

go-zero 开源项目成长史

万俊峰Kevin

About me

- go-zero 作者
- 阿里云MVP
- 腾讯云TVP
- ArchSummit 明星讲师
- GopherChina 主持人&金牌讲师
- 极客时间 Go 专题出品人
- 腾讯云开发者大会讲师

TOC

- go-zero 的由来
- go-zero 的现状
- go-zero 的成长阶段
- go-zero 推广思路
- go-zero 社区发展思考
- go-zero 的未来

go-zero 的由来

- 源起 2013
- 业务驱动、业务支撑
- 经历过的业务
- 经历过的 Go 发展阶段

go-zero 的现状

- star 增长趋势
- contributors
- 用户
 - 社区用户
 - 企业用户
- 质量标准
 - 需求
 - PR 标准

go-zero 数据统计

zeromicro/go-zero

Total stars

☆ 24,339

Avg. stars/day

23.79

Avg. commits/day

1.77

Avg. forks/day

3.31

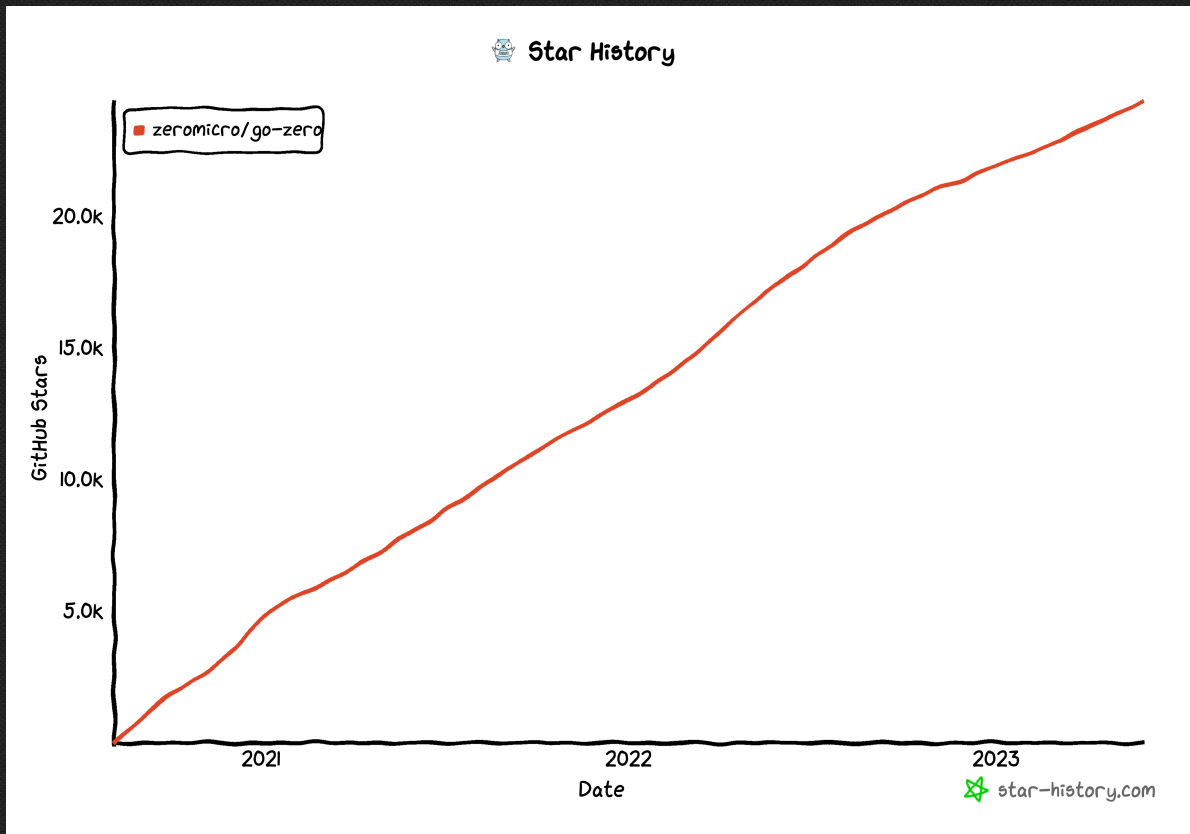
Avg. issues/day

1.23

Avg. pull requests/day

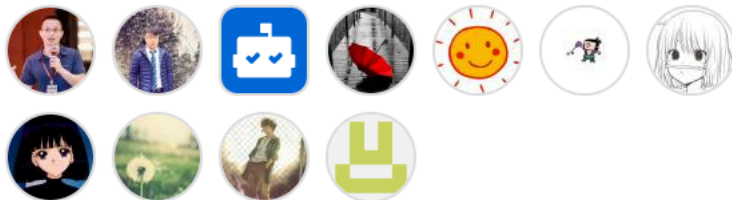
1.92

star 增长趋势



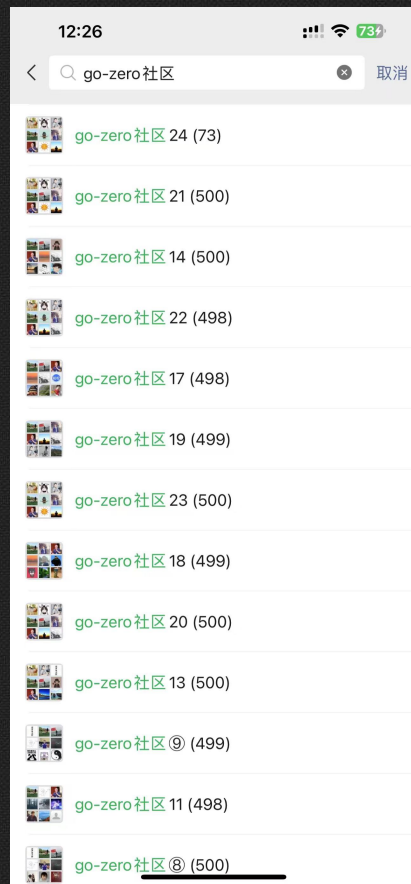
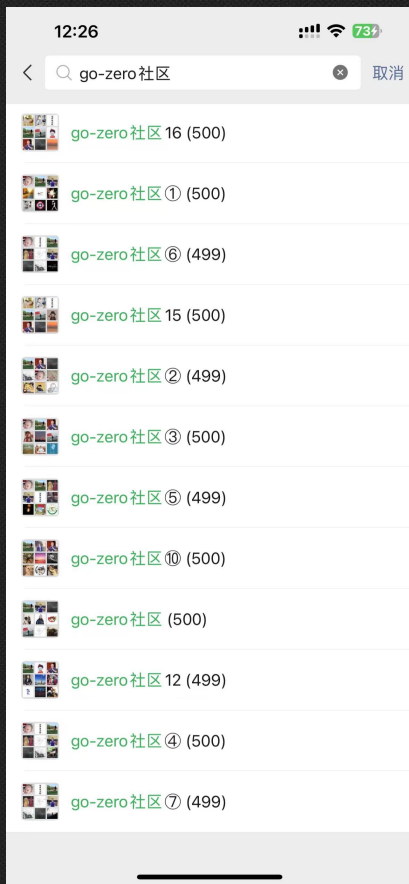
Contributors

Contributors 215



