```
            // 设置连接时长，读取超时时间，以及其他必要参数设置
                    okHttpClient = new OkHttpClient.Builder()
                            .connectTimeout(15, TimeUnit.SECONDS)
                            .writeTimeout(20, TimeUnit.SECONDS)
                            .readTimeout(20, TimeUnit.SECONDS)
.sslSocketFactory(createSSLSocketFactory(trustManagers), (X509TrustM
anager) trustManagers[0])
                            .hostnameVerifier((hostname, session) -> true)
                            .retryOnConnectionFailure(true)
                            .build();
                    addHeader("User-Agent", "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/78.0.3904.108 Safari/537.36");
                }
            }
        }
    }

    // 发起请求
    public OkHttpUtil initPost() {
        RequestBody requestBody =
RequestBody.create(MediaType.parse("application/json; charset=utf-8"),
!param.equals("") ? param : "");
        request = new Request.Builder().post(requestBody).url(url);
        return this;
    }

    /**
     * @Description:同步请求
     * @Author: yn
     * @Date: 2023-01-04 18:06
     * @return: java.lang.String
     **/
    public String sync() {
        setHeader(request);
        try {
            Response result =
okHttpClient.newCall(request.build()).execute();
            if (Objects.nonNull(result)) {
                return result.body().string();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return "请求失败";
```

```
    }

    /**
     * @Description:异步请求，有返回值
     * @Author: yn
     * @Date: 2023-01-04 18:05
     * @return: java.lang.String
     **/
    public String async() {
        StringBuffer buffer = new StringBuffer();
        setHeader(request);
        okHttpClient.newCall(request.build()).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                buffer.append("请求出错").append(e.getMessage());
            }

            @Override
            public void onResponse(Call call, Respo onse response) throws
IOException {
                if (Objects.nonNull(response.body())) {
                    buffer.append(response.body().string());
                    getSemaphore().release();
                }
            }
        });
```

```
结论与优化
-----


虽然最后发现问题不大，由OkHttp引起，最终是 **OkHttp+Semaphore** 结合使用问题。


我们可以在此基础上再进行下优化，**假如不用Semaphore** 也照样可以实现在异步线程中做到一个同步的效果。
刚才也举例了，可以用**CompletableFuture** 实现



```
public void async(BiConsumer<Call, String> onSuccess,
BiConsumer<Call, String> onFailure) {
    setHeader(request);
    okHttpClient.newCall(request.build()).enqueue(new Callback() {
```
```

```
        @Override
        public void onFailure(Call call, IOException e) {
            onFailure.accept(call, e.getMessage());
        }
        @Override
        public void onResponse(Call call, Response response) throws
IOException {
            try (ResponseBody responseBody = response.body()) {
                if (Objects.nonNull(responseBody)) {
                    onSuccess.accept(call, responseBody.string());
                }
            }
        }
    });
}

```

用 **BiConsumer** 函数做回调接收。

```
@RequestMapping("/test1")
public ResponseBO test1(){

    CompletableFuture<ResponseBO> future = new
CompletableFuture<>();
    JSONObject jsonObject = new JSONObject();
    jsonObject.put("userId","123456");
OkHttpUtil.builder().url("https://api.example.com/data").addParam(jsonO
bject.toJSONString()).initPost()
    .async((call, response) -> {
        // 处理成功响应
        logger.info("数据推送成功，url -> {}，req -> {}，res -> {}",
"127.0.0.1" , jsonObject.toJSONString(), response);
        future.complete(ResponseBO.responseOK());
    }, (call, error) -> {
        // 处理失败响应
        logger.info("数据推送失败，url -> {}，req -> {}，res -> {}",
"127.0.0.1", jsonObject.toJSONString(), error);
        future.complete(ResponseBO.responseFail("数据处理失败"));
    });

    try {
        return future.join();
    } catch (Exception e) {
        e.getMessage();
```

```
            // 处理异常情况
            return ResponseBO.responseFail(”数据推送失败”);
        }
    }

```
原文链接: https://juejin.cn/post/7374045077452996635