**java后台调用HttpURLConnection类模拟浏览器请求实例(可用于接口调用)**

投稿：shichen2014 字体：[[增加](javascript:turnbig()) [减小](javascript:turnsmall())] 类型：转载

这篇文章主要介绍了java后台调用HttpURLConnection类模拟浏览器请求实例,该实例可用于接口调用,具有一定的实用价值,需要的朋友可以参考下

一般在项目开发中难免遇到外部接口的调用,本文实例讲述了java后台调用HttpURLConnection类模拟浏览器请求的方法。可用于接口调用。分享给大家供大家参考。具体实现方法如下：

复制代码代码如下:

package com.cplatform.movie.back.test;

import java.io.BufferedReader;  
import java.io.DataOutputStream;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.net.URLEncoder;

public class HttpURLConnectionTest {  
    public static final String GET\_URL = "http://112.4.27.9/mall-back/if\_user/store\_list?storeId=32";  
    public static final String POST\_URL = "http://112.4.27.9/mall-back/if\_user/store\_list";  
      
    /\*\*  
     \* 接口调用 GET  
     \*/  
    public static void httpURLConectionGET() {  
        try {  
            URL url = new URL(GET\_URL);    // 把字符串转换为URL请求地址  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();// 打开连接  
            connection.connect();// 连接会话  
            // 获取输入流  
            BufferedReader br = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
            String line;  
            StringBuilder sb = new StringBuilder();  
            while ((line = br.readLine()) != null) {// 循环读取流  
                sb.append(line);  
            }  
            br.close();// 关闭流  
            connection.disconnect();// 断开连接  
            System.out.println(sb.toString());  
        } catch (Exception e) {  
            e.printStackTrace();  
            System.out.println("失败!");  
        }  
    }  
      
    /\*\*  
     \* 接口调用  POST  
     \*/  
    public static void httpURLConnectionPOST () {  
        try {  
            URL url = new URL(POST\_URL);  
              
            // 将url 以 open方法返回的urlConnection  连接强转为HttpURLConnection连接  (标识一个url所引用的远程对象连接)  
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();// 此时cnnection只是为一个连接对象,待连接中  
              
            // 设置连接输出流为true,默认false (post 请求是以流的方式隐式的传递参数)  
            connection.setDoOutput(true);  
              
            // 设置连接输入流为true  
            connection.setDoInput(true);  
              
            // 设置请求方式为post  
            connection.setRequestMethod("POST");  
              
            // post请求缓存设为false  
            connection.setUseCaches(false);  
              
            // 设置该HttpURLConnection实例是否自动执行重定向  
            connection.setInstanceFollowRedirects(true);  
              
            // 设置请求头里面的各个属性 (以下为设置内容的类型,设置为经过urlEncoded编码过的from参数)  
            // application/x-javascript text/xml->xml数据 application/x-javascript->json对象 application/x-www-form-urlencoded->表单数据  
            connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");  
              
            // 建立连接 (请求未开始,直到connection.getInputStream()方法调用时才发起,以上各个参数设置需在此方法之前进行)  
            connection.connect();  
              
            // 创建输入输出流,用于往连接里面输出携带的参数,(输出内容为?后面的内容)  
            DataOutputStream dataout = new DataOutputStream(connection.getOutputStream());  
            String parm = "storeId=" + URLEncoder.encode("32", "utf-8"); //URLEncoder.encode()方法  为字符串进行编码  
              
            // 将参数输出到连接  
            dataout.writeBytes(parm);  
              
            // 输出完成后刷新并关闭流  
            dataout.flush();  
            dataout.close(); // 重要且易忽略步骤 (关闭流,切记!)   
              
            System.out.println(connection.getResponseCode());  
              
            // 连接发起请求,处理服务器响应  (从连接获取到输入流并包装为bufferedReader)  
            BufferedReader bf = new BufferedReader(new InputStreamReader(connection.getInputStream()));   
            String line;  
            StringBuilder sb = new StringBuilder(); // 用来存储响应数据  
              
            // 循环读取流,若不到结尾处  
            while ((line = bf.readLine()) != null) {  
                sb.append(bf.readLine());  
            }  
            bf.close();    // 重要且易忽略步骤 (关闭流,切记!)   
            connection.disconnect(); // 销毁连接  
            System.out.println(sb.toString());  
      
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
      
    public static void main(String[] args) {  
//        httpURLConectionGET();  
        httpURLConnectionPOST();  
    }  
}