



链滴

# 技术日常 | 还在增删改查? 如何在工作中提升技术水平

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1571988286601>

来源网站: 链滴

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



## 开篇词

互联网行业996已是见怪不怪的问题，在这背后大家常常讨论的是被老板压榨的时间，导致自己没有法看书学习，扩宽自己的知识面。不管这工作外的8小时是不是用来学习，总之就是自己的时间被占了，对于这种无法更改的事实，抱怨不是常法，最近在读一本书《活着的意义》，里面有这么一段话：

苦难之意义，我们一定不能忘记，即使在看似毫无希望的境地，即使面对无可改变的厄运，人们也能到生命之意义。

同样最近大火的国漫电影《哪吒之魔童降世》里面燃爆全场的一句话“我命由我不由天”。对于每天96的重复式机械工作，我们还是可以提升自己的技术能力的。

## 提出问题时一定要带着解决方案

这点在日常工作中很重要，我们会抱怨很多很多的事情，向家人和朋友抱怨没什么，但是在公司向上老板抱怨的时候就要注意了，如标题所示，抱怨时，带上解决方案，常见的如：

dubbo 2.5.X 这些老版本太low了，我觉得我们可以尝试下新的dubbo 2.7.X，我已经调研写了一个demo，与老版相比，有如下好处：巴拉巴拉巴拉

mybatis还在用xml配置有点太老了，现在spring都推荐使用注解来配置，mybatis新版也支持注解去置mapper和sql语句，我们可以尝试下，对已有的代码，可以做以下迁移过渡方案：巴拉巴拉巴拉

上面这些是我们公司的真实案例，同样的，对应的改变，我们的新项目，dubbo 都升级到了2.7.X，mybatis 用上了注解的方式，还有很多别的变更就不一一细述了，大家可以审视下自己负责的功能或者模块，看看应的问题有什么样的解决方案

## 对分给自己的任务多想几种方案

行业有句话：面试造航母，入职拧螺丝，嘴里喊着高可用，高并发，天天写着增删改查。

做为一个两年开发,入职的时候,我也确实天天写增删改查,分给我的第一个任务是,建张医生的常用药品,然后就是根据id增删改查.就这样我写完上线之后还有bug,然后查bug的时候肯定要打日志,然后学到了 `ogger.info("xxxxx {}",paramA)`. 这样打印参数,而不是用+拼接的方式.然后就顺手把项目里面看到的改了下.

## 案例,接口计时

然后刚入职的时候确实技术比较渣,写的那个增删改查接口很慢,但是慢到什么程度没有一个定论,就是试反应慢,产品也反应慢,那就记录下时间,看看到底慢到什么程序,这时候就有两种方案选择了,一是在法入口和出方法的时候,用

```
Long start = System.nanoTime();
doMethod();
Long end = System.nanoTime();
Long time = end-start;
```

这种当然是可以实现需求的,当然我们也可以切面去写,用 Spring AOP去切到那个方法,然后在around里面记录时间,然后这个时间可以新开一个线程去做持久化. 在新开线程的时候,我们可以用 `Executors.newCachedThreadPool()`去构建线程池,然后这认识到这种方法创建线程池会导致线程数过多,可以再开自定义一个线程池实现,这样一步步的发展下

## 案例,if-else的改造

我们公司的图片等一些资是存在的oss上的,但是比较坑的是,测试环境是阿里云的OSS,线上环境是华云的OBS,这就要求在调用dubbo接口的时候,不同的环境不同的实现,这时候也是有多种方案实现的.

第一种就是我们常用的if-else ,

```
if("huaweiyun"){
obsClient.get()
}else if("aliyun"){
ossClient.get()
}
```

这种写法可以实现要求,但是我们又常说,我们会用策略模式消灭代码中的if-else,那这个要怎么改呢? 代码及思路如下,两个实现类都实现同一个接口,接口中有一个 `getObject()`方法.

```
AliClient implement ResourceClient{
    OssClient client;
    getObject(){
        return client.get();
    }
}
HwClient implement ResourceClient{
    ObsClient client;
    getObject(){
        return client.get();
    }
}
```

```
ResourceClient client = getResourceClient(env);
client.getObject();
```

有些小伙伴要说了,这里面只不过是把if-else转移了下,没有放到getObject里面,放到了getResourceClient(env)里面,对的,但肯定是有解决方案的.那就是

```
getResourceClient(String env){  
return applicationContext.getBean(env);  
}
```

至于这个env和beanName在哪里对应,我们可以配置在数据库里面,redis里面,apollo里面,都可以做到上面这样之后,我们会发现,不管是在哪个环境,这两个实现类都要初始化,但实际上我们在线上环境是只需要初始化华为云的ObsClient,在测试环境只需要初始化阿里云的OssClient. 想做到这样,就要初始化Bean的权利从Spring手中接管过来,这时候的方案可以是在spring初始化Bean的时候加入自己逻辑.

如上两个例子,一个是简单的增删改查计时,一个是if-else不同实现类,对每一个需求,我们都有多个方案完成,我们通常会选用自己最熟悉的方案,然后每天抱怨着接触的都是这些一样的东西,工作学不到东西.

## 最后说两句

通过前面的案例,我们不难做以下总结:

一是要多思考,思考问题有没有解决方案,思考问题有没有多个解决方案

二是要多实践,实践我们想的解决方案能否生效,在实践中找到最适用当前场景的解决方案

大家有什么想法欢迎随时留言和小刀交流