1. **java自带的方法**
2. /\*\*
3. \* Http get请求
4. \* @param httpurl
5. \* @return
6. \* @throws IOException
7. \*/
8. **public** String sendGet(String httpurl) **throws** IOException {
9. URL url = **new** URL(httpurl);
10. HttpURLConnection conn = (HttpURLConnection) url.openConnection();
11. //设置连接方式
12. conn.setRequestMethod("GET");
13. //设置主机连接时间超时时间3000毫秒
14. conn.setConnectTimeout(3000);
15. //设置读取远程返回数据的时间3000毫秒
16. conn.setReadTimeout(3000);
18. //发送请求
19. conn.connect();
20. //获取输入流
21. InputStream is = conn.getInputStream();
22. //封装输入流
23. BufferedReader br = **new** BufferedReader(**new** InputStreamReader(is, "UTF-8"));
24. //接收读取数据
25. StringBuffer sb = **new** StringBuffer();
27. String line = ;
28. **while**((line = br.readLine())!=) {
29. sb.append(line);
30. sb.append("\r\n");
31. }
32. **if**(!=br) {
33. br.close();
34. }
35. **if**(!=is) {
36. is.close();
37. }
38. //关闭连接
39. conn.disconnect();
40. **return** sb.toString();
41. }
43. /\*\*
44. \* Http post请求
45. \* @throws IOException
46. \*/
47. **public** String doPost(String httpurl, Map<Object, Object> param) **throws** IOException {
48. String jsonparam = JSON.toString(param);
50. URL url = **new** URL(httpurl);
51. //获取httpurlConnection连接
52. HttpURLConnection conn = (HttpURLConnection) url.openConnection();
53. //设置读取超时
54. conn.setConnectTimeout(3000);
55. //设置读取超时
56. conn.setReadTimeout(3000);
58. //传送数据
59. conn.setDoOutput(**true**);
60. //读取数据
61. conn.setDoInput(**true**);
62. //设置请求方式
63. conn.setRequestMethod("POST");
64. //设置传入参数格式
65. conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
66. // 设置鉴权信息：Authorization: Bearer da3efcbf-0845-4fe3-8aba-ee040be542c0
67. conn.setRequestProperty("Authorization", "Bearer da3efcbf-0845-4fe3-8aba-ee040be542c0");
69. //获取输出流
70. OutputStream os = conn.getOutputStream();
71. //输出数据
72. os.write(jsonparam.getBytes());
73. //获取输入流
74. InputStream is = conn.getInputStream();
75. //封装输入流
76. BufferedReader br = **new** BufferedReader(**new** InputStreamReader(is,"UTF-8"));
78. StringBuffer sb = **new** StringBuffer();
79. String line = ;
81. **while**((line = br.readLine())!=) {
82. sb.append(line);
83. sb.append("\r\n");
84. }
85. **if**( != br) {
86. br.close();
87. }
88. **if**( != is) {
89. is.close();
90. }
91. **if**( != os) {
92. os.close();
93. }
94. //关闭连接
95. conn.disconnect();
97. **return** sb.toString();
98. }

# **Apache的HttpClient例子**

1. /\*\*
2. \* 发送HttpGet请求
3. \* @param url
4. \* @return
5. \*/
6. **public** **static** String sendGet(String url) {
7. //1.获得一个httpclient对象
8. CloseableHttpClient httpclient = HttpClients.createDefault();
9. //2.生成一个get请求
10. HttpGet httpget = **new** HttpGet(url);
11. CloseableHttpResponse response = ;
12. **try** {
13. //3.执行get请求并返回结果
14. response = httpclient.execute(httpget);
15. } **catch** (IOException e1) {
16. e1.printStackTrace();
17. }
18. String result = ;
19. **try** {
20. //4.处理结果，这里将结果返回为字符串
21. HttpEntity entity = response.getEntity();
22. **if** (entity != ) {
23. result = EntityUtils.toString(entity);
24. }
25. } **catch** (ParseException | IOException e) {
26. e.printStackTrace();
27. } **finally** {
28. **try** {
29. response.close();
30. } **catch** (IOException e) {
31. e.printStackTrace();
32. }
33. }
34. **return** result;
35. }

38. /\*\*
39. \* 发送HttpPost请求，参数为map
40. \* @param url
41. \* @param map
42. \* @return
43. \*/
44. **public** **static** String sendPost(String url, Map<String, String> map) {
45. CloseableHttpClient httpclient = HttpClients.createDefault();
46. List<NameValuePair> formparams = **new** ArrayList<NameValuePair>();
47. **for** (Map.Entry<String, String> entry : map.entrySet()) {
48. //给参数赋值
49. formparams.add(**new** BasicNameValuePair(entry.getKey(), entry.getValue()));
50. }
51. UrlEncodedFormEntity entity = **new** UrlEncodedFormEntity(formparams, Consts.UTF\_8);
52. HttpPost httppost = **new** HttpPost(url);
53. httppost.setEntity(entity);
54. CloseableHttpResponse response = ;
55. **try** {
56. response = httpclient.execute(httppost);
57. } **catch** (IOException e) {
58. e.printStackTrace();
59. }
60. HttpEntity entity1 = response.getEntity();
61. String result = ;
62. **try** {
63. result = EntityUtils.toString(entity1);
64. } **catch** (ParseException | IOException e) {
65. e.printStackTrace();
66. }
67. **return** result;
68. }