IOException;
9. import java.io.InputStream;
10. import java.io.InputStreamReader;
11. import java.io.OutputStreamWriter;
12. import java.net.InetSocketAddress;
13. import java.net.Socket;
14. import java.net.SocketAddress;
15. import java.net.URL;
16. import java.net.URLEncoder;
17. import java.net.UnknownHostException;
18. import java.security.KeyManagementException;
19. import java.security.NoSuchAlgorithmException;
20. import java.security.Security;
21. import java.security.cert.CertificateException;
22. import java.security.cert.X509Certificate;
23.
24. import javax.net.ssl.*;
25. import com.sun.net.ssl.internal.www.protocol.https.*;
26.
27. public class SocketReq {
28.     private int port;
29.     private String host;
30.     private Socket socket;
31.     private BufferedReader bufferedReader;
32.     private BufferedWriter bufferedWriter;
33.
34.     public SocketReq() {
35.
36.     }
37.
38.     public SocketReq(String host, int port) {
39.         this.host = host;
40.         this.port = port;
41.
42.         try {
43.             socket = (SSLSocket) ((SSLSocketFactory) SSLSocketFactory
44.                 .getDefault()).createSocket(this.host, this.port);
45.         } catch (UnknownHostException e1) {
46.             // TODO Auto-generated catch block
47.             e1.printStackTrace();
48.         } catch (IOException e1) {
49.             // TODO Auto-generated catch block
50.             e1.printStackTrace();
51.         }
52.
53.         // SSLSocket
54.         // try {
55.         // socket = (SSLSocket) factory.createSocket("https://192.168.111.30",
56.         // 443);
57.         // } catch (UnknownHostException e) {
58.         // // TODO Auto-generated catch block
59.         // e.printStackTrace();
60.         // } catch (IOException e) {
61.         // // TODO Auto-generated catch block
62.         // e.printStackTrace();
63.         // }
64.
65.         // socket = new Socket();
```

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

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

广告

他的最新文章 更多文章

- 汉明码详细讲解
- CentOS 7 安装 MySQL
- linux端修改文件句柄
- OpenStack中SWIFT对接测试
- 一步搭建openstack

文章存档

2017年8月	1篇
2017年7月	1篇
2017年6月	3篇
2017年5月	2篇
2017年3月	2篇
2016年12月	1篇

展开

他的热门文章

- APK文件结构和安装过程
11124
- 有限域GF(2^8)内乘法代码实现以及原理
9927
- 如何在word中插入好看的C/C++代码
8069
- 在windows上编译openssl供Android NDK使用
7559
- 软著中写源代码60页快速实现方法
6807
- 中间件是什么？
6652
- 用C++写android程序（包含界面+发短信）
5636
- windows查看硬盘序列号
5106
- 1A1A打成1A1B后所有的调优方法

