

目 录

- 音频和音乐
- 身份验证和OAuth
- 机器人相关
- 标准CLI
- 构建用户界面库
- 配置
- 持续集成
- CSS预处理器
- 数据结构
- 数据库
 - 数据库
 - 数据库架构迁移
 - 数据库工具
 - SQL查询构建器
- 数据库驱动
 - 关系数据库
 - NoSQL数据库
 - 搜索和分析数据库
 - 多个后端
- 日期和时间
- 分布式系统
- 电子邮件
- 嵌入式脚本语言
- 错误处理
- 文件
- 金融
- Forms
- 功能性
- 游戏开发
- 生成与泛型
- 地理位置

编译器

Goroutines

图形界面

图片

物联网

工作计划

JSON格式

Logging

机器学习

实现消息传递

微软办公软件

依赖注入

项目布局

Strings

其他

自然语言处理

网络

HTTP客户端

OpenGL

ORM

包管理

性能

查询语言

资源嵌入

科学与数据分析

安全

序列化

服务器应用

流处理

模板引擎

测试

文字处理

第三方API

实用工具

UUID

验证方式

版本控制

视频

Web框架

中间件

路由器

视窗

XML格式

代码分析

编辑器插件

硬件

go生成工具

go工具

DevOps工具

音频和音乐

音频和音乐

用于处理音频的库。

- [EasyMIDI](#) - EasyMidi是一个简单可靠的库，用于处理标准Midi文件（SMF）。
- [flac](#) - 支持FLAC流的Native Go FLAC编码器/解码器。
- [gaad](#) - 本机Go AAC比特流解析器。
- [go-sox](#) - 用于go的libsox绑定。
- [go_mediaminfo](#) - 用于go的libmediainfo绑定。
- [gosamplerate](#) - 用于go的libsamplerate绑定。
- [id3v2](#) - 用于Go的快速，稳定的ID3解析和编写库。
- [malgo](#) - 迷你音频库。
- [minimp3](#) - 轻量级MP3解码器库。
- [mix](#) - 为音乐应用程序基于序列转到本地音频混合器。
- [mp3](#) - Native Go MP3解码器。
- [music-theory](#) - Go中的音乐理论模型。
- [Oto](#) - 在多个平台上播放声音的低级库。
- [PortAudio](#) - 用于PortAudio音频I / O库的绑定。
- [portmidi](#) - 绑定PortMidi。
- [taglib](#) - 为taglib绑定。
- [vorbis](#) - “本机” Go Vorbis解码器（使用CGO，但没有依赖项）。
- [waveform](#) - Go程序包，能够从音频流生成波形图像。

身份验证和OAuth

身份验证和OAuth

用于实施认证方案的库。

- [authboss](#) - Web的模块化身份验证系统。它尝试删除尽可能多的样板文件和“硬东西”，以便每次在Go中启动新的Web项目时，都可以将其插入，配置并开始构建应用程序，而不必每次都构建身份验证系统。
- [branca](#) - Branca令牌的Golang实现。
- [casbin](#) - 授权库，支持ACL，RBAC，ABAC等访问控制模型。
- [cookiestxt](#) - 提供cookies.txt文件格式的解析器。
- [go-jose](#) - JOSE工作组的JSON Web令牌，JSON Web签名和JSON Web加密规范的相当完整的实现。
- [go-oauth2-server](#) - server-用Golang编写的独立，符合规范的OAuth2服务器。
- [gologin](#) - 用于使用OAuth1和OAuth2身份验证提供程序登录的可链接处理程序。
- [gorbac](#) - 在Golang中提供基于角色的轻量级访问控制（RBAC）实现。
- [goth](#) - 提供一种简单，干净且惯用的方式使用OAuth和OAuth2。开箱即用地处理多个提供程序。
- [httpauth](#) - HTTP身份验证中间件。
- [jeff](#) - 具有可插入后端的简单，灵活，安全且惯用的Web会话管理。
- [jwt](#) - JSON Web令牌（JWT）的简洁易用实现。
- [jwt](#) - 轻量级JSON Web令牌（JWT）库。
- [jwt-auth](#) - 用于Golang http服务器的JWT中间件，具有许多配置选项。
- [jwt-go](#) - JSON Web令牌（JWT）的Golang实现。
- [loginsrv](#) - 具有可插入后端的JWT登录微服务，例如OAuth2（Github），htpasswd和osiam。
- [oauth2](#) - goauth2的后继者。JWT，Google API，Compute Engine和App Engine支持随附的通用OAuth 2.0程序包。
- [osin](#) - Golang OAuth2服务器库。
- [paseto](#) - 平台无关安全令牌（PASETO）的Golang实现。
- [permissions2](#) - 用于跟踪用户，登录状态和权限的库。使用安全的cookie和bcrypt。
- [rbac](#) - 用于Go应用程序的简约RBAC软件包。
- [scope](#) - 在Go中轻松管理OAuth2范围。
- [scs](#) - HTTP服务器的会话管理器。
- [securecookie](#) - 高效的安全cookie编码/解码。
- [session](#) - Web服务器的Go会话管理（包括对Google App Engine-GAE的支持）。
- [sessiongate-go](#) - go-使用SessionGate Redis模块进行的会话管理。
- [sessions](#) - 用于go http服务器的简单，高性能，高度可定制的会话服务。
- [sessionup](#) - 简单而有效的HTTP会话管理和标识包。
- [signedvalue](#) - 签署并兼容时间戳的字符串Tornado's `create_signed_value`，`decode_signed_value`，以及 `set_secure_cookie` 和 `get_secure_cookie`。
- [sjwt](#) - 简单的jwt生成器和解析器。
- [gin-oauth2](#) - 使用OAuth2的Gin Framework用户的中间件

机器人相关

机器人相关

用于构建和使用机器人的库。

- [go-chat-bot](#) - bot-用Go编写的IRC, Slack和Telegram机器人。
- [go-joe](#) - 受Hubot启发但用Go编写的通用bot库。
- [go-sarah](#) - 为所需的聊天服务(包括LINE, Slack, Gitter等)构建机器人的框架。
- [go-tgbot](#) - 纯Golang电报Bot API包装器, 由swagger文件, 基于会话的路由器和中间件生成。
- [Golang CryptoTrading Bot](#) - 基于控制台的交易机器人的golang实现, 用于加密货币交易。
- [govkbot](#) - 简单的Go Go VK 机器人库。
- [hanu](#) - 编写Slack机器人的框架。
- [Kelp](#) - DEX的官方交易和做市机器人。开箱即用, 用Golang编写, 可与集中交易和自定义交易策略兼容。
- [margelet](#) - 构建Telegram机器人的框架。
- [micha](#) - 用于电报Bot API的库。
- [slacker](#) - 易于使用的框架来创建Slack机器人。
- [slackscot](#) - 用于构建Slack机器人的另一个框架。
- [tbot](#) - 具有类似于net / http的API的电报bot服务器。
- [telebot](#) - 用Go编写的电报bot框架。
- [telegram-bot-api](#) - 简单干净的Telegram bot客户端。
- [Tenyks](#) - 使用Redis和JSON进行消息传递的面向服务的IRC机器人。

