**post请求**

依赖

|  |
| --- |
| <!-- 发送http请求 -->  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpcore</artifactId>  <version>4.4.10</version>  </dependency>  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  <version>4.5.6</version>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>fastjson</artifactId>  <version>1.2.73</version>  </dependency> |

## 方式一

正常传对象方式，将对象转为json传输

|  |
| --- |
| public static String sendPostByJsonString(String url, String jsonString) {  *log*.info("网络请求开始####url:{}####jsonString:{}", url, jsonString);  String response = null;  try {  CloseableHttpClient httpclient = null;  CloseableHttpResponse httpresponse = null;  try {  httpclient = HttpClients.*createDefault*();  HttpPost httppost = new HttpPost(url);  httppost.setHeader("Content-Type", "application/json");  String newData = new String(jsonString.getBytes(), "utf-8");  StringEntity s = new StringEntity(newData, "utf-8");  s.setContentEncoding(new BasicHeader(HTTP.*CONTENT\_TYPE*, "application/json"));  httppost.setEntity(s);  httpresponse = httpclient.execute(httppost);  response = EntityUtils  .*toString*(httpresponse.getEntity());  } finally {  if (httpclient != null) {  httpclient.close();  }  if (httpresponse != null) {  httpresponse.close();  }  }  } catch (Exception e) {  e.printStackTrace();  }  *log*.info("网络请求结束####url:{}####jsonString:{}response:{}", url, jsonString, response);  return response; } |

## 方式二

在url后面拼接方式，如

<http://192.168.1.11:8080?aaa={sss}>

后面可能是多个字符串或者是一个key然后value是json。 或者?aa=1&bb=2像get请求一样。

|  |
| --- |
| public static void abcd() throws UnsupportedEncodingException {  *//1.创建HttpClient对象* CloseableHttpClient httpClient = HttpClients.*createDefault*();  *//2.创建HttpPost对象，设置URL地址* HttpPost httpPost = new HttpPost("https://udid.nmpa.gov.cn/api/v1/sharing/get");  ;   *//声明list集合，用来分装表单中的参数  //要求：设置请求的地址是：http://yun.itheima.com/search?keys=java* List<NameValuePair> params = new ArrayList<NameValuePair>();  Map map = new HashMap();  map.put("accessToken", "F7E47B12147282112453AB349579063E5E9764F953BA667E57EC3BA8859110D8B2F73B0D3F512C4FDDF6EA0EE64FD23647BB51E10D6C7CC7BBEB1175E92A4BADCF3F651C13B3E06002FFDB269741FBA67B255E740FFAACE1");  map.put("requestType", "3");  map.put("rangeValue", "");  map.put("currentPageNumber", "1");  params.add(new BasicNameValuePair("params", JSON.*toJSONString*(map)));  *// 创建表单的Entity对象,第一个参数是封装好的表单数据，第二个参数就是编码方式* UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(params, "utf8");  *//设置表单的Entity对象到Post请求中* httpPost.setEntity(formEntity);    *//使用httpClient发起响应获取repsonse* CloseableHttpResponse response = null;  try {  response = httpClient.execute(httpPost);  String s = JSON.*toJSONString*(response);  System.*out*.println("请求响应值：" + s);  *//4.解析响应，获取数据  //判断状态码是否是200* HttpEntity httpEntity = response.getEntity();  String s1 = JSON.*toJSONString*(httpEntity);  System.*out*.println("json的值是：" + s1);   String content = EntityUtils.*toString*(response.getEntity(), "UTF-8");  System.*out*.println("最终返回值：" + content);*//这里的返回值是最后的结果 // if (response.getStatusLine().getStatusCode() == 200) { // String content = EntityUtils.toString(httpEntity, "utf8"); // System.out.println(content.length()); // }* } catch (IOException e) {  e.printStackTrace();  } finally {  try {  response.close();  } catch (IOException e) {  e.printStackTrace();  }  try {  httpClient.close();  } catch (IOException e) {  e.printStackTrace();  }  }  } |