

把长活儿搬出前台

网站服务进程只该收单，不该在自己身体里干几个小时的重活

王建硕

费曼式写法 · 由多个 AI 代理撰写与互相审校

第一章 · 柜台不烤全牛

餐馆的柜台小姐只有一件事要做：记下你要什么，收你的钱，给你一张小票。她不烤肉，不洗碗，更不会因为你要了一桌酒席，就挽起袖子亲自去后厨烤一头全牛。

网站的服务进程也该这样。它的活儿是**收单**：接住一个请求，看明白要干什么，然后尽快回话。这个「尽快」通常是几十到几百毫秒。

可偏偏有程序员让柜台小姐去烤全牛。用户点一下「生成全年报表」，服务进程就真的埋头算八个小时——连接一直挂着，内存一直占着。更要命的是，柜台小姐就那么几个。三个在烤牛，第四个客人进门，连碗牛肉面都点不了。再过一会儿，用户等不及按了刷新，牛还在烤，单子又下了一遍。

大白话

大白话：服务进程是柜台，不是后厨。它的天职是快点回话，不是把活儿干完。活儿干得慢不丢人，让后面所有人跟着排队才丢人。

那几小时的活儿谁来干？答案是：搬出前台，交给后厨。怎么交，是下一章的事。

第二章 · 小票与后厨

搬法很朴素，学快餐店。

用户点「生成报表」，服务进程只做两件事：把这笔活儿连同参数**记进一张单子**，然后立刻回用户一句「已受理，单号是 8848」。前后几百毫秒，柜台始终是空的。

那张单子放进的地方叫队列——一个按先来后到排队的本子，谁都能往里添单子，但单子在里面稳稳躺着，不会丢。后厨另有专门的工人进程——一批只管干活、不接待用户的程序——轮流从本子上撕单子，慢慢烤牛。烤八小时是它的事，烤完把结果存好，在单子上盖个「完成」。

大白话

大白话：前台和后厨之间隔着一本单子。前台只负责写单子、给小票；后厨只负责看单子、干活。两边谁也不等谁。

好处是连锁的：用户刷新页面，顶多再下一张同样的单，去重一下就行；活儿太多，就多开几个工人，柜台一个都不用加；工人半夜挂了，单子还在本子上，捡起来重做就是。

只剩一个问题：用户拿着小票，怎么知道他的牛烤好了？

第三章 · 凭票取货

烤好的牛不会自己跑到桌上，得有取货的方式。常见三种，都是小事：

- **自己来问**：页面隔几秒查一次单号，柜台只回三个字——「在烤呢」「烤好了」。查一万次也不累，因为回话永远便宜。
- **广播叫你**：烤好了，柜台的取餐屏上跳出你的号，网页上弹个通知。你不用问，号到了自然响。
- **留个地址**：填个邮箱或回调地址，做完了系统寄给你。慢活儿最合适，你甚至不用守在店里。

回头看，这套「收单、排队、后厨、凭票取货」不止用在报表上。视频转码是它，AI 画图是它，发十万封邮件也是它——凡是干起来以小时计、用户却指望几秒钟有回音的活儿，都该走这条路。

大白话

大白话：前台的从容不是因为它能干，而是因为它从不硬扛。把长活儿搬出去，它才有余力对每个人都快点回话。

所以检查一个网站健不健康，我常常只看一眼：它的柜台小姐，有没有在烤全牛。