标准CLI

标准CLI

用于构建标准或基本命令行应用程序的库。

- [1build](#) - 命令行工具，可轻松管理特定于项目的命令。
- [argparse](#) - 受Python的argparse模块启发的命令行参数解析器。
- [argv](#) - Go库使用bash语法将命令行字符串拆分为参数数组。
- [cli](#) - 基于golang结构标签易于使用的命令行包功能丰富。
- [cli](#) - 用于在Go中构建命令行界面的简单而完整的API。
- [cli-init](#) - 开始构建Golang命令行应用程序的简单方法。
- [climax](#) - 按照Go命令的精神，具有“人脸”功能的替代CLI。
- [clir](#) - 一个简单清晰的CLI库。无依赖关系。
- [cmd](#) - 扩展标准flag软件包以支持子命令，并以idomatic方式提供更多支持。
- [cmdr](#) - 一种POSIX / GNU样式，类似于getopt的命令行UI Go库。
- [cobra](#) - 现代Go CLI交互的指挥官。
- [commandeer](#) - 开发人员友好的CLI应用程序：基于结构字段和标签设置标志，默认值和用法。
- [complete](#) - 在Go + Go命令bash完成中编写bash完成。
- [Dnote](#) - 适用于开发人员的简单，端到端加密笔记本。
- [docopt.go](#) - 命令行参数解析器，它将使您微笑。
- [env](#) - 基于标记的结构环境配置。
- [flag](#) - 简单但功能强大的命令行选项解析库，用于支持围棋子。
- [flaggy](#) - 健壮且惯用的标志包，具有出色的子命令支持。
- [flagvar](#) - Go的标准flag软件包的标志参数类型的集合。
- [go-arg](#) - Go中基于结构的参数解析。
- [go-commander](#) - Go库可简化CLI工作流程。
- [go-flags](#) - go命令行选项解析器。
- [go-getoptions](#) - Go选项解析器，它启发了Perl的GetOpt :: Long的灵活性。
- [gocmd](#) - 用于构建命令行应用程序的Go库。
- [hiboot cli](#) - 具有自动配置和依赖项注入的cli应用程序框架。
- [job](#) - 将您的短期命令作为一项长期工作。
- [kingpin](#) - 支持子命令的命令行和标志解析器。
- [liner](#) - 用于命令行界面的类似于readline的库。
- [mitchellh/cli](#) - 用于实现命令行界面的Go库。
- [mow.cli](#) - Go库，用于使用复杂的标志和参数解析和验证来构建CLI应用程序。
- [ops](#) - Unikernel生成器/协调器。
- [pflag](#) - Go的标志包的直接替换，实现POSIX / GNU样式的-flags。
- [readline](#) - 纯的golang实现，根据MIT许可在GNU-Readline中提供大多数功能。
- [sand](#) - 用于创建解释器的简单API等。
- [sflags](#) - 基于结构的标志生成器，用于标志，urfave / cli，pflag，眼镜蛇，主销和其他库。
- [strumt](#) - 创建提示链的库

标准CLI

- [ts](#) - 时间戳转换和比较工具。
- [ukautz/clif](#) - 小型命令行界面框架。
- [urfave/cli](#) - 简单，快速，有趣的软件包，用于在Go中构建命令行应用程序（以前为codegangsta / cli）。
- [wlog](#) - 支持跨平台颜色和并发性的简单日志记录界面。
- [wmenu](#) - 用于cli应用程序的易于使用的菜单结构，提示用户做出选择。

构建用户界面库

构建用户界面库

用于构建控制台应用程序和控制台用户界面的库。

- [asciigraph](#) - Go软件包，可在命令行应用程序中制作轻量级的ASCII线图，而无需其他依赖项。
- [aurora](#) - ANSI终端颜色支持fmt.Printf / sprintf的。
- [cfmt](#) - 受引导程序颜色类启发的上下文fmt。
- [chalk](#) - 直观的包装，用于美化终端/控制台输出。
- [colourize](#) - Go库，用于终端中的ANSI颜色文本。
- [ctc](#) - 非侵入性跨平台终端颜色库不需要修改Print方法。
- [go-ataman](#) - 用于在终端中呈现ANSI彩色文本模板的Go库。
- [go-colorable](#) - Windows 的可着色书写器。
- [go-colortext](#) - 用于在终端输出颜色的Go库。
- [go-isatty](#) - golang的isatty。
- [go-prompt](#) - 受python-prompt-toolkit启发，用于构建功能强大的交互式提示的库。
- [gocui](#) - 极简主义Go库，旨在创建控制台用户界面。
- [gommon/color](#) - 样式终端文本。
- [gokit/color](#) - 终端颜色渲染工具库，支持16种颜色，256种颜色，RGB颜色渲染输出，与Windows兼容。
- [mpb](#) - 终端应用程序的多进度条。
- [progressbar](#) - 适用于每个操作系统的基本线程安全进度条。
- [simpletable](#) - Go终端中的简单表。
- [tabby](#) - 超级简单Golang表一个小小的图书馆。
- [tabular](#) - 从命令行实用程序打印ASCII表，而无需将大量数据传递给API。
- [termbox-go](#) - Termbox是一个用于创建跨平台基于文本的界面的库。
- [termdash](#) - 基于termbox的并受termui启发的终端仪表盘。
- [termui](#) - 基于termbox-go并受blessed-contrib启发的终端仪表盘。
- [uilive](#) - 用于实时更新终端输出的库。
- [uiprogress](#) - 灵活的库，用于在终端应用程序中呈现进度条。灵活的库，用于在终端应用程序中呈现进度条。
- [uitable](#) - 使用表格数据提高终端应用程序可读性的库。

配置

配置

用于配置解析的库。

- [config](#) - 用于Go项目的轻量级但功能强大的配置包。
- [config](#) - 云本机应用程序配置。仅两行将ENV绑定到结构。
- [config](#) - JSON或YAML配置与包装环境变量和标志解析。
- [configure](#) - 通过多个源提供配置，包括JSON，标志和环境变量。
- [confita](#) - 将配置从多个后端级联加载到结构中。
- [conflate](#) - 图书馆/工具合并来自任意URL，验证多个JSON / YAML / TOML文件针对JSON模式，并在架构中定义的默认值应用。
- [env](#) - 解析环境变量去结构（使用默认值）。
- [envcfg](#) - 将环境变量解编为Go结构。
- [envconf](#) - 来自环境的配置。
- [envconfig](#) - 从环境变量中读取您的配置。
- [envh](#) - 帮助程序管理环境变量。
- [gcfg](#) - 将INI样式的配置文件读取到Go结构中；支持用户定义的类型和子节。
- [genv](#) - 通过dotenv支持轻松读取环境变量。
- [go-up](#) - 一个简单的配置库，具有递归占位符解析度并且没有魔术。
- [goConfig](#) - 将结构解析为输入，并使用命令行，环境变量和配置文件中的参数填充此结构的字段。
- [godotenv](#) - Ruby的dotenv库的Go端口（从加载环境变量.env）。
- [gofigure](#) - 简化 Go应用程序的配置。
- [gone/jconf](#) - 模块化JSON配置。在不牺牲完整的配置序列化的情况下，使您的配置结构以及它们配置的代码以及将解析委派给子模块的过程。
- [gokit/config](#) - 应用程序配置管理（加载，获取，设置）。支持JSON，YAML，TOML，INI，HCL。多文件加载，数据覆盖合并。
- [harvester](#) - 收割机，一个易于使用的静态和动态配置包supportig播种，ENV VAR和集成领事。
- [hjson](#) - Human JSON，人类的配置文件格式。轻松的语法，更少的错误，更多的注释。
- [ingo](#) - 标志保留在类似ini的配置文件中。
- [ini](#) - 读取和写入INI文件的软件包。
- [joshbetz/config](#) - Go的小型配置库，可解析环境变量，JSON文件，并在SIGHUP上自动重新加载。
- [kelseyhightower/envconfig](#) - Go库，用于管理来自环境变量的配置数据。
- [koanf](#) - 轻巧，可扩展的库，用于读取Go应用程序中的配置。内置对JSON，TOML，YAML，env，命令行的支持。
- [konfig](#) - 在分布式处理时代，Go的可组合，可观察和高性能配置处理。
- [mini](#) - Golang软件包，用于解析ini样式的配置文件。
- [nasermirzaei89/env](#) - 用于读取环境变量的简单有用的程序包。
- [onion](#) - Go的基于层的配置，支持JSON，TOML，YAML，属性，etcd，env和使用PGP加密。
- [sprbox](#) - 支持构建环境的工具箱工厂和不可知的配置解析器（YAML，TOML，JSON和环境变量）。
- [store](#) - Go的轻量级配置管理器。
- [viper](#) - 使用尖牙进行配置。

