



[回到首页](#)

芋道源码 —— 知识星球

我是一段不羁的公告！

记得给芬芳这 3 个项目加油，添加一个 STAR 噢。

<https://github.com/YunaiV/SpringBoot-Labs>

<https://github.com/YunaiV/oneMall>

<https://github.com/YunaiV/ruoyi-vue-pro>

2017-01-01

精尽学习指南

芬芳：这是第一个版本，后续会继续优化。嘿嘿。

后面，芬芳会把每个学习资料给补充进去。

1. 概述

无论我们是工作多久的工程师，我们总是在某一刻会大呼“学不动了学不动了”。

并且，我们是无可避免需要有大量加班，比较好的是 985，比较苦逼的是 996。一旦陷入其中，我们可能就忘却了学习这个事情。

另外，我们还会有一个困扰，技术栈这么多，我们应该怎么学习呢？

芬芳的想法是：

点
线
面
立体

什么是点？我们平时会在技术社区、公众号、博客上，学习一些技术，但是大家普遍会觉得比较琐碎。

什么是线？当我们学习了同一个技术的多个点之后，我们会逐渐形成这个知识线，对这个知识会相对有整体的认识。

什么是面？当我们学习了多个技术之后，我们会逐步自己的知识面，脑海里会有一颗若隐若现的知识树。

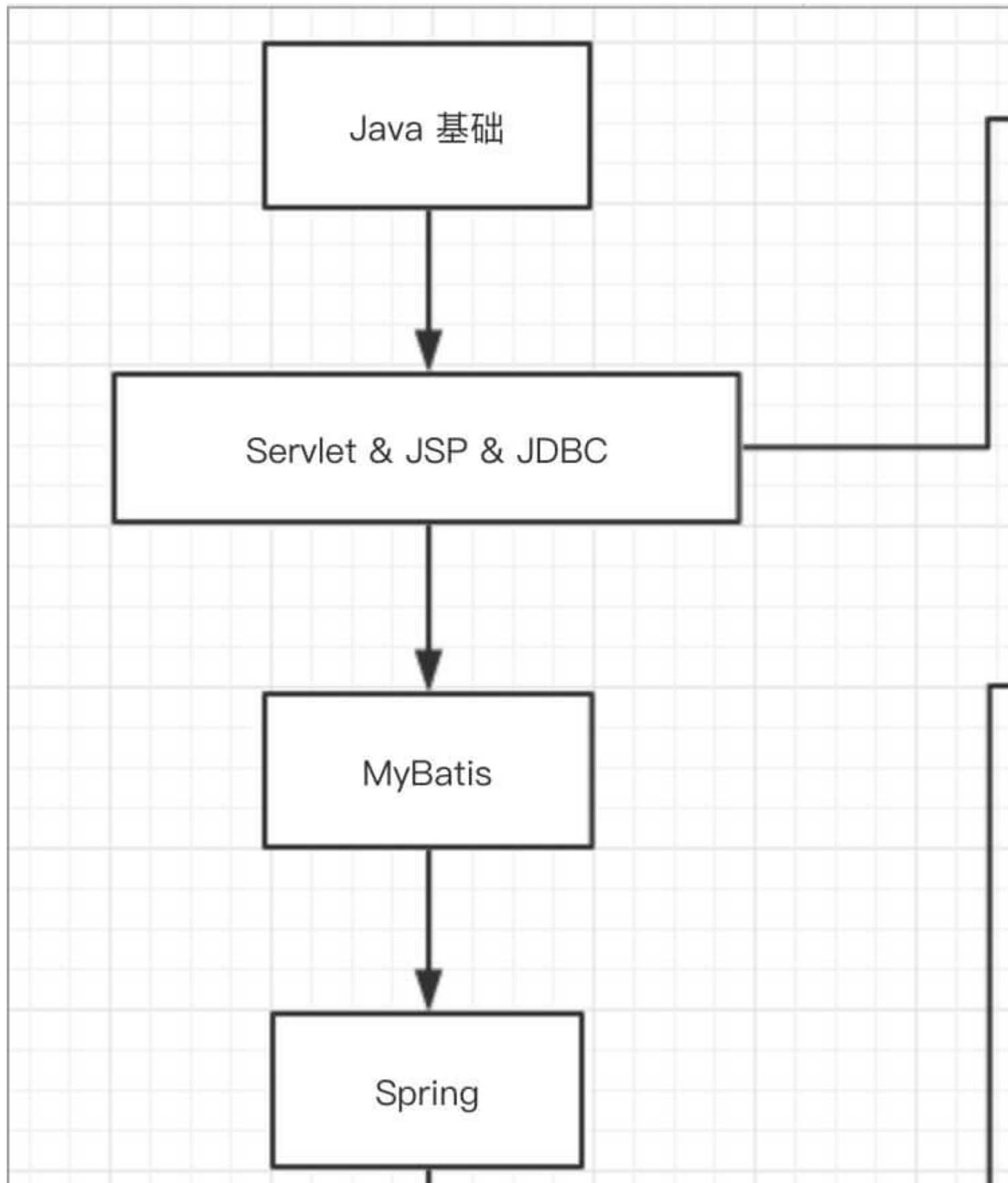
什么是立体？当我们形成自己的知识面之后，我们肯定会惊呼“老子会很多东西，但是好像没有什么擅长的”。有木有？哈哈哈。这个时候，我们需要针对自己感兴趣的技术栈，做深入的理解，慢慢形成某几个技术的深度，这样，我们就开始立体了。当然，一定是几个技术的深入，而不是方块，而是一个锥子，一锥刺骨的那种，杀手锏。