+ 204 contributors

社区用户



企业用户

晓黑板

TAL 好未来
TAL EDUCATION

七牛云

keep

天翼云

Lenovo 联想

51CTO

又拍云

YOOZOO
游族网络

POIZON

撸月狗

云犀

好特卖 HotMaxx

M+

千帆云
qianfan.com

XSHOPPY

三合通信
SAMHO TELECOM

释空

速优云

一起销售
17XIAOSHOU.CN

ZHONGKE
Life Service Corporation

IND@CHAT

数赞 参与者共赢

点购广场
DG-MALL.COM

VSPN

时代经纬科技
Era-Link Technology

33.复杂美
33.cn

URIT 优利特

智橙互动
Wisdom orange mall

鲸品云

都市到家 旺铺

馨科智科技

亿景智联
changjing.ai

Clobotics

SANGFOR
深信服科技

VictorySoft
胜软科技

高小鹿

openDapper

任维软件技术有限公司
Renwei Software Technology Co., Ltd

鼎翰文化
DINGHAN CULTURE

PuerHub

茄椒

叮当跳动
tinkdance.com

SimbaM

寻梦合伙人
Dream partner
安徽寻梦信息技术有限公司

代码质量

Report for github.com/zeromicro/go-zero

(v1.5.2)

A+ Excellent! Found **0** issues
across **597** files

go report: **A+** [Tweet](#)