配置

- `xdg` - 遵循XDG Standard标准的跨平台软件包。

持续集成

持续集成

持续集成帮助工具。

- [CDS](#) - 企业级CI / CD和DevOps自动化开源平台。
- [drone](#) - Drone是一个基于Go编写的基于Docker的持续集成平台。
- [duci](#) - 简单的ci服务器，不需要特定于域的语言。
- [gomason](#) - 在干净的工作区中测试，构建，签名和发布go二进制文件。
- [goveralls](#) - Go Coveralls.io连续代码覆盖率跟踪系统的集成。
- [overalls](#) - 多包go go项目封面，适用于工装裤等工具。
- [roveralls](#) - 递归覆盖率测试工具。

CSS预处理器

CSS预处理器

用于预处理CSS文件的库。

- [gcss](#) - Pure Go CSS预处理器。
- [go-libsass](#) - 转到100%与Sass兼容的libsass项目的包装器。

数据结构

数据结构

Go中的通用数据结构和算法。

- [algorithms](#) - 算法和数据结构。CLRS研究。
- [binpacker](#) - 二进制打包程序和解包程序可帮助用户构建自定义二进制流。
- [bit](#) - 具有额外的位旋转功能的Golang设置数据结构。
- [bitset](#) - 实现位集的Go包。
- [bloom](#) - 在Go中实现的Bloom过滤器。
- [bloom](#) - Golang Bloom过滤器实现。
- [boomfilters](#) - 用于处理连续无界流的概率数据结构。
- [concurrent-writer](#) - 高并发直接替换bufio.Writer。
- [conjungo](#) - 一个小型，强大而灵活的合并库。
- [count-min-log](#) - 执行Count-Min-Log草图：使用近似计数器进行近似计数（类似于Count-Min草图，但使用较少的内存）。
- [crunch](#) - Go包实现了用于轻松处理各种数据类型的缓冲区。
- [cuckoofilter](#) - Cuckoo过滤器：是Go中实现的计数布隆过滤器的很好替代。
- [deque](#) - 高度优化的双端队列。
- [deque](#) - 快速的环形缓冲区双端队列（双端队列）。
- [dict](#) - Go的类似Python的字典（dict）。
- [encoding](#) - Go的整数压缩库。
- [go-adaptive-radix-tree](#) - 自适应基数树的Go实现。
- [go-datastructures](#) - 有用，高性能和线程安全的数据结构的集合。
- [go-ef](#) - Elias-Fano编码的Go实现。
- [go-geoindex](#) - 内存中的地理索引。
- [go-mcache](#) - 快速内存键：值存储/缓存库。指针缓存。
- [go-rquad](#) - 具有有效点定位和邻居发现功能的区域四叉树。
- [gocache](#) - 具有多个存储（内存，memcache，redis等），可链接，可加载，指标缓存等的完整Go缓存库。
- [goconcurrentqueue](#) - 并发FIFO队列。
- [gods](#) - 数据结构。容器，集合，列表，堆栈，地图，BidiMap，树，HashSet等。
- [gofal](#) - Go的小数api。
- [golang-set](#) - Go的线程安全和非线程安全高性能集。
- [goset](#) - Go的有用的Set集合实现。
- [goskiplist](#) - Go中的跳过列表实现。
- [gota](#) - Go的数据框，序列和数据整理方法的实现。
- [hide](#) - ID类型，将其编组进/出哈希以防止将ID发送给客户端。
- [hilbert](#) - Go程序包，用于在空间填充曲线（例如Hilbert和Peano曲线）之间映射值。
- [hyperloglog](#) - HyperLogLog实施，具有稀疏，LogLog-Beta偏差校正和TailCut空间减少功能。
- [iter](#) - C++ STL迭代器和算法的实现。
- [levenshtein](#) - Levenshtein距离和相似性度量标准，具有可自定义的编辑费用和通用前缀的类似于Winkler的奖金。

- [levenshtein](#) - 在Go中计算levenshtein距离的实现。
- [mafsa](#) - 具有最小完美散列的MA-FSA实现。
- [merkletree](#) - merkle树的实现，可对数据结构的内容进行有效且安全的验证。
- [mspm](#) - 用于信息检索的多字符串模式匹配算法。
- [null](#) - 可空转到类型，可以被编组/解组到/从JSON。
- [parsefields](#) - 用于解析类似JSON的日志的工具，以收集唯一的字段和事件。
- [pipeline](#) - 具有扇入和扇出的管线的实现。
- [ptrie](#) - 前缀树的实现。
- [remember-go](#) - 缓存慢速数据库查询的通用接口（由redis, memcached, ristretto或内存支持）。
- [ring](#) - 围棋实现了高性能，线程安全的布隆过滤器。
- [roaring](#) - 实施压缩位集的软件包。
- [set](#) - 使用LinkedHashMap的围棋设置简单的数据结构实现。
- [skiplist](#) - 非常快的Go Skiplist实施。
- [skiplist](#) - Go中的跳过列表实现。
- [timedmap](#) - 具有过期的键/值对的地图。
- [treap](#) - 使用树堆的持久快速排序的地图。
- [trie](#) - Go中的Trie实现。
- [ttlcache](#) - 内存中的LRU字符串接口{}映射，其中包含golang的到期时间。
- [typ](#) - 空类型，安全的原始类型转换和从复杂结构中获取值。
- [willf/bloom](#) - Go包实现Bloom过滤器。

数据库

数据库

数据库

数据库架构迁移

数据库工具

SQL查询构建器

数据库

数据库

Go中实现的数据库。

- [badger](#) - Go中的快速键值存储。
- [bcache](#) - 最终一致的分布式内存缓存Go库。
- [BigCache](#) - 高效的键/值缓存，用于千兆字节的数据。
- [Bitcask](#) - Bitcask是使用纯Go编写的可嵌入，持久且快速的键值（KV）数据库，由于具有bitcask磁盘布局（LSM + WAL），因此具有可预测的读/写性能，低延迟和高吞吐量。
- [bolt](#) - Go的低级键/值数据库。
- [buntdb](#) - 用于Go的快速，可嵌入的内存中键/值数据库，具有自定义索引和空间支持。
- [cache](#) - 内存中键：具有到期时间，0个依存关系，<100 LoC，100%覆盖率的值存储。
- [cache2go](#) - 内存键：值缓存，支持基于超时的自动失效。
- [clusteredBigCache](#) - 具有集群支持和单个项到期的BigCache。
- [cockroach](#) - 可扩展，地理复制，事务性数据存储。
- [Coffer](#) - 支持事务的简单ACID键值数据库。
- [couchcache](#) - 由Couchbase服务器支持的RESTful缓存微服务。
- [CovenantSQL](#) - CovenantSQL是区块链上的SQL数据库。
- [dgraph](#) - 可伸缩，分布式，低延迟，高吞吐量图形数据库。
- [diskv](#) - 本地磁盘支持的键值存储。
- [eliasdb](#) - 具有REST API，短语搜索和类似SQL的查询语言的无依赖关系的事务图数据库。
- [fastcache](#) - 快速线程安全的内存高速缓存，用于大量条目。最大限度地减少GC开销。
- [GCache](#) - 缓存库，支持过期的缓存，LFU，LRU和ARC。
- [go-cache](#) - Go的内存中键：值存储/缓存（类似于Memcached）库，适用于单机应用程序。
- [goleveldb](#) - Go中LevelDB键/值数据库的实现。
- [gorocksdb](#) - Gorocksdb是用Go编写的 [RocksDB](#) 的包装。
- [groupcache](#) - Groupcache是一个缓存和缓存填充库，在许多情况下可以替代memcached。
- [influxdb](#) - 可扩展的数据存储区，用于指标，事件和实时分析。
- [Kivik](#) - Kivik为CouchDB，PouchDB和类似数据库提供了通用的Go和GopherJS客户端库。
- [ledisdb](#) - Ledisdb是像基于LevelDB的Redis一样的高性能NoSQL。
- [levigo](#) - Levigo是LevelDB的Go包装器。
- [moss](#) - Moss是用100%Go编写的简单LSM键值存储引擎。
- [nutsdb](#) - Nutsdb是用纯Go编写的简单，快速，可嵌入，持久的键/值存储。它支持完全可序列化的事务和许多数据结构，例如列表，集合，排序集合。
- [piladb](#) - 基于堆栈数据结构的轻量级RESTful数据库引擎。
- [prometheus](#) - 监视系统和时间序列数据库。
- [pudge](#) - 使用Go的标准库编写的快速简单的键/值存储。
- [rqlite](#) - 基于SQLite构建的轻型，分布式，关系数据库。
- [Scribble](#) - 小型平面文件JSON存储。
- [slowpoke](#) - 具有持久性的键值存储。