```

69. public void sendGet(String path) throws IOException {
70.
71.     // 自定义的管理器
72.     X509TrustManager xtm = new TrustAnyTrustManager();
73.     TrustManager mytm[] = { xtm };
74.     // 得到上下文
75.     SSLContext ctx;
76.     try {
77.         ctx = SSLContext.getInstance("SSL");
78.         // 初始化
79.         ctx.init(null, mytm, null);
80.
81.         // 获得工厂
82.         SSLSocketFactory factory = ctx.createSocketFactory();
83.
84.         // 从工厂获得Socket连接
85.         Socket socket = factory.createSocket(this.host, 443);
86.
87.         // 剩下的就和普通的Socket操作一样了
88.         BufferedWriter out = new BufferedWriter(new OutputStreamWriter(
89.             socket.getOutputStream()));
90.         BufferedReader in = new BufferedReader(new InputStreamReader(
91.             socket.getInputStream()));
92.         out.write("GET " + path + " HTTP/1.0\n\n");
93.         out.flush();
94.         System.out.println("start work!");
95.         String line;
96.         StringBuffer sb = new StringBuffer();
97.         while ((line = in.readLine()) != null) {
98.             sb.append(line + "\n");
99.         }
100.        out.close();
101.        in.close();
102.        String outcome = sb.toString();
103.        System.out.println(outcome);
104.
105.    } catch (NoSuchAlgorithmException e) {
106.        // TODO Auto-generated catch block
107.        e.printStackTrace();
108.    } catch (KeyManagementException e) {
109.        // TODO Auto-generated catch block
110.        e.printStackTrace();
111.    }
112.
113. }
114.
115.
116.
117. public void sendPost(String path) throws IOException {
118.     // String path = "/zhigang/postDemo.php";
119.     String data = URLEncoder.encode("name", "utf-8") + "="
120.         + URLEncoder.encode("gloomyfish", "utf-8") + "&"
121.         + URLEncoder.encode("age", "utf-8") + "="
122.         + URLEncoder.encode("32", "utf-8");
123.     // String data = "name=zhigang_jia";
124.     SocketAddress dest = new InetSocketAddress(this.host, this.port);
125.     socket.connect(dest);
126.     OutputStreamWriter streamWriter = new OutputStreamWriter(
127.         socket.getOutputStream(), "utf-8");
128.     bufferedWriter = new BufferedWriter(streamWriter);
129.
130.     bufferedWriter.write("POST " + path + " HTTP/1.1\n\n");
131.     bufferedWriter.write("Host: " + this.host + "\r\n");
132.     bufferedWriter.write("Content-Length: " + data.length() + "\r\n");
133.     bufferedWriter
134.         .write("Content-Type: application/x-www-form-urlencoded\r\n");
135.     bufferedWriter.write("\r\n");
136.     bufferedWriter.write(data);
137.     bufferedWriter.flush();
138.     bufferedWriter.write("\r\n");
139.     bufferedWriter.flush();
140.
141.     BufferedInputStream streamReader = new BufferedInputStream(
142.         socket.getInputStream());
143.     bufferedReader = new BufferedReader(new InputStreamReader(streamReader,
144.         "utf-8"));
145.     String line = null;
146.     while ((line = bufferedReader.readLine()) != null) {
147.         System.out.println(line.getBytes("utf-8"));
148.     }
149.     bufferedReader.close();
150.     bufferedWriter.close();
151.     socket.close();

```

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

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

```

156.     IOException {
157. byte[] ret = null;
158.
159. try {
160.     SSLContext sc = SSLContext.getInstance("SSL");
161.     sc.init(null, new TrustManager[] { new TrustAnyTrustManager() },
162.         new java.security.SecureRandom());
163.
164.     URL console = new URL(url);
165.     HttpURLConnection conn = (HttpURLConnection) console
166.         .openConnection();
167.     conn.setRequestMethod("POST");
168.
169.     conn.setRequestProperty("Connection", "Keep-Alive");
170.     conn.setRequestProperty("Charset", "UTF-8");
171.     conn.setRequestProperty("Content-Type",
172.         "multipart/form-data; boundary=*****");
173.     // 在与服务器连接之前, 设置一些网络参数
174.     conn.setConnectTimeout(10000);
175.
176.     conn.setDoOutput(true);
177.     conn.setSSLSocketFactory(sc.getSocketFactory());
178.     conn.setHostnameVerifier(new TrustAnyHostnameVerifier());
179.     conn.connect();
180.     DataOutputStream out = new DataOutputStream(conn.getOutputStream());
181.
182.     out.write(content.getBytes(charset));
183.     // 刷新、关闭
184.     out.flush();
185.     out.close();
186.     InputStream is = conn.getInputStream();
187.     try {
188.         if (is != null) {
189.             ByteArrayOutputStream outStream = new ByteArrayOutputStream();
190.             byte[] buffer = new byte[1024];
191.             int len = 0;
192.             while ((len = is.read(buffer)) != -1) {
193.                 outStream.write(buffer, 0, len);
194.             }
195.             // is.close();
196.             ret = outStream.toByteArray();
197.         }
198.     } catch (Exception e) {
199.         e.printStackTrace();
200.     } finally {
201.         if (null != is) {
202.             is.close();
203.         }
204.     }
205.
206. } catch (Exception e) {
207.     e.printStackTrace();
208. }
209. return ret;
210. }
211.
212. public static byte[] get(String url, String content, String charset)
213.     throws NoSuchAlgorithmException, KeyManagementException,
214.     IOException {
215.     byte[] ret = null;
216.     try {
217.         SSLContext sc = SSLContext.getInstance("SSL");
218.         sc.init(null, new TrustManager[] { new TrustAnyTrustManager() },
219.             new java.security.SecureRandom());
220.
221.         URL console = new URL(url);
222.         HttpURLConnection conn = (HttpURLConnection) console
223.             .openConnection();
224.         conn.setRequestMethod("GET");
225.
226.         // conn.setRequestProperty("Content-type",
227.         // "multipart/form-data; boundary=*****");
228.         // 在与服务器连接之前, 设置一些网络参数
229.         conn.setConnectTimeout(10000);
230.         conn.setReadTimeout(10000);
231.         // conn.setDoOutput(true);
232.         conn.setSSLSocketFactory(sc.getSocketFactory());
233.         conn.setHostnameVerifier(new TrustAnyHostnameVerifier());
234.
235.         conn.setRequestProperty("Connection", "Keep-Alive");
236.         conn.setRequestProperty("Charset", "UTF-8");
237.         conn.setRequestProperty("Content-Type",
238.             "application/x-www-form-urlencoded");
239.     }

```

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

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

