<https://blog.csdn.net/yyq411377361/article/details/78617088>

转

# JAVA利用HttpClient进行POST请求（HTTPS）

2017年11月23日 17:26:13 [姚永强](https://me.csdn.net/yyq411377361) 阅读数：4461

目前，要为另一个项目提供接口，接口是用HTTP URL实现的，最初的想法是另一个项目用JQuery post进行请求。

但是，很可能另一个项目是部署在别的机器上，那么就存在跨域问题，而JQuery的post请求是不允许跨域的。

这时，就只能够用HttpClient包进行请求了，同时由于请求的URL是HTTPS的，为了避免需要证书，所以用一个类继承DefaultHttpClient类，忽略校验过程。

1.写一个SSLClient类，继承至HttpClient

[java] [view plain](http://blog.csdn.net/rongyongfeikai2/article/details/41659353/) [copy](http://blog.csdn.net/rongyongfeikai2/article/details/41659353/)

1. **import** java.security.cert.CertificateException;
2. **import** java.security.cert.X509Certificate;
3. **import** javax.net.ssl.SSLContext;
4. **import** javax.net.ssl.TrustManager;
5. **import** javax.net.ssl.X509TrustManager;
6. **import** org.apache.http.conn.ClientConnectionManager;
7. **import** org.apache.http.conn.scheme.Scheme;
8. **import** org.apache.http.conn.scheme.SchemeRegistry;
9. **import** org.apache.http.conn.ssl.SSLSocketFactory;
10. **import** org.apache.http.impl.client.DefaultHttpClient;
11. //用于进行Https请求的HttpClient
12. **public** **class** SSLClient **extends** DefaultHttpClient{
13. **public** SSLClient() **throws** Exception{
14. **super**();
15. SSLContext ctx = SSLContext.getInstance("TLS");
16. X509TrustManager tm = **new** X509TrustManager() {
17. @Override
18. **public** **void** checkClientTrusted(X509Certificate[] chain,
19. String authType) **throws** CertificateException {
20. }
21. @Override
22. **public** **void** checkServerTrusted(X509Certificate[] chain,
23. String authType) **throws** CertificateException {
24. }
25. @Override
26. **public** X509Certificate[] getAcceptedIssuers() {
27. **return** **null**;
28. }
29. };
30. ctx.init(**null**, **new** TrustManager[]{tm}, **null**);
31. SSLSocketFactory ssf = **new** SSLSocketFactory(ctx,SSLSocketFactory.ALLOW\_ALL\_HOSTNAME\_VERIFIER);
32. ClientConnectionManager ccm = **this**.getConnectionManager();
33. SchemeRegistry sr = ccm.getSchemeRegistry();
34. sr.register(**new** Scheme("https", 443, ssf));
35. }
36. }

2.写一个利用HttpClient发送post请求的类

[java] [view plain](http://blog.csdn.net/rongyongfeikai2/article/details/41659353/) [copy](http://blog.csdn.net/rongyongfeikai2/article/details/41659353/)

1. **import** java.util.ArrayList;
2. **import** java.util.Iterator;
3. **import** java.util.List;
4. **import** java.util.Map;
5. **import** java.util.Map.Entry;
6. **import** org.apache.http.HttpEntity;
7. **import** org.apache.http.HttpResponse;
8. **import** org.apache.http.NameValuePair;
9. **import** org.apache.http.client.HttpClient;
10. **import** org.apache.http.client.entity.UrlEncodedFormEntity;
11. **import** org.apache.http.client.methods.HttpPost;
12. **import** org.apache.http.message.BasicNameValuePair;
13. **import** org.apache.http.util.EntityUtils;
14. /\*
15. \* 利用HttpClient进行post请求的工具类
16. \*/
17. **public** **class** HttpClientUtil {
18. **public** String doPost(String url,Map<String,String> map,String charset){
19. HttpClient httpClient = **null**;
20. HttpPost httpPost = **null**;
21. String result = **null**;
22. **try**{
23. httpClient = **new** SSLClient();
24. httpPost = **new** HttpPost(url);
25. //设置参数
26. List<NameValuePair> list = **new** ArrayList<NameValuePair>();
27. Iterator iterator = map.entrySet().iterator();
28. **while**(iterator.hasNext()){
29. Entry<String,String> elem = (Entry<String, String>) iterator.next();
30. list.add(**new** BasicNameValuePair(elem.getKey(),elem.getValue()));
31. }
32. **if**(list.size() > 0){
33. UrlEncodedFormEntity entity = **new** UrlEncodedFormEntity(list,charset);
34. httpPost.setEntity(entity);
35. }
36. HttpResponse response = httpClient.execute(httpPost);
37. **if**(response != **null**){
38. HttpEntity resEntity = response.getEntity();
39. **if**(resEntity != **null**){
40. result = EntityUtils.toString(resEntity,charset);
41. }
42. }
43. }**catch**(Exception ex){
44. ex.printStackTrace();
45. }
46. **return** result;
47. }
48. }

3.调用post请求的测试代码

[java] [view plain](http://blog.csdn.net/rongyongfeikai2/article/details/41659353/) [copy](http://blog.csdn.net/rongyongfeikai2/article/details/41659353/)

1. **import** java.util.HashMap;
2. **import** java.util.Map;
3. //对接口进行测试
4. **public** **class** TestMain {
5. **private** String url = "https://192.168.1.101/";
6. **private** String charset = "utf-8";
7. **private** HttpClientUtil httpClientUtil = **null**;
9. **public** TestMain(){
10. httpClientUtil = **new** HttpClientUtil();
11. }
13. **public** **void** test(){
14. String httpOrgCreateTest = url + "httpOrg/create";
15. Map<String,String> createMap = **new** HashMap<String,String>();
16. createMap.put("authuser","\*\*\*\*\*");
17. createMap.put("authpass","\*\*\*\*\*");
18. createMap.put("orgkey","\*\*\*\*");
19. createMap.put("orgname","\*\*\*\*");
20. String httpOrgCreateTestRtn = httpClientUtil.doPost(httpOrgCreateTest,createMap,charset);
21. System.out.println("result:"+httpOrgCreateTestRtn);
22. }
24. **public** **static** **void** main(String[] args){
25. TestMain main = **new** TestMain();
26. main.test();
27. }
28. }

httpClient4.2的jar包下载路径：http://download.csdn.net/detail/hqmryang/4582440#comment