数据库

- [tempdb](#) - 临时项目的键值存储。
- [tidb](#) - TiDB是分布式SQL数据库。受到Google F1设计的启发。
- [tiedot](#) - 由Golang提供支持的NoSQL数据库。
- [Vasto](#) - 分布式高性能键值存储。在磁盘上。最终一致。哈。能够增长或收缩而不会中断服务。
- [VictoriaMetrics](#) - 快速，资源有效且可扩展的开源时间序列数据库。可用作Prometheus的长期远程存储。支持PromQL。

数据库架构迁移

数据库架构迁移

- [avro](#) - 发现SQL模式并将其转换为AVRO模式。将SQL记录查询为AVRO字节。
- [darwin](#) - Go的数据库架构演化库
- [go-fixtures](#) - Golang出色的内置数据库/ sql库的Django样式的夹具。
- [go-pg-migrations](#) - 一个Go软件包，可帮助使用go-pg / pg编写迁移。
- [gondolier](#) - 使用结构修饰符的数据库迁移库。
- [goose](#) - 数据库迁移工具。您可以通过创建增量SQL或Go脚本来管理数据库的演变。
- [gormigrate](#) - Gorm ORM的数据库架构迁移帮助器。
- [migrate](#) - 数据库迁移。CLI和Golang库。
- [migrator](#) - 简单的Go数据库迁移库。
- [pravasan](#) - 简单迁移工具-当前用于MySQL，但计划很快支持Postgres，SQLite，MongoDB等。
- [schema](#) - 用于在Go二进制文件中嵌入数据库/ sql兼容数据库的模式迁移的库。
- [skeema](#) - MySQL的纯SQL模式管理系统，支持分片和外部在线模式更改工具。
- [soda](#) - 用于MySQL，PostgreSQL和SQLite的数据库迁移，创建，ORM等。
- [sql-migrate](#) - 数据库迁移工具。允许使用go-bindata将迁移嵌入到应用程序中。

数据库工具

数据库工具

- [bucket](#) - 为Couchbase优化数据结构框架，专门在一个水桶使用。
- [chproxy](#) - ClickHouse数据库的HTTP代理。
- [clickhouse-bulk](#) - 收集小的广告素材并将大的请求发送到ClickHouse服务器。
- [datagen](#) - 可以识别多表并支持多行DML的快速数据生成器。
- [dbbench](#) - 数据库基准测试工具，支持多个数据库和脚本。
- [go-mysql](#) - Go工具集，用于处理MySQL协议和复制。
- [go-mysql-elasticsearch](#) - elasticsearch-自动将您的MySQL数据同步到Elasticsearch。
- [kingshard](#) - kingshard是由Golang支持的MySQL的高性能代理。
- [myreplication](#) - MySQL二进制日志复制侦听器。支持基于语句和行的复制。
- [octillery](#) - 用于分片数据库的Go软件包（支持每个ORM或原始SQL）。
- [orchestrator](#) - MySQL复制拓扑管理器和可视化器。
- [pgweb](#) - 基于Web的PostgreSQL数据库浏览器。
- [prep](#) - 使用准备好的SQL语句而不更改代码。
- [pREST](#) - 从任何PostgreSQL数据库提供RESTful API。
- [rwdb](#) - rwdb为多个数据库服务器设置提供只读副本功能。
- [vitess](#) - vitess提供了有助于大规模数据库服务扩展MySQL数据库的服务器和工具。

SQL查询构建器

SQL查询构建器

用于构建和使用SQL的库。

- [dbq](#) - Go的零样板数据库操作。
- [Dotsql](#) - Go库，可帮助您将sql文件保存在一个位置并轻松使用它们。
- [gendry](#) - 非侵入性SQL构建器和强大的数据绑定器。
- [godbal](#) - go的数据库抽象层（dbal）。支持SQL Builder，轻松获得结果。
- [goqu](#) - 惯用的 SQL构建器和查询库。
- [igor](#) - PostgreSQL的抽象层，支持高级功能并使用类似于gorm的语法。
- [jet](#) - 框架写在Go类型安全的SQL查询，有能力轻松地转换数据库查询结果到期望的任意对象结构。
- [ormlite](#) - 轻量级软件包，包含一些类似于ORM的功能和sqlite数据库的帮助程序。
- [ozzo-dbx](#) - 强大的数据检索方法以及与数据库无关的查询构建功能。
- [qry](#) - 使用原始SQL查询从文件生成常量的工具。
- [scaneo](#) - 生成Go代码以将数据库行转换为任意结构。
- [sqlf](#) - 快速的SQL查询生成器。
- [sqr1](#) - SQL查询生成器，Squirrel的分支，具有改进的性能。
- [Squalus](#) - Go SQL包上的薄层，使执行查询更加容易。
- [Squirrel](#) - Go库，可帮助您构建SQL查询。
- [xo](#) - 根据现有的架构定义或支持PostgreSQL，MySQL，SQLite，Oracle和Microsoft SQL Server的自定义查询为数据库生成惯用的Go代码。

数据库驱动

关系数据库

NoSQL数据库

搜索和分析数据库

多个后端

关系数据库

关系数据库

- [avatica](#) - 用于数据库/ sql的Apache Avatica / Phoenix SQL驱动程序。
- [bgc](#) - BigQuery的数据存储连接。
- [firebirdsql](#) - Go的Firebird RDBMS SQL驱动程序。
- [go-adodb](#) - 用于使用数据库/ sql的go的Microsoft ActiveX对象数据库驱动程序。
- [go-mssqldb](#) - Go的Microsoft MSSQL驱动程序。
- [go-oci8](#) - 使用数据库/ sql的go的Oracle驱动程序。
- [go-sql-driver/mysql](#) - Go的MySQL驱动程序
- [go-sqlite3](#) - 使用数据库/ sql的go的SQLite3驱动程序。
- [gofreetds](#) - Microsoft MSSQL驱动程序。在[FreeTDS](#)上进行包装。
- [goracle](#) - 使用ODPI-C驱动程序的Go的Oracle驱动程序。
- [pgx](#) - PostgreSQL驱动程序支持的功能超出了数据库/ sql所公开的功能。
- [pq](#) - 用于数据库/ sql的Pure Go Postgres驱动程序。

