



链滴

java 发送 http 请求工具类

作者: [Pirates](#)

原文链接: <https://ld246.com/article/1602816736794>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

import com.alibaba.fastjson.JSONObject;
import lombok.extern.slf4j.Slf4j;
import org.apache.http.HttpEntity;
import org.apache.http.HttpStatus;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

import java.io.IOException;

/**
 * <h3>jeecg-boot-parent</h3>
 * <p>发送请求</p>
 *
 * @author : LiYu
 * @date : 2020-10-15 15:13
 **/
@Slf4j
public class HttpClientUtils {

    private static RequestConfig requestConfig = null;

    static {
        // 设置请求和传输超时时间
        requestConfig = RequestConfig.custom().setSocketTimeout(2000).setConnectTimeout(2000).build();
    }

    /**
     * post请求传输json参数
     *
     * @param url
     *         url地址
     * @param jsonParam
     *         参数
     * @return
     */
    public static JSONObject httpPost(String url, JSONObject jsonParam) {
        // post请求返回结果
        CloseableHttpClient httpClient = HttpClients.createDefault();
        JSONObject jsonResult = null;
        HttpPost httpPost = new HttpPost(url);
        // 设置请求和传输超时时间
        httpPost.setConfig(requestConfig);
        try {
            if (null != jsonParam) {
                // 解决中文乱码问题
                StringEntity entity = new StringEntity(jsonParam.toString(), "utf-8");
                entity.setContentEncoding("UTF-8");
            }
        }
    }
}

```

```

        entity.setContentType("application/json");
        httpPost.setEntity(entity);
    }
    CloseableHttpResponse result = httpClient.execute(httpPost);
    // 请求发送成功, 并得到响应
    if (result.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
        String str = "";
        try {
            // 读取服务器返回过来的json字符串数据
            str = EntityUtils.toString(result.getEntity(), "utf-8");
            // 把json字符串转换成json对象
            jsonResult = JSONObject.parseObject(str);
        } catch (Exception e) {
            log.error("post请求提交失败:" + url, e);
        }
    }
} catch (IOException e) {
    log.error("post请求提交失败:" + url, e);
} finally {
    httpPost.releaseConnection();
}
}
return jsonResult;
}

/**
 * post请求传输String参数 例如: name=Jack&sex=1&type=2
 * Content-type:application/x-www-form-urlencoded
 *
 * @param url
 *         url地址
 * @param strParam
 *         参数
 * @return
 */
public static JSONObject httpPost(String url, String strParam) {
    // post请求返回结果
    CloseableHttpClient httpClient = HttpClients.createDefault();
    JSONObject jsonResult = null;
    HttpPost httpPost = new HttpPost(url);
    httpPost.setConfig(requestConfig);
    try {
        if (null != strParam) {
            // 解决中文乱码问题
            StringEntity entity = new StringEntity(strParam, "utf-8");
            entity.setContentEncoding("UTF-8");
            entity.setContentType("application/x-www-form-urlencoded");
            httpPost.setEntity(entity);
        }
        CloseableHttpResponse result = httpClient.execute(httpPost);
        // 请求发送成功, 并得到响应
        if (result.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
            String str = "";
            try {
                // 读取服务器返回过来的json字符串数据

```

```

        str = EntityUtils.toString(result.getEntity(), "utf-8");
        // 把json字符串转换成json对象
        jsonResult = JSONObject.parseObject(str);
    } catch (Exception e) {
        log.error("post请求提交失败:" + url, e);
    }
}
} catch (IOException e) {
    log.error("post请求提交失败:" + url, e);
} finally {
    httpPost.releaseConnection();
}
}
return jsonResult;
}

/**
 * 发送get请求
 *
 * @param url
 *         路径
 * @return
 */
public static JSONObject httpGet(String url) {
    // get请求返回结果
    JSONObject jsonResult = null;
    CloseableHttpClient client = HttpClients.createDefault();
    // 发送get请求
    HttpGet request = new HttpGet(url);
    request.setConfig(requestConfig);
    try {
        CloseableHttpResponse response = client.execute(request);

        // 请求发送成功, 并得到响应
        if (response.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
            // 读取服务器返回过来的json字符串数据
            HttpEntity entity = response.getEntity();
            String strResult = EntityUtils.toString(entity, "utf-8");
            // 把json字符串转换成json对象
            jsonResult = JSONObject.parseObject(strResult);
        } else {
            log.error("get请求提交失败:" + url);
        }
    } catch (IOException e) {
        log.error("get请求提交失败:" + url, e);
    } finally {
        request.releaseConnection();
    }
    return jsonResult;
}
}
}

```