```

243.
244.         try {
245.             if (is != null) {
246.                 ByteArrayOutputStream outStream = new ByteArrayOutputStream();
247.                 byte[] buffer = new byte[1024];
248.                 int len = 0;
249.                 while ((len = is.read(buffer)) != -1) {
250.                     outStream.write(buffer, 0, len);
251.                 }
252.                 ret = outStream.toByteArray();
253.             }
254.         } catch (Exception e) {
255.             e.printStackTrace();
256.         } finally {
257.             if (null != is) {
258.                 is.close();
259.             }
260.         }
261.     } catch (Exception e) {
262.         e.printStackTrace();
263.     }
264.    // 关闭输入流
265.
266.    return ret;
267. }
268.
269. private static class TrustAnyTrustManager implements X509TrustManager {
270.     TrustAnyTrustManager() {
271.         // 这里可以进行证书的初始化操作
272.     }
273.
274.     public void checkClientTrusted(X509Certificate[] chain, String authType)
275.         throws CertificateException {
276.         // System.out.println("检查客户端的可信任状态...");
277.     }
278.
279.     public void checkServerTrusted(X509Certificate[] chain, String authType)
280.         throws CertificateException {
281.         // System.out.println("检查服务器的可信任状态");
282.     }
283.
284.     public X509Certificate[] getAcceptedIssuers() {
285.         // System.out.println("获取接受的发行商数组...");
286.         return new X509Certificate[] {};
287.     }
288. }
289.
290. private static class TrustAnyHostnameVerifier implements HostnameVerifier {
291.     public boolean verify(String hostname, SSLSession session) {
292.         return true;
293.     }
294. }
295. }

```

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

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

CipherMain.java

```

[java]
1. package com.test.CipherIndex;
2.
3. import java.io.IOException;
4. import java.io.UnsupportedEncodingException;
5. import java.security.KeyManagementException;
6. import java.security.NoSuchAlgorithmException;
7.
8. import com.test.MainTest;
9.
10. public class CipherMain {
11.
12.     public static void main(String[] args) {
13.
14.         CipherMain cm = new CipherMain();
15.         // cm.test_post();
16.         cm.test_get();
17.     }
18.
19.     void test_post() {
20.
21.         SocketReq socketreq = new SocketReq();
22.         try {
23.             byte[] getret = socketreq.get(
24.                 "https://192.168.111.30/Mycloud/index.html", "1", "utf-8");

```