NoSQL数据库

NoSQL数据库

- [aerospike-client-go](#) - Go语言的Aerospike客户端。
- [arangolite](#) - ArangoDB的轻量级golang驱动程序。
- [asc](#) - 数据存储连通性的塞式走。
- [dynago](#) - Dynago是DynamoDB的最少惊喜客户端的原则。
- [forestdb](#) - 森林数据库的绑定。
- [go-couchbase](#) - Go中的Couchbase客户端。
- [go-pilosa](#) - Pilosa的 Go客户库。
- [go-rejson](#) - 使用Redigo golang客户端的RedislabsReJSON模块的Golang客户端。轻松将结构作为JSON对象存储和操作在Redis中。
- [gocb](#) - 官方Couchbase Go SDK。
- [gocql](#) - Apache Cassandra的Go语言驱动程序。
- [godis](#) - 受jedis启发，golang的redis客户端实现。
- [godscache](#) - Google Cloud Platform Go数据存储区程序包的包装，可使用memcached添加缓存。
- [gomemcache](#) - Go编程语言的memcache客户端库。
- [gorethink](#) - RethinkDB的 Go语言驱动程序。
- [goriak](#) - Riak KV的Go语言驱动程序。
- [mgo](#) - （无需维护）用于Go语言的MongoDB驱动程序，该驱动程序遵循标准的Go习惯用法，通过非常简单的API来实现丰富且经过测试的功能选择。
- [mongo-go-driver](#) - Go语言的官方MongoDB驱动程序。
- [neo4j](#) - 用于Golang的Neo4j Rest API绑定。
- [Neo4j-GO](#) - golang中的Neo4j REST客户端。
- [neoism](#) -Golang的 Neo4j客户。
- [redeo](#) - 与Redis协议兼容的TCP服务器/服务。
- [redigo](#) - Redigo是Redis数据库的Go客户端。
- [redis](#) - Golang的Redis客户端
- [xredis](#) - 类型安全，可自定义，干净且易于使用的Redis客户端。

搜索和分析数据库

搜索和分析数据库

- [bleve](#) - go的现代文本索引库。
- [elastic](#) - Go的Elasticsearch客户端
- [elasticsql](#) - 转换SQL到elasticsearch DSL围棋。
- [elastigo](#) - Elasticsearch客户端库。
- [go-elasticsearch](#) - Go的官方Elasticsearch客户。
- [goes](#) - 与Elasticsearch进行交互的库。
- [riot](#) - 使用开源，分布式，简单高效的搜索引擎。
- [skizze](#) - 概率数据结构服务和存储。

多个后端

多个后端

- [cachego](#) - 用于多个驱动程序的Golang缓存组件。
- [cayley](#) - 支持多个后端的Graph数据库。
- [dsc](#) - SQL, NoSQL, 结构化文件的数据存储连接。
- [gokv](#) - Go的简单键值存储抽象和实现（Redis, Consul, etcd, bbolt, BadgerDB, LevelDB, Memcached, DynamoDB, S3, PostgreSQL, MongoDB, CockroachDB等）。

日期和时间

日期和时间

用于处理日期和时间的库。

- [carbon](#) - 简单的时间延长了很多UTIL方法，从PHP碳库移植。
- [cronrange](#) - 解析Cron风格的时间范围表达式，检查给定时间是否在任何范围内。
- [date](#) - Augments用于处理日期，日期范围，时间跨度，期间和日期的时间。
- [dateparse](#) - 在不知道格式的情况下解析日期。
- [durafmt](#) - Go的持续时间格式库。
- [feiertage](#) - 组用于计算德国公共假期的函数，包括德国（Bundesländer）州的专业。像复活节，五旬节，感恩节之类的东西。
- [go-persian-calendar](#) - Go（golang）中波斯（Solar Hijri）日历的实现。
- [go-sunrise](#) - 计算给定位置的日出和日落时间。
- [go-week](#) - 使用ISO8601周日期的有效软件包。
- [iso8601](#) - 无需正则表达式即可有效解析ISO8601日期时间。
- [kair](#) - 日期和时间-Golang格式库。
- [now](#) - 现在是golang的时间工具包。
- [NullTime](#) - Nullable time.Time。
- [strftime](#) - 与C99兼容的strftime格式化程序。
- [timespan](#) - 用于与时间间隔互动，定义为开始时间和持续时间。
- [timeutil](#) - golang时间包的有用扩展（Timedelta，Strftime等）。
- [tuesday](#) - 兼容Ruby的Strftime函数。

分布式系统

分布式系统

有助于构建分布式系统的软件包。

- [celeriac](#) - 用于在Go中添加支持以交互和监视Celery工作者，任务和事件的库。
- [consistent](#) - 具有受限负载的一致哈希
- [dht](#) - BitTorrent Kademlia DHT实施。
- [digota](#) - grpc电子商务微服务。
- [dot](#) - 使用操作转换/ OT进行分布式同步。
- [doublejump](#) - 改进后的Google的跳转一致性哈希。
- [dragonboat](#) - Go中功能齐全的高性能多组Raft库。
- [drmaa](#) - 基于DRMAA标准的集群调度程序的作业提交库。
- [dynamolock](#) - DynamoDB支持的分布式锁定实现。
- [dynatomic](#) - 将DynamoDB用作原子计数器的库。
- [emitter-io](#) - 使用MQTT, Websockets和love构建的高性能，分布式，安全和低延迟的发布-订阅平台。
- [flowgraph](#) - 基于流的编程包。
- [gleam](#) - 用纯围棋和Luajit快速和可扩展的分布式的map / reduce系统，具有Luajit的高性能结合Go的高并发，单独运行或分发。
- [glow](#) - 易于使用的可扩展的分布式大数据处理，Map-Reduce，DAG执行，全部在纯Go中进行。
- [go-health](#) - health-用于在服务中启用异步依赖项运行状况检查的库。
- [go-jump](#) - Google的“Jump”一致性哈希函数的端口。
- [go-kit](#) - 支持服务发现，负载平衡，可插拔传输，请求跟踪等的微服务工具包
- [go-sundheit](#) - 建立用于支持为golang服务定义异步服务运行状况检查的库。
- [gorpc](#) - 简单，快速和可扩展的RPC库，可实现高负载。
- [grpc-go](#) - gRPC的Go语言实现。基于HTTP / 2的RPC。
- [hprose](#) - 十分新颖的RPC库，现在支持25种以上的语言。
- [jsonrpc](#) - jsonrpc软件包可帮助实现JSON-RPC 2.0。
- [jsonrpc](#) - JSON-RPC 2.0 HTTP客户端实现。
- [KrakenD](#) - 具有中间件的超高性能API网关框架。
- [liftbridge](#) - NATS的轻量级，容错消息流。
- [micro](#) - 可插拔的microService工具箱和分布式系统平台。
- [NATS](#) - 用于微服务，IoT和云本机系统的轻量级高性能消息传递系统。
- [outboxer](#) - Outboxer是一个实现库模式的go库。
- [pglock](#) - PostgreSQL支持的分布式锁定实现。
- [raft](#) - HashiCorp的Raft共识协议的Golang实现。
- [raft](#) - 围棋实施筏一致协议，由CoreOS的。
- [rain](#) - BitTorrent客户端和库。
- [redis-lock](#) - 使用Redis的简化分布式锁定实现。
- [resgate](#) - 用于构建REST，实时和RPC API的实时API网关，其中所有客户端都可以无缝同步。
- [ringpop-go](#) - Go应用程序的可扩展，容错应用程序层分片。

- [rpcx](#) - 分布式可插拔RPC服务框架，例如阿里巴巴Dubbo。
- [sleuth](#) - 用于在HTTP服务之间进行无主p2p自动发现和RPC的库（[ZeroMQ](#)）。
- [tendermint](#) - 高性能中间件，用于使用Tendermint共识和区块链协议将以任何编程语言编写的状态机转换为拜占庭容错复制状态机。
- [torrent](#) - BitTorrent客户端软件包。