Results

gofmt **100%**

go_vet **100%**

ineffassign **100%**

gocyclo **100%**

license **100%**

misspell **100%**

Last refresh: 3 days ago

微服务实践

zeromicro / go-zero / **P** master

Coverage [Flags](#) [Commits](#) [Pulls](#)

Branch Context

master ▼

Source: latest commit [452c9db](#)

Coverage on branch

 **93.74%**

15962 of 17028 lines covered

3 Months ▼ trend

+4.43%

Code tree

File list

Files ↑	Tracked lines	Covered	Partial	Missed	Coverage %
core	13465	12618	201	646	93.71%
gateway	269	224	9	36	83.27%
internal	124	113	4	7	91.13%
rest	1856	1780	21	55	95.91%
zrpc	1314	1227	10	77	93.38%

功能评估 & 代码合并

- 功能评估
 - 如非必要，勿增实体
 - MVP，最小化接口
 - 社区需求驱动
- PR要求
 - CI检测，格式、安全、单测
 - 修改或新增代码必须单测覆盖

go-zero 发展阶段

- 一年一万星
- 两年两万星
- 第三个年头

go-zero 开源伊始

- 开源 ≠ 代码提交到 GitHub
- 规划发展路线
- 梳理推广思路
- 找渠道、建社区
- 分析用户

go-zero 开源伊始

- 定目标
- 找渠道
- 建社区
- 有数据驱动的思维
 - 分析渠道
 - 分析用户
 - 分析ROI

go-zero 开源伊始

- 技术分享
 - GoCN
 - Go夜读
 - InfoQ
 - ArchSumit

go-zero 开源伊始

- 技术文章
 - 微服务实践公众号
 - 知乎
 - InfoQ
 - GoCN
 - 开源中国
 - ...

go-zero 开源伊始

- 技术文章 (国外)
 - hacker news
 - reddit
 - medium
 - hashnode
 - ...

为什么早期要多分享？

- 冷启动
- 让更多的用户知道，提升项目影响力
- 更深入的理解用户的需求，避免闭门造车

分享要注重口碑

- GopherChina金牌讲师
- ArchSumit金牌讲师
- Go夜读年度最佳、最多播放量的分享

技术文章

2023/05/29 15:40

go-zero 对代码质量的追求由此可见!

👁 4,660 👍 74 🔄 7

2023/03/30 09:39

当 go-zero 邂逅 chatgpt... 原创

👁 6,355 👍 42 🔄 2 💰 ¥5

2022/12/06 09:42

玩转 Go 链路追踪 原创

👁 5,503 👍 17 🔄 4

2022/11/29 09:35

Go 服务自动收集线上问题现场 原创

👁 4,411 👍 38 🔄 8

动态 回答 224 视频 12 提问 5 文章 134 专栏 1 想法 2 收藏 16 关注

我的动态

回答了问题 2022-02-27 13:31

置顶 如何看待国内后台开发从Java转向Go的趋势?

 万俊峰Kevin
go-zero.dev & 公众号: 微服务实践 

408 人赞同了该回答

作为一个 Java 和 GO 都差不多写了8年（其中有两年重叠）的老牌程序员来强答一波！存在即合理！遥想当年 Java 高速发展，同样一堆 C++ 程序员质疑和看衰 Java（包括我），然而现实已证明一切，后来我也写 Java 了！我再来分析分析大家为啥转：服务器成本。同样一个服务，你肯定知道一个简单服务 Java 打... [阅读全文](#)

👍 赞同 408 99 条评论 分享 收藏 设置

技术文章

go-zero会发展成下一个spring boot吗?



万俊峰Kevin

go-zero.dev & 公众号: 微服务实践

651 人赞同了该回答

谢谢, 刚发现这个邀请。作为 go-zero 作者, 我很诚实的说我没有用过 spring boot, 之前写了很多年各种语言, 包括 Java, 但我都是做 IM 相关的架构的, 没机会深度学习和使用 spring boot, 略遗憾! 我在写 go-zero 的时候确实只是不停的从实际业务中抽象, 奉行『保持简单』、『工具大于约定和... [阅读全文](#) ↓

▲ 赞同 651 ● 65 条评论 ↗ 分享 ★ 收藏 ⚙ 设置

如何看待国内后台开发从Java转向Go的趋势?