当然，本文暂时先对这个泛泛而谈，本文的重心在于如何先形成自己的“点 -> 线 -> 面”。我们先来看一个图：

一切的核心，都是为了实战。
入门，是为了让我们能够实战。
原理，是为了让我们更好的实战。
源码，是为了让我们更具象的理解原理，从而更好的实战。
所以，一切的核心，都是为了实战。

也因此，一个技术栈的学习的逐步顺序是：入门 => 实战 => 原理 => 源码。

那么，这个和本文又有什么关系呢？我们再来看一张图：



- 1、可能胖友会懵逼，卧槽，要学习的有这么多。
- 2、可能胖友会说，为什么没有 Dubbo、Zookeeper 等等
- 3、可能胖友会说，怎么没有 AI、大数据呢？

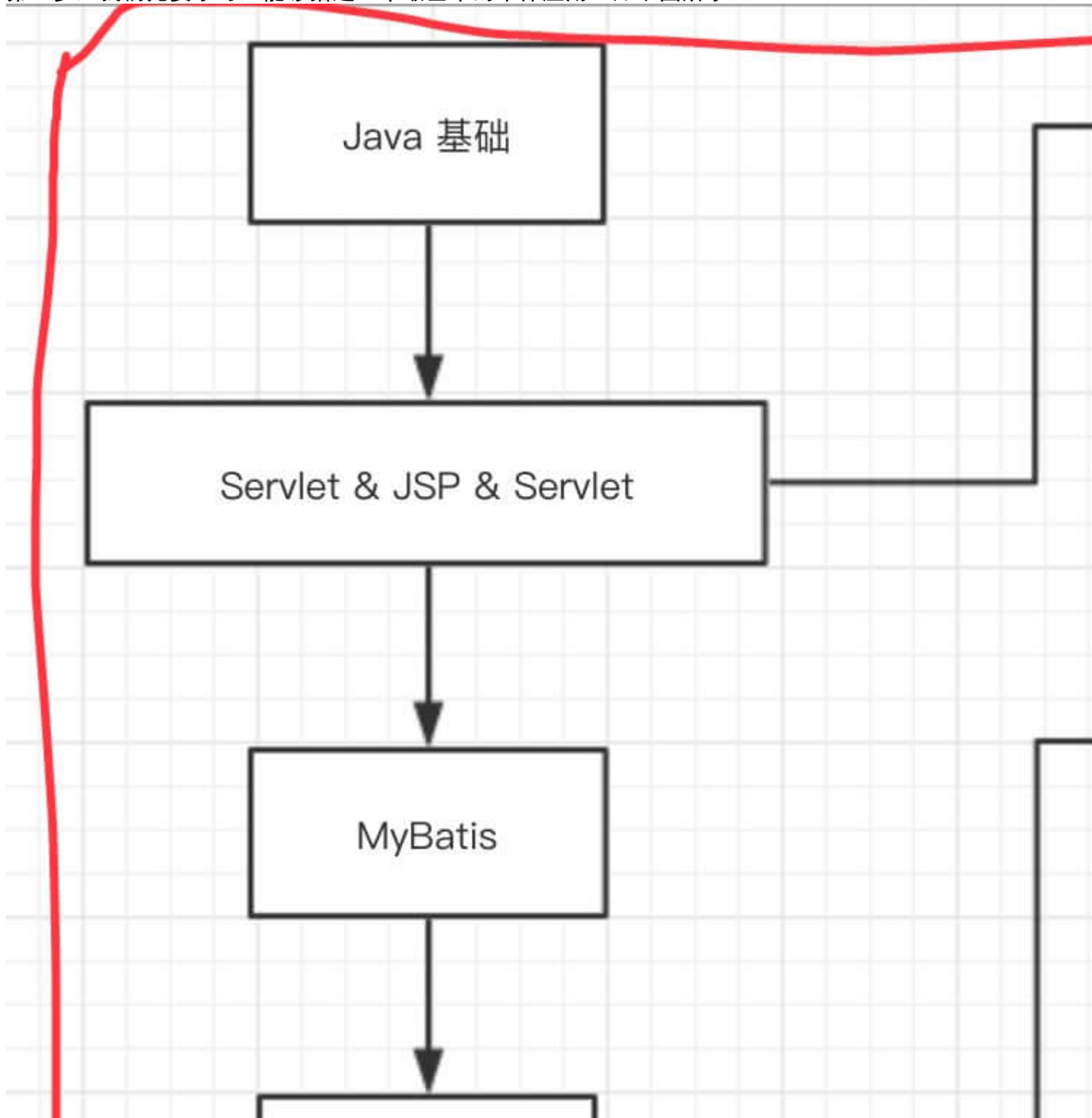
这个路线图，我们定位是让业务开发工程师，有一条相对明确的学习路线，最终实现搭建自己的知识体系。

