java 发送 HttpClient 请求

作者：NiuGeH
原文链接：https://ld246.com/article/1566375844216
来源网站：链滴

Java发送请求多用于一些安全性比较高的请求，比如说是调用支付接口，调用接口返的私密数据

需要的jar包 pom.xml 中引入 还有gson的jar自行在maven中搜索

```
<dependency>
   <groupId>commons-httpclient</groupId>
   <artifactId>commons-httpclient</artifactId>
   <version>3.1</version>
</dependency>
```

java 方法请求

```
//访问路径  自行修改
final String HTTP_URL = "https://dldecorate.club/solo";
final String CONTENT_TYPE_TEXT_JSON = "text/json";
try {

        DefaultHttpClient client = new DefaultHttpClient( new PoolingClientConnectionMana
er());
      //设置访问URL
        HttpPost httpPost = new HttpPost(HTTP_URL);
        //设置请求头为 json 格式的
      httpPost.setHeader("Content-Type", "application/json;charset=UTF-8");
        //创建HashMap 保存请求参数 便于json 转换
      HashMap<String,String> hashMap = new HashMap<>();
        hashMap.put("id","8888");
        hashMap.put("name","zhangsan");
        //创建Gson对象 用于hashMap的转换
      Gson gson = new Gson();
        String s = gson.toJson(hashMap);
      //设置请求参数为utf-8 这里会遇到坑 虽然说上面设置了header的utf-8但是这里还是需要设置
则会乱码
        StringEntity se = new StringEntity(s, HTTP.UTF_8);
      //设置 text/json
        se.setContentType(CONTENT_TYPE_TEXT_JSON);
        //传入StringEntity对象进入httpPost中
      httpPost.setEntity(se);
        //发送请求
        CloseableHttpResponse response2 = client.execute(httpPost);
      //接收到响应体
        org.apache.http.HttpEntity entity2 = response2.getEntity();
      //转换为string 类型
        String string = EntityUtils.toString((HttpEntity) entity2, "UTF-8");
        System.err.println("返回"+string);
        //将string转换为Gson
      Gson gson = new Gson();
      //你自己需要创建对象接收 如果不想再创建实体类  就使用JsonObject进行解析
      PayBack pay = gson.fromJson(string,PayBack.class);
     if(StringUtiles.isEmpty(pay)){
        System.out.println("返回为空");
      }else{
            System.out.println("进行逻辑操作");
```

原文链接：java 发送 HttpClient 请求

```
    }
}catch (Exception e){
    e.printStackTrace();
}
```