万俊峰Kevin

go-zero.dev & 公众号: 微服务实践

408 人赞同了该回答

作为一个 Java 和 GO 都差不多写了8年(其中有两年重叠)的老牌程序员来强答一波! 存在即合理! 遥想当年 Java 高速发展, 同样一堆 C++ 程序员质疑和看衰 Java (包括我), 然而现实已证明一切, 后来我也写 Java 了! 我再来分析分析大家为啥转: 服务器成本。同样一个服务, 你肯定知道一个简单服务 Java 打... [阅读全文](#) ↓

▲ 赞同 408 ● 99 条评论 ↗ 分享 ★ 收藏 ⚙ 设置

Implementing Service Discovery for Microservices



Kevin Wan

Published in Dev Genius · 6 min read · Sep 13, 2021



390



1



What is service discovery?

For developers who work with microservices, the concepts of service discovery should be familiar.



Alex Paredes

For example, if service A depends on service B, we need to tell service A where to invoke service B. This is the problem that service discovery has to solve.

技术文章

The martial arts of writing Go Dockerfile



Kevin Wan

Published in FAUN Publication · 7 min read · Mar 14, 2022



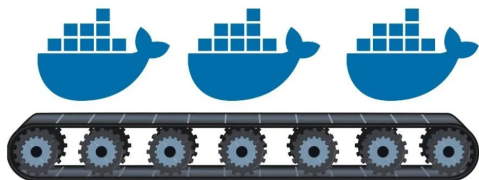
310



6



Multi-Stage



The pitfalls on using timeout in Golang



Kevin Wan

Published in FAUN Publication · 6 min read · Jun 1, 2021



270



1



技术文章

312 r/golang · Posted by u/go-zero 1 year ago 2 3

How I learned a programming language, and how I wrote a popular microservice framework.

The programming language that I'm talking about is Go, but I think it's almost the same for most programming languages.

I've been writing code more than 20 years. Before using Go, I've used many programming languages, includes C, C++, C#, Java. Early in year 2013, I decided to learn Go.

I spent one day to read through Go syntax and keywords, and spent couple days to read the official docs and blogs. [Effective Go](#) is highly recommended. After that, I started to write a small crawler project, and found that writing Go code was very comfortable.

Then I started to use Go to rewrite our production services, which were written in Java and PHP. And In 2015, all our production services were in Go. We have more than 30 million users and more than 3 million daily active users. In comparison with Java and PHP, we saved about 50% server/cloud cost.

In 2018, I joined another company, more than 50 million users and about 7 million daily active users. Because of lots of service downtime, I decided to rewrite the services with Go and my framework. We spent about 5 months to work on it, and we got uptime better than 99.99%. Also, we worked out some dev tools that helped us improved much productivity.

All the framework and tools are open sourced in GitHub in later 2020. Until now, lots of users and feedbacks helped us improved pretty much. And 15.6k stars 2.2k forks for about 1.5 years.

<https://github.com/zeromicro/go-zero>

I made ~2k contributions on github in the last year. And I open sourced some other projects, all these efforts helped me a lot on both technical skills and job opportunities.

Hope it helps!

50 Comments Share Save ...

Post Insights

Insights are only available for 45 days after post creation. [Create a new post](#) and see how it performs!



kevwon

Keep it simple!

帐号设置



5.9K 6K 392 3 56
经验值 开源豆 粉丝 关注 收藏夹

草稿箱 (6)

软件作品

- go-zero
Go Web 与 RPC 框架 2.3K ★
- go-stash
高效的 Logstash 替代方案 30 ★
- cds
ClickHouse 数据同步 19 ★

全部博文 资讯 问答 评论 消息 按时间 创作历程 搜索 kevwon

玩转 Go 链路追踪

前言 链路追踪是每个微服务架构下必备的利器。go-zero 当然早已经为我们考虑好了，只需要在配置中添加配置即可使用。关于 go-zero 如何追踪的原理追溯，之前已经有同学分享，这里我就不再多...

go-zero 2022/12/07 09:46 6.3K 0 编辑 删除

Go 服务自动收集线上问题现场

前言 对于 pprof，相信熟悉 Go 语言的程序员基本都不陌生，一般线上的问题都是靠它可以快速定位。但是实际项目中，很多时候我们为了性能都不会开启它，但是出了问题又要靠它来分析。好在 go...