2. 路线图

下面，我们来分块来看看路线图。

第一步，单体应用

第一步，我们先要学习，能够搭建一个最基本的单体应用。如下图所示：



在这个过程中，我建议观看视频。

Java

[《精尽 Java【基础】学习指南》](#) 入门即可。

芬芳当前看的马士兵。

Servlet & JSP & JDBC

这个的视频资源，自己找，芬芳当前看的马士兵。

MySQL

[《精尽 MySQL 学习指南》](#) 入门即可。

MyBatis

[《精尽 MyBatis 学习指南》](#) 入门即可。

Spring MVC

[《精尽 Spring MVC 学习指南》](#) 入门即可。

Spring Boot

[《精尽 Spring Boot 学习指南》](#)

看看尚硅谷的《Spring Boot 视频教程》是不错的。

Redis

[《精尽 Redis 学习指南》](#) 入门即可。

MongoDB

[《精尽 MongoDB 学习指南》](#) 入门即可。

Spring Security

TODD 待补充

Nginx

[《精尽 Nginx 学习指南》](#) 入门即可。

在这一步中，我们需要对每个技术点只要达到“入门”和“实战”两点。

第二步，微服务架构

第二步，我们要再学习，能够搭建一个微服务架构的应用。如下图所示：



在这个过程中，我建议以观看视频为主，同时能够认真看完 Spring Cloud 相关的书籍。

为什么要推荐开始看书？因为，到了这个阶段，我们要开始从被动的从视频学习，慢慢变成主动的观看书籍。

[《精尽 Spring Cloud 学习指南》](#)

推荐的书籍是，翟永超的 [《Spring Cloud微服务实战》](#)

Eureka
Hystrix
Zuul
Spring Cloud Config
RabbitMQ
Zipkin

那么，可能对微服务技术选型有一定的了解会问，为什么不是学习 Dubbo 呢？注意噢，我们是为了搭建知识体系，从这方面上来说，芳芳觉得 Spring Cloud 更适合达成这个目标，周边的书籍、资料更加完善。

当然，等未来回过头，芳芳还是建议大家去学习下 Dubbo 。

在这一步中，我们需要对每个技术点只要达到“入门”和“实战”和“原理”三点。

还有啊，其实每个类型的组件，还有其他的选择，胖友后续也要去学习下：

Eureka VS Zookeeper

TODO 待补充

Hystrix VS Sentinel

TODO 待补充

Zuul VS Spring Cloud Gateway

TODO 待补充

Spring Cloud Config VS Apollo

TODO 待补充

RabbitMQ VS RocketMQ VS Kafka

- [《精尽 RocketMQ 学习指南》](#)
- [《精尽 Kafka 学习指南》](#)
- [《精尽 RabbitMQ 学习指南》](#)

Zipkin VS SkyWalking

TODO 待补充

不要慌，知识这个东西，总是一通百通的。

还有噢，到了这个阶段的学习，我们要关注每个组件怎么实现高性能，怎么实现高可用。

第三步，其它中间件

第三步，我们还要学习，微服务架构需要的其他中间件。如下图所示：

【定时器】 Quartz



【搜索引擎】 Elasticsearch



【分库分表】 Sharding-Sphere



【分布式事务】 TCC-Transaction



【日志收集】 ELK