



链滴

2016 回顾

作者: [tiangao](#)

原文链接: <https://ld246.com/article/1485054715522>

来源网站: [链滴](#)

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

看到别人发总结我也忍不住来一发，确实这一年经历的东西还挺多的。

本来想写的东西还挺多，转念一想记流水账有毛意思，这个快节奏的时代谁愿意看又臭又长的东西，且我文笔又不怎么样，所以还是直接上总结吧。

心态变化

之前，面对冗长繁杂代码心里非常不愿意看，怎么写得这么烂，垃圾啊；之后，代码烂一定有其原因能改能重构就改，改不动就指点打点的修问题，抱怨和骂没有什么卵用，完成任务为第一要务。

之前，读书一晃而过，看完就下一本，觉得了解了就行；之后，看书得思考，得总结，最好是能记下一点东西，记忆总是靠不住的。

之前，给我任务就开始写，怎么想的就怎么实现；之后，接到任务，先搜索一遍有没有成熟的解决方案或方法，少造轮子，能避免很多坑。

之前，给我分配任务我就干活，没任务时就闲着，也不愿意主动研究工作相关的东西，觉得工作挺无的；之后，工作也是需要很多的思考，有些东西需要技术去改，有些东西通过业务解决更直接有效，且在用技术的解决过程中也要多思考方法，我们处在一个很好的实在，工具非常丰富，花时间学习用些工具往往能省下很多时间。

技术成长

Java

从 JDK1.6 + SSH 到 JDK8 + SpringBoot

- 创建项目 so easy，几分钟就创建一个项目，立马可以写业务
- 用 JPA 写数据层方法，常用的都不用写代码，命名即实现
- 使用 Hibernate 的 hbm2ddl 直接生成数据库表，原来 Java 也能像 .Net MVC 一样代码即数据库表
- 用 SpringSecurity 管理用户权限和接口权限是如此的方便，还不用担心问题
- 单元测试，真的能有效提高代码质量，写的时候就能避免很多 bug

试用了 **SpringCloud** 全家桶，体会微服务的感觉

用 Jfinal 完成了一个小外包项目，这个框架代码不多，非常易用，不得不说设计很精巧，做个小型 we 项目绝对配的上极速开发的称谓。

Golang

从入门到实用

写过一个项目才体会到 golang 是如何用的，以及如何在运行时找错误。

不过写的还不是很多，也就刚刚入门的水平吧，一直想好好看下 gogs 的源码

JavaScript

最后一个月里参与了一个 react + redux + react-redux-router 为核心的项目，写了一些模块

感叹前端之复杂不亚于后端

工具进化

从 Eclipse 过渡到 IntelliJ IDEA

- 智能提示太棒了
- Debug 在变量旁边显示值真爽，原来还能 remote debug
- 重构真轻松

会用 Mac OS 了

- **brew & brew cask** 安装应用和服务真轻松
- **Alfred** go to anything 的感觉如此之畅快
- **iterm2 + zsh** 爱上命令行操作，轻松自如
- **多指手势 + 多桌面** 切换自如，触摸板可以完全替代鼠标

从 Sublime 到 Sublime + vscode

- sublime 用来看代码和文件不错，速度快
- vscode 插件丰富，简单易用，写前端写 go 代码体验都很好

从 SVN 到 **GIT**

- 这个见仁见智吧，git 确实很好用，多人合作体验很好
- svn 我觉得也有适用场合，不过我们现在只用 git 了

使用 Docker

- docker 常用命令的使用，了解整个的体系结构和运作方式
- 编写 Dockerfile，能打包应用到 docker 然后部署
- Java 和 golang 对于 docker api 的使用

vagrant & ansible

- **vagrant** 虚拟机管理工具，写好脚本直接生成虚拟机，要啥系统要几个统统搞定
- **ansible** Linux 远程部署工具，安装服务运行服务轻松方便
- 不过这里两个了解不深，只能说会用