```
28.         // TODO Auto-generated catch block
29.         e.printStackTrace();
30.     } catch (NoSuchAlgorithmException e) {
31.         // TODO Auto-generated catch block
32.         e.printStackTrace();
33.     }
34.     catch (UnsupportedEncodingException e) {
35.         // TODO Auto-generated catch block
36.         e.printStackTrace();
37.     } catch (IOException e) {
38.         // TODO Auto-generated catch block
39.         e.printStackTrace();
40.     }
41. }
42.
43. void test_get() {
44.     SocketReq socketreq = new SocketReq();
45.     try {
46.         byte[] ret = socketreq.post(
47.             "https://192.168.111.30/Mycloud/index.html", "1", "utf-8");
48.
49.         String strRead = new String(ret, "utf-8");
50.
51.         System.out.println(strRead);
52.
53.     } catch (KeyManagementException e) {
54.         // TODO Auto-generated catch block
55.         e.printStackTrace();
56.     } catch (NoSuchAlgorithmException e) {
57.         // TODO Auto-generated catch block
58.         e.printStackTrace();
59.     }
60.     catch (UnsupportedEncodingException e) {
61.         // TODO Auto-generated catch block
62.         e.printStackTrace();
63.     } catch (IOException e) {
64.         // TODO Auto-generated catch block
65.         e.printStackTrace();
66.     }
67. }
68. }
69. }
```

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

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

👤 目前您尚未登录，请 [登录](#) 或 [注册](#) 后进行评论

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

C 语言Socket 实现http 带参数的POST请求

C 语言Socket 实现带参数的 http post请求

👤 liujiaoyage 2016年09月12日 15:59 💬 7234

SOCKET模仿HTTP POST请求

📄 ppt0501 2014年06月21日 17:11 💬 4280

通过Socket实现Http中Get请求

对于网络通信这部分，一直搞不清楚网络模型中传输层协议和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...

lmy86263

2016年11月23日 20:28

353

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

一个简单演示如果使用JAVA Socket实现HTTP GET请求...与HTTP POST提交数据的客户端 希望让更多的人知道写HTTP客户端并不难,离开HTTP Client等第三方库,JAVA一样...

jia20003

2013年12月03日 23:28

31123

四款网络通信测试工具-含socket/post/get http请求

2016年09月10日 14:09 9.33MB 下载

C# SOCKET GET/POST的实现

using System; using System.Net.Sockets; using System.Text;namespace ConsoleApplication1 { class WebT...

kjlrzzyffmx

2016年06月27日 05:14

2820

C++网络编程 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。 请求报文 一个HTTP请求报文由请求行（request li...

huangxy10

2012年06月14日 11:07

12190

socket编程---实现get post 向http发送请求

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。 请求报文 一个HTTP请求报文由请求行（request li...

lanzhihui_10086

2014年11月01日 21:31

4552

坑爹的hostname

1.这几天一直在搭建hbase环境，再虚拟机上搭建

u014133453

2014年09月02日 16:30

2082

纯C++实现的HTTP请求封装（POST/GET）

纯C++实现的HTTP请求（POST/GET），支持windows和linux，进行简单的封装，方便调用。实现如下：#include "HttpConnect.h"#ifdef WIN32 #p...

liukang325

2016年09月14日 15:18

7368

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

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

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册

http://blog.csdn.net/bupt073114/article/details/52092489

6/7

C++ Socket 发送 http post请求

isfirst

2016年07月20日 23:06

258

#include "stdafx.h" #include #include #include #include #include #incl
ude //dont forget to add wsock...

HttpsURLConnection实现SSL的GET/POST请求

0

CertificateException; import java.securi...

zmx729618

2016年05月11日 11:02

13632

https post请求，Nginx报405错误

u010882234

2017年02月13日 10:02

2943

最近在做微信开发，公司打算对产品增加微信支付功能，但是微信支付费率较高，于是与某微信支付的第三方合作，利用他们的平台，完成微信支付。经过测试，基本确定使用公众号支付方式。但在联调时，遇到一个非常坑的问...

轻松把玩HttpClient之配置ssl，采用绕过证书验证实实现https

上篇文章说道httpclient不能直接访问https的资源，这次就来模拟一下环境，然后配置https测试一下。在前面的文章中，分享了一篇自己生成并在tomcat中配置ssl的文章《Tomcat配置S...

xiaoxian8023

2015年11月16日 15:11

39905

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

一言不合就写socket的post和get请求

bupt073114

2016年08月02日 11:14

2310

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...

socket构造post请求实例

lifushan123

2015年04月05日 01:14

321

//请使用socket相关函数（非curl）实现如下功能：构造一个post请求，发送到指定http server的指定端口的指定请求路径（如http://www.example.com:8080/te...

用Python socket实现一个简单的http服务器（post 与get 的区别）、CGIHT...

预备知识：关于http协议的基础请参考这里。关于socket基础函数请参考这里。关于python网络编程基础请参考这里。一、python socke...

haluoluo211

2014年12月30日 09:55

3618

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

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