

# Go语言微服务框架

七米老师

<https://liwenzhou.com>



# 目录

- 框架的本质
- 如何选择框架
- 大厂现状
- 如何学习框架
- 微服务架构组件
- 常用微服务框架



**框架的本质是什么？**



# 框架的本质是什么？

## 脚手架

该有的都有  
减少重复工作  
方便干活（写业务代码）

## 标准化模板

项目目录一致  
项目风格一致  
依赖组件一致



# 为什么公司要求统一框架？

公司里面要推标准化的项目框架（脚手架），目录结构一致，代码风格一致，  
接手起来容易，  
维护起来省心，  
业务开发就只管写业务，你写的代码越少越不容易出问题。



# 选择框架的本质是什么？

省点心。

Java里面的Spring框架为什么那么火？

全世界都在用，你用到的组件都有现成的世界级轮子，拿来就能用。

不怕出问题，出了问题也不怕，随手一搜就能找到解决方案。



# 应该怎么选择框架呢？

哲学问题，对于不同的人有不同的选择。

如果是想搞技术或者为了搞技术，基本都自研。

如果是想学个框架找工作，那Go社区还真没有类似Spring那样的框架。

所以，还是看个人喜好！看看哪个框架用着顺手用哪个。

或者，让用哪个就用哪个。



# 大厂现状

就我知道的几个大厂内部使用的基本都是自研框架。

技术委员会推动+专业的基础架构团队开发维护。

- 提供基础组件轮子
- 提供稳定性保证
- 提供安全保证
- 提供与厂内基础设施联通性。

大厂的框架基本都有某些特殊性，可能并不适合厂外。



# 如何学习框架？

**学习框架的编程思想，而不是单纯学用法。**

这也是我个人不想讲那种比较重的框架的原因  
因为视频教程应该是最大化受众的，  
如果完全按照一套框架的模式走，受众就固定住了。

当然这么做利弊都有。

利：我以为的受众增加了

弊：为什么不用xxx



# 为什么不看好国内出现类似Spring的框架？

- 某大厂出了框架，其他大厂不会用
- 某小厂出了框架，大厂也看不上。
- 某人出了那种框架，那个人肯定会去上面的某一家公司。

绝不是因为国内程序员技术不行，我个人还是很尊敬和崇拜这些框架的开发者们。



# 微服务框架组件

日志

配置

数据库

消息队列

metrics

tracing

服务注册发现

依赖注入



# 社区常用框架

[go-kit](#) : go 微服务套件库

[go-micro](#) : 知名微服务框架, 插件理念, 不兼容更新

[go-kratos](#) : B站开源, v2版本

[kitex](#) : 字节开源, 自定义netpoll

[go-zero](#) : 好未来开源, 一站式微服务框架