电子邮件

电子邮件

实现电子邮件创建和发送的库和工具。

- [chasquid](#) - 用Go编写的SMTP服务器。
- [douceur](#) - CSS内衬为您的HTML电子邮件。
- [email](#) - 用于Go的强大而灵活的电子邮件库。
- [go-dkim](#) - DKIM库，用于签名和验证电子邮件。
- [go-imap](#) - 用于客户端和服务器的IMAP库。
- [go-message](#) - Internet消息格式和邮件消息的流库。
- [go-premailer](#) - Go中HTML邮件的内联样式。
- [go-simple-mail](#) - 使用SMTP保持活动状态和两个超时发送电子邮件的非常简单的程序包：连接和发送。
- [Hectane](#) - 提供HTTP API的轻型SMTP客户端。
- [hermes](#) - Golang软件包，可生成干净的响应式HTML电子邮件。
- [mailchain](#) - 将加密的电子邮件发送到用Go编写的区块链地址。
- [mailgun-go](#) - Go库，用于使用Mailgun API发送邮件。
- [MailHog](#) - 通过Web和API界面进行电子邮件和SMTP测试。
- [SendGrid](#) - SendGrid的Go库，用于发送电子邮件。
- [smtp](#) - SMTP服务器协议状态机。

嵌入式脚本语言

嵌入式脚本语言

在您的go代码中嵌入其他语言。

- [anko](#) - 用Go语言编写的可编写脚本的解释器。
- [binder](#) - 转到基于[gopher-lua](#)的 Lua绑定库。
- [cel-go](#) - 具有渐进式输入功能的快速，便携式，非图灵完整表达评估。
- [expr](#) - 可以评估表达式的引擎。
- [gentee](#) - 可嵌入的脚本编程语言。
- [gisp](#) - Go中的简单LISP。
- [go-duktape](#) - Go的Duktape JavaScript引擎绑定。
- [go-lua](#) - Lua 5.2 VM到纯Go的端口。
- [go-php](#) - Go的PHP绑定。
- [go-python](#) - 与CPython C-API的幼稚go绑定。
- [golua](#) - Lua C API的绑定。
- [gopher-lua](#) - 用Go编写的Lua 5.1 VM和编译器。
- [gval](#) - 用Go编写的高度可定制的表达语言。
- [ngaro](#) - 可嵌入的Ngaro VM实现，支持在Retro中编写脚本。
- [otto](#) - 用Go编写的JavaScript解释器。
- [purl](#) - Go中嵌入的Perl 5.18.2。
- [tengo](#) - 用于Go的字节码编译脚本语言。

错误处理

错误处理

用于处理错误的库。

- [emperror](#) - Go库和应用程序的错误处理工具和最佳实践。
- [errlog](#) - 可破解的软件包，用于确定错误的责任的源代码（以及其他一些快速调试功能）。可插入任何现成的记录器。
- [errors](#) - 下拉更换为标准库的错误包和github.com/pkg/errors。提供各种错误处理原语。
- [errors](#) - 提供简单错误处理原语的软件包。
- [errors](#) - 简单golang错误处理与分类元。
- [errorx](#) - 具有堆栈跟踪，错误组成等的功能丰富的错误包。
- [Falcon](#) - 一个简单但功能强大的错误处理软件包。
- [go-multierror](#) - Go（golang）软件包，用于将错误列表表示为单个错误。
- [tracerr](#) - 带有堆栈跟踪和源代码片段的Golang错误。
- [werr](#) - 错误包装程序为Go中的错误类型创建了一个包装程序，该包装程序捕获了调用它的文件，行和堆栈。

文件

文件

用于处理文件和文件系统的库。

- [afero](#) - Go的文件系统抽象系统。
- [afs](#) - Go的抽象文件存储（mem, scp, zip, tar, 云: s3, gs）。
- [bigfile](#) - 文件传输系统，支持使用http api, rpc调用和ftp客户端管理文件。
- [checksum](#) - 计算大型文件的消息摘要，例如MD5和SHA256。
- [flop](#) - 文件操作库，旨在与GNU cp镜像功能奇偶校验。
- [go-csv-tag](#) - tag-使用标签加载csv文件。
- [go-decent-copy](#) - 复制human文件。
- [go-exiftool](#) - ExifTool的Go绑定，这是众所周知的库，用于从文件（图片，PDF，office，...）提取尽可能多的元数据（EXIF，IPTC等）。
- [go-gtfs](#) - 在go中加载gtfs文件。
- [notify](#) - 具有简单API的文件系统事件通知库，类似于os / signal。
- [opc](#) - 为Go加载Open Packaging Conventions（OPC）文件。
- [parquet](#) - 读取和写入 [parquet](#)文件。
- [pdfcpu](#) - PDF 处理器。
- [skywalker](#) - 一种软件包，允许一个人轻松地同时通过文件系统。
- [stl](#) - 读取和写入STL（立体光刻）文件的模块。并发读取算法。
- [tarfs](#) - tar文件F `FileSystem` [interface](#)接口的实现。
- [vfs](#) - 跨多种文件系统类型（例如os, S3和GCS）的Go的一组可插拔，可扩展且自以为是的文件系统功能。

金融

金融

会计和财务软件包。

- [accounting](#) - [golang](#)的货币和货币格式。
- [currency](#) - 高性能和准确的货币计算包。
- [decimal](#) - 任意精度定点十进制数字。
- [go-finance](#) - Go中的综合金融市场数据。
- [go-finance](#) - 金融功能库，用于货币时间价值（年金），现金流量，利率转换，债券和折旧计算。
- [go-finance](#) - 获取汇率，通过VIES检查增值税号和检查IBAN银行帐号的模块。
- [go-money](#) - Fowler的Money模式的实现。
- [ofxgo](#) - 查询OFX服务器和/或解析响应（使用示例命令行客户端）。
- [orderbook](#) - 匹配引擎的限价订单在Golang。
- [techan](#) - 具有高级市场分析和交易策略的技术分析库。
- [transaction](#) - 以多线程模式运行的嵌入式帐户嵌入式事务数据库。
- [vat](#) - 增值税号验证和欧盟增值税率。

Forms

Forms

用于处理表单的库

- [bind](#) - 绑定表单数据到任何围棋值。
- [binding](#) - 将net / http请求中的表单和JSON数据绑定到结构。
- [conform](#) - 检查用户输入。根据struct标签对数据进行整理，清理和清理。
- [form](#) - 将url.Values解码为Go值，并将Go值编码为url.Values。双阵列和全图支持。
- [formam](#) - 将表单的值解码为结构。
- [forms](#) - 与框架无关的库，用于解析和验证支持多部分表单和文件的表单/JSON数据。
- [gorilla/csrf](#) - Go Web应用程序和服务的CSRF保护。
- [nosurf](#) - Go的CSRF保护中间件。
- [queryparam](#) - 解码url.Values为标准或自定义类型的可用结构值。

功能性

功能性

支持Go中功能编程的软件包。

- [fpGo](#) - Monad，用于Golang的功能编程功能。
- [fuego](#) - Go中的功能实验。
- [go-underscore](#) - 有用的Go收集实用程序的有用集合。

游戏开发

游戏开发

很棒的游戏开发库。

- [Azul3D](#) - 用Go语言编写的3D游戏引擎。
- [Ebiten](#) - Go中死的简单2D游戏库。
- [engo](#) - Engo是用Go语言编写的开源2D游戏引擎。它遵循实体组件系统范式。
- [g3n](#) - Go 3D游戏引擎。
- [GarageEngine](#) - 用Go语言编写的2D游戏引擎，可在OpenGL上使用。
- [glop](#) - Glop（权力游戏库）是一个相当简单的跨平台游戏库。
- [go-astar](#) - A 路径查找算法的Go实现。
- [go-collada](#) - Go包，用于Collada文件格式。
- [go-sdl2](#) - Simple DirectMedia Layer的 Go绑定。
- [go3d](#) - 用于Go的面向性能的2D/3D数学软件包。
- [gonet](#) - 使用golang实现的游戏服务器框架。
- [goworld](#) -可扩展的游戏服务器引擎，具有空间实体框架和热插拔功能。
- [Leaf](#) - 轻量级游戏服务器框架。
- [nano](#) - 重量轻，设备，高性能的基于golang游戏服务器架构。
- [Oak](#) - Pure Go游戏引擎。
- [Pitaya](#) - 可扩展的游戏服务器框架，具有群集支持和通过C SDK的iOS，Android，Unity等客户端库。
- [Pixel](#) - Go中的手工制作2D游戏库。
- [raylib-go](#) - 去绑定raylib，简单和易于使用的库，以了解电子游戏编程。
- [termloop](#) - Go的基于终端的游戏引擎，建立在Termbox之上。