go-zero 2022/12/01 07:03 6.7K 0 编辑 删除

熔断原理分析与源码解读

熔断机制 (Circuit Breaker) 指的是在股票市场的交易时间中，当价格的波动幅度达到某一个限定的目标 (熔断点) 时，对其暂停交易一段时间的机制。此机制如同保险丝在电流过大时候熔断，故得...

go-zero 2022/08/29 10:12 2.4K 0 编辑 删除

负载均衡原理分析与源码解读

上一篇文章一起学习了解决了Resolver的原理和源码分析，本篇继续和大家一起学习下和Resolver关系密切的Balancer的相关内容。这里说的负载均衡主要指数据中心内的负载均衡，即RPC间的负载均衡。传送...

go-zero 2022/08/10 09:44 8.4K 0 编辑 删除

服务发现原理分析与源码解读

在微服务架构中，有许多绕不开的技术话题。比如服务发现、负载均衡、指标监控、链路追踪，以及服务治理相关的超时控制、熔断、降级、限流等，还有RPC框架。这些都是微服务架构的基础，只有打...

go-zero 2022/07/26 08:44 1.7K 0 编辑 删除

go-zero 推广思路

- 项目本身才是关键
- 技术大会
- 线上线下分享
- 社区建设
- 国内国外兼顾
- 推广自己，但不要排斥、攻击别人
- 生态发展，上下游协同

go-zero 社区发展

- 社区交流群的选择
 - 微信、QQ、飞书、slack、discord、TG?
 - 多个群？一个超级大群？
 - geek 抱怨微信不好用？
 - TG 更适合技术交流？

go-zero 社区发展

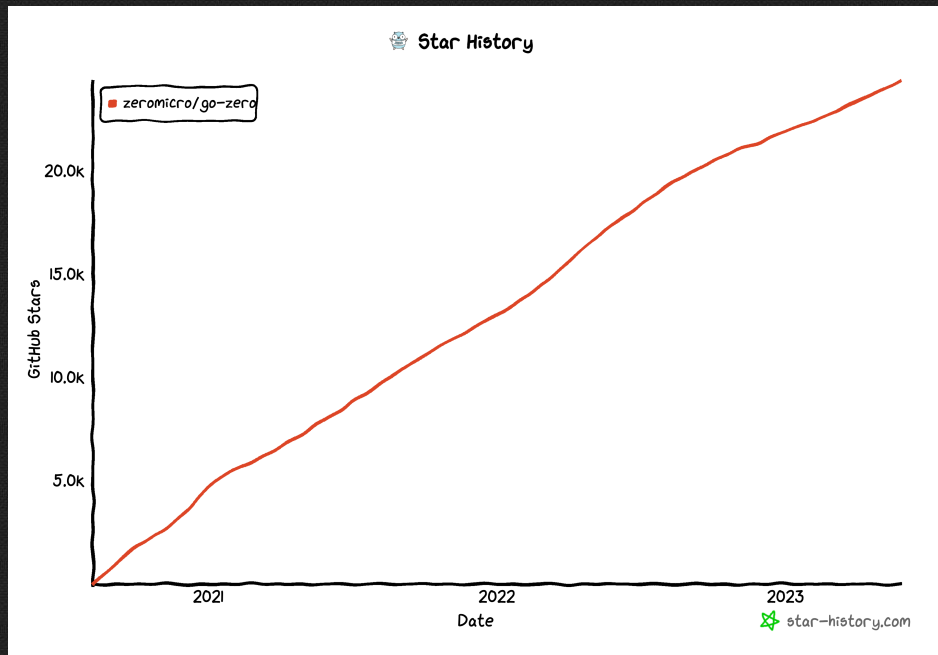
- 如何打造社区氛围
 - 怎样培养技术氛围?
 - 是否允许吹水?
 - 内容越界是否管理?
 - 群里冲突怎么解决?

go-zero 社区发展

- 壮大社区
 - 核心成员
 - 热心成员
 - 如何成就大家（技术和机会）
 - 极个别极端成员如何应对？

go-zero 的未来

- 功能规划
- 开源和商业化的思考



<https://github.com/zeromicro/go-zero>

欢迎 star, fork, issue, PR! 🙌



微服务实践

微信扫描二维码，关注我的公众号



GitHub

<https://github.com/zeromicro/go-zero>