

Golang 后端开发方向

WaveYo

后端开发工程师 (Golang)

邮箱 waveyo@waveyo.cn

GitHub <https://github.com/WavesMan>

个人主页 <https://www.waveyo.cn/>

期望城市 **北京/上海/广州/深圳/杭州/南京/合肥**

期望薪资 **实习**

求职意向

岗位: 后端开发工程师 / Golang 后端开发

城市: 北京/上海/广州/深圳/杭州/南京/合肥

薪资: 实习

个人优势

- 熟悉 Go 后端服务开发, 能够使用标准库 HTTP、Gin、Fiber、Go-Zero 等方式构建 API 服务、网关服务和后台工具。
- 具备对象存储相关项目实践, 做过 S3/OSS/COS 接入、文件列表缓存、预签名下载 URL、下载重定向和镜像同步。
- 在多个项目中实践过 Redis、本地 LRU/LFU、缓存分片、异步刷新、锁保护和布隆去重, 用于降低热点请求压力。
- 了解 BFF/网关类服务的基础设计, 做过控制面/数据面拆分、管理 API、动态配置、审计日志、限流和反向代理。
- 重视工程化交付, 能够补充 Docker、docker-compose、systemd、配置样例、健康检查和基础可观测能力。

专业技能

后端开发 Go / Golang / Gin / Fiber / Go-Zero / 标准库 HTTP / RESTful API / OpenAPI	存储与缓存 Redis / 本地 LRU/LFU / 缓存分片 / 异步刷新 / SQLite / MySQL / PostgreSQL	对象存储 S3 / OSS / COS / 预签名 URL / 文件列表缓存 / 下载跳转 / 镜像同步
工程交付 Docker / Docker Compose / systemd / Zap 日志 / Prometheus 指标 / Git / GitHub		

项目经历

YoOSF-API / Koishi-Mirror 对象存储镜像服务 2025/10 - 2026/01

后端开发 / 个人项目实践

围绕对象存储文件分发和 Koishi 插件镜像场景构建 Go 后端服务, 覆盖文件列表、搜索、包详情、预签名下载和同步任务。

- 基于 Go 对接 S3/OSS/COS, 提供文件列表、搜索、版本解析、包详情和预签名下载能力。
- 设计 Redis + 本地 LRU/LFU 的多级缓存方案, 对文件列表采用分片缓存和异步刷新, 并使用锁保护避免重复刷新任务。
- 在下载链路中实现 CDN 优先、S3 预签名 URL 兜底和未同步包回源代理, 减少服务端带宽占用并提升下载可用性。
- 为同步任务实现数据源拉取、并发上传、状态持久化、启动对账、失败重试和周期同步, 提升镜像数据一致性。
- 补充 Docker、docker-compose、systemd 和配置样例; README 记录的压测对比显示, 引入 LRU+Redis 后核心查询接口 QPS 从约 28k 提升到约 41.7k。

YoBFF Go BFF 网关与管理控制面

2026

后端开发 / 个人项目实践

使用 Go 实现 BFF 网关服务，将控制面、数据面和内嵌管理端 UI 整合到单进程服务中，支持单二进制部署。

- 实现基于 Host 的路由、负载均衡池、路由规则、HTTPS/SNI 证书匹配、HSTS 和可信代理 CIDR 校验等网关能力。
- 为管理端 API 设计 Bearer Token 登录、验证码防爆破、配置校验、版本快照、回滚和审计日志查询能力。
- 使用 SQLite 保存配置、审计和站点状态，并通过 OpenAPI 维护管理端接口契约。
- 支持配置 watcher 和 CDN/源站同步流程，提升配置变更与源站保护的维护性。

ip-api IP 地理信息查询服务

2026

后端开发 / 数据服务实践

使用 Go 标准库 HTTP 构建 IP 查询服务，整合 PostgreSQL、Redis、本地 IP 库和插件化数据源。

- 设计 KV 覆盖、Redis 热点缓存、本地动态缓存、数据库范围回退和插件融合的多级查询链路。
- 使用 Redis 布隆去重减少重复统计写入，并暴露请求量、耗时、Redis 命中率、插件调用等 Prometheus 指标。
- 实现令牌桶限流和 EdgeOne 地理请求头解析，在入口层注入上下文供融合层使用。

xi_cloud_disk 云盘系统协作改进

2026

PR 贡献者 / 协作开发

参与 Go-Zero 云盘系统后端链路改进，项目使用 MySQL、Redis、Aliyun OSS、JWT、Xorm 等技术栈。

- 围绕文件生命周期参与删除、回收、定时清理或脏数据处理相关改进，关注误删恢复和数据一致性。
- 参与 OSS 上传、分片上传、预签名下载或对象删除链路相关改进，提升文件访问和对象存储操作可靠性。
- 参与文件上传场景认证中间件优化，避免认证逻辑读取完整 multipart body 造成大文件上传性能问题。

Rainyun API Go SDK

2026

SDK 开发实践

围绕 Rainyun API 设计 Go SDK 客户端结构，将多个业务领域模块聚合到统一 Client。

- 抽象 Transport 接口，使用 Resty 实现统一 BaseURL、API Key、超时、请求选项和错误处理。
- 提供 MockTransport 支持预置响应和错误注入，隔离真实网络依赖并提升单元测试稳定性。

教育经历

长沙南方职业学院 大专（在读，预计 2027 年毕业） 计算机网络技术 2024 - 2027

荣誉奖项