生成与泛型

生成与泛型

通过代码生成通过泛型等功能增强语言的工具。

- [efaceconv](#) - 代码生成工具，用于从接口{ }到不分配的不可变类型的高性能转换。
- [gen](#) - 用于类似“泛型”功能的代码生成工具。
- [generis](#) - 代码生成工具，提供泛型，自由格式的宏，条件编译和HTML模板。
- [go-enum](#) - 从代码注释生成枚举的代码。
- [go-linq](#) - Go的.NET LINQ式查询方法。
- [go-xray](#) - 帮助简化反射的助手。
- [goderive](#) - 从输入类型派生函数。
- [gotype](#) - Golang源代码解析，用法类似于反射包。
- [GoWrap](#) - 使用简单的模板为Go界面生成装饰器。
- [interfaces](#) - 用于生成接口定义的命令行工具。
- [jennifer](#) - 生成没有模板的任意Go代码。
- [pkgreflect](#) - 用于包范围反射的预处理器。

地理位置

地理位置

地理工具和服务器

- [geocache](#) - 适用于基于地理位置的应用程序的内存中缓存。
- [geoserver](#) - geoserver是Go软件包，用于通过GeoServer REST API操纵GeoServer实例。
- [gismanager](#) - 将 GIS数据（矢量数据）发布到PostGIS和Geoserver。
- [osm](#) - 用于读取，编写和使用OpenStreetMap数据和API的库。
- [pbf](#) - OpenStreetMap PBF golang编码器/解码器。
- [S2 geometry](#) - Go中的S2几何库。
- [Tile38](#) - 具有空间索引和实时地理围栏的地理位置数据库。
- [WGS84](#) - 库坐标转换和变换（ETRS89，OSGB36，NAD83，RGF93，网络墨卡托UTM）。

编译器

编译器

编译工具转到其他语言。

- [c4go](#) - 将C代码转换为Go代码。
- [f4go](#) - 将FORTRAN 77代码转换为Go代码。
- [gopherjs](#) - 从Go到JavaScript的编译器。
- [llgo](#) - Go的基于LLVM的编译器。
- [tardisgo](#) - Golang转换为CPP / CSharp / Java / JavaScript转译器。

Goroutines

Goroutines

用于管理和使用Goroutines的工具。

- [ants](#) - 用于golang的高性能goroutine池。
- [artifex](#) - Golang使用基于工作程序的分派的简单内存中作业队列。
- [async](#) - 一种异步执行功能的安全方法，以防万一。
- [breaker](#) - 使执行流程可中断的灵活机制。
- [cyclicbarrier](#) - 用于golang的CyclicBarrier。
- [go-floc](#) - 轻松编排goroutine。
- [go-flow](#) - 控制goroutine的执行顺序。
- [go-tools/multithreading](#) - 使用带有简单API的轻量级库管理goroutine池。
- [go-trylock](#) - 支持Golang的读写锁的TryLock。
- [go-waitgroup](#) - sync.WaitGroup与错误处理和并发控制类似。
- [gohive](#) - Go的高性能和易于使用的Goroutine池。
- [gollback](#) - 异步简单函数实用程序，用于管理闭包和回调的执行。
- [GoSlaves](#) - 简单和异步Goroutine池库。
- [goworker](#) - goworker是基于Go的后台工作者。
- [gowp](#) - gowp是并发限制goroutine池。
- [gpool](#) - 管理可调整大小的上下文感知goroutine池以绑定并发。
- [grpool](#) - 轻巧的Goroutine池。
- [Hunch](#) - 预感提供功能，如：All, First, Retry, Waterfall等等，这使得异步流控制更加直观。
- [oversight](#) - 监督是Erlang监督树的完整实现。
- [parallel-fn](#) - 并行运行功能。
- [pool](#) - 有限的消费者goroutine池或无限制的goroutine池，以便更轻松地处理和取消goroutine。
- [queue](#) - 为您提供sync.WaitGroup类似的队列组可访问性。帮助您节流和限制goroutine，等待所有goroutine结束等等。
- [routine](#) - 具有上下文和支持的例程控制：Main, Go, Pool和一些有用的Executors。
- [semaphore](#) - 基于通道和上下文的具有锁定/解锁操作超时的信号量模式实现。
- [semaphore](#) - 基于CAS的快速可调整大小的信号量实现（比基于通道的信号量实现更快）。
- [stl](#) - 基于软件交易内存（STM）并发控制机制的软件交易锁。
- [threadpool](#) - Golang线程池实现。
- [tunny](#) - 线程池golang。
- [worker-pool](#) - goworker是一个简单的Go异步工作池。
- [workerpool](#) - Goroutine池，它限制了任务执行的并发性，而不是排队的任务数。

图形界面

图形界面

用于构建GUI应用程序的库。

工具包

- [app](#) - 打包以使用GO, HTML和CSS创建应用的程序。支持: MacOS, Windows正在开发中。
- [fyne](#) - 为Go设计的跨平台本机GUI, 使用EFL呈现。支持: Linux, macOS, Windows。
- [go-astilelectron](#) - 使用GO和HTML / JS / CSS (由Electron支持) 构建跨平台GUI应用。
- [go-gtk](#) - GTK的绑定。
- [go-sciter](#) - Go绑定: 用于现代桌面UI开发的可嵌入HTML / CSS / script引擎。跨平台。
- [gotk3](#) - GTK3的绑定。
- [gowd](#) - 使用GO, HTML, CSS和NW.js进行快速简单的桌面UI开发。跨平台。
- [qt](#) - Go的Qt绑定 (支持Windows / macOS / Linux / Android / iOS / Sailfish OS / Raspberry Pi)。
- [ui](#) - Go的平台本地GUI库。跨平台。
- [Wails](#) - 使用内置OS HTML渲染器的HTML UI的Mac, Windows, Linux桌面应用程序。
- [walk](#) - Go的Windows应用程序库工具包。
- [webview](#) - 具有简单双向JavaScript绑定的跨平台Webview窗口 (Windows / macOS / Linux)。

相互作用

- [go-appindicator](#) - libappindicator3 C库的Go绑定。
- [gosx-notifier](#) - Go的OSX桌面通知库。
- [mac-activity-tracker](#) - OSX库, 用于通知计算机上的任何 (可插入) 活动。
- [mac-sleep-notifier](#) - golang中的OSX睡眠/唤醒通知。
- [robotgo](#) - Go本机跨平台GUI系统自动化。控制鼠标, 键盘等。
- [systray](#) - 跨平台的Go库, 用于在通知区域中放置图标和菜单。
- [trayhost](#) - 跨平台的Go库, 用于在主机操作系统的任务栏中放置一个图标。

图片

图片

用于处理图像的库。

- [bild](#) - 纯Go中图像处理算法的集合。
- [bimg](#) - 使用libvips进行快速有效的图像处理的小包装。
- [cameron](#) - Go的头像生成器。
- [canvas](#) - 将矢量图形转换为PDF, SVG或光栅图像。
- [darkroom](#) - 具有可变存储后端的图像代理和侧重于速度和弹性的图像处理引擎。
- [geopattern](#) - 从字符串创建漂亮的生成图像图案。
- [gg](#) - 纯Go中的2D渲染。
- [gift](#) - 图像处理过滤器的包装。
- [gltf](#) - 高效, 强大的glTF 2.0读取器, 写入器和验证器。
- [go-cairo](#) - 用于cairo图形库的绑定。
- [go-gd](#) - GD库的Go绑定。
- [go-nude](#) - Go的裸露检测。
- [go-opencv](#) - 用于OpenCV的绑定。
- [go-webcolors](#) - webcolors库的端口, 从Python到Go。
- [gocv](#) - 使用OpenCV 3.3+进行计算机视觉的Go软件包。
- [goimagehash](#) - Go感知图像哈希包。
- [goimghdr](#) - imghdr模块确定Go文件中包含的图像类型。
- [govatar](#) - 用于生成有趣头像的库和CMD工具。
- [image2ascii](#) - 将图像转换为ASCII。
- [imagemick](#) - 绑定到ImageMagick的MagickWand C API。
- [imaginary](#) - 用于图像大小调整的快速, 简单的HTTP微服务。
- [imaging](#) - 简单的Go图像处理包。
- [img](#) - 选择图像处理工具。
- [ln](#) - Go中的3D线条艺术渲染。
- [mergi](#) - 用于图像处理(合并, 裁切, 调整大小, 水印, 动画)的Tool&Go库。
- [mort](#) - 用Go编写的存储和图像处理服务器。
- [mpo](#) - 用于MPO 3D照片的解码器和转换工具。
- [picfit](#) - 用Go编写的图像大小调整服务器。
- [pt](#) - 用Go语言编写的路径跟踪引擎。
- [resize](#) - 使用常见的插值方法为Go调整图像大小。
- [rez](#) - 在纯Go和SIMD中调整图像大小。
- [smartcrop](#) - 查找适合任何图像和尺寸的优质作物。
- [steganography](#) - 用于LSB隐写术的Pure Go库。
- [stegify](#) - 用于LSB隐写术的Go工具, 能够隐藏图像中的任何文件。
- [svgo](#) - 用于SVG生成的Go语言库。
- [tga](#) - 软件包tga是TARGA图像格式的解码器/编码器。

图片

物联网

物联网

物联网设备编程库。

- [connectordb](#) - 量化自我和物联网的开源平台。
- [devices](#) - IoT设备库套件，针对x / exp / io进行实验。
- [eywa](#) - Project Eywa本质上是一个连接管理器，用于跟踪连接的设备。
- [flogo](#) - Project Flogo是一个用于IoT Edge应用和集成的开源框架。
- [gatt](#) - 盖特是一个围棋包构建低功耗蓝牙外设。
- [gobot](#) - Gobot是机器人技术，物理计算和物联网的框架。
- [huego](#) - 适用于Go的飞利浦Hue扩展客户端库。
- [iot](#) - IoT是用于实现Google IoT Core设备的简单框架。
- [mainflux](#) - 工业物联网消息和设备管理服务器。
- [periph](#) - 外设I / O与低级别的主板设备接口。
- [sensorbee](#) - 用于物联网的轻量级流处理引擎。

工作计划

工作计划

用于计划作业的库。

- [clockwerk](#) - 使用简单，流利的语法打包程序以安排定期作业。
- [clockwork](#) - Go中简单直观的工作计划库。
- [go-cron](#) - 用于go的简单Cron库，可以在不同的时间间隔（从每秒一次到每年的特定日期和时间）执行关闭或函数。主要用于Web应用程序和长期运行的守护程序。
- [gron](#) - 使用简单的Go API定义基于时间的任务，Gron的调度程序将相应地运行它们。
- [JobRunner](#) - 内置功能，工作队列和实时监控功能的智能，功能强大的cron作业计划程序。
- [jobs](#) - 持久灵活的后台作业库。
- [leprechaun](#) - 作业调度，支持网络挂接，crons和经典的调度。
- [scheduler](#) - Cronjobs调度变得容易。

JSON格式

JSON格式

用于处理JSON的库。

- [ajson](#) - 具有JSONPath支持的golang的抽象JSON。
- [gjo](#) - 用于创建JSON对象的小型实用程序。
- [GJSON](#) - 使用一行代码获取JSON值。
- [go-jsonerror](#) - Go-JsonError可让我们轻松创建遵循JsonApi规范的json响应错误。
- [go-respond](#) - Go包，用于处理常见的HTTP JSON响应。
- [gojq](#) - Golang中的JSON查询。
- [gojson](#) - 从示例JSON自动生成Go（golang）结构定义。
- [JayDiff](#) - 用Go编写的JSON diff实用程序。
- [jettison](#) - 用于Go的高性能，无反射JSON编码器。
- [JSON-to-Go](#) - 将JSON转换为Go结构。
- [json2go](#) - 高级JSON到Go结构转换。提供可以解析多个JSON文档并创建适合所有JSON的结构的包。
- [jsonapi-errors](#) -根据JSON API错误参考进行绑定。
- [jsonf](#) - 突出显示格式和获取JSON的结构查询的控制台工具。
- [jsongo](#) -Fluent API，可以更轻松地创建Json对象。
- [jsonhal](#) - 简单的Go包，用于将自定义结构编组为HAL兼容的JSON响应。
- [kazaam](#) - 用于JSON文档的任意转换的API。
- [mp](#) - 简单的cli电子邮件解析器。当前，它使用标准输入并输出JSON。

Logging

Logging

用于生成和使用日志文件的库。

- [distillog](#) - 精简级别的日志记录（将其视为stdlib + 日志级别）
- [glg](#) - glg是用于Go的简单快速的日志记录库。
- [glo](#) - PHP独白灵感记录设备具有相同的严重性级别。
- [glog](#) - Go的级别执行日志。
- [go-cronowriter](#) - 简单的作家，可以根据当前日期和时间自动旋转日志文件，例如cronolog。
- [go-log](#) - 具有跟踪跟踪，对象转储和可选时间戳记的日志记录库。
- [go-log](#) - Go中的简单且可配置的日志记录，包括级别，格式化程序和编写器。
- [go-log](#) - 日志库支持级别和多处理程序。
- [go-log](#) - Go中的Log4j实现。
- [go-logger](#) - Go程序的简单记录器，带有级别处理程序。
- [gologger](#) - 用于go的简单易用的日志库，可在彩色控制台，简单控制台，文件或Elasticsearch中登录。
- [gomol](#) - 具有可扩展日志输出的Go的多输出，结构化日志。
- [gone/log](#) - 快速，可扩展，全功能，std-lib源兼容的日志库。
- [journald](#) - 实施systemd Journal的本机API进行日志记录的实现。
- [log](#) - 日志记录系统，允许您将一个日志连接到多个编写器（例如stdout，文件和TCP连接）。
[log](#) -Go的结构化日志记录包。
- [log](#) -Go的结构化日志记录包。
- [log](#) - 简单，可配置和可扩展的Go结构化日志记录。
- [log](#) - Go的结构化日志界面将日志记录外观与其实现完全分开。
- [log-voyage](#) - 用golang编写的全功能日志记录saas。
- [log15](#) - 简单，强大的Go日志记录。

Logging

- [logdump](#) -用于多级日志记录的软件包。
- [logex](#) - Golang日志库，支持跟踪和级别，按标准日志库包装。
- [logger](#) -简约日志库进行搜索。
- [logmatic](#) - 具有动态日志级别配置的Golang彩色记录器。
- [logo](#) -Golang记录器可用于其他可配置的编写器。
- [logrus](#) - Go的结构化记录器。
- [logrusiowriter](#) - `io.Writer`使用实施[logrus](#)记录。
- [logrusly](#) -logrus插件将错误发送到Loggly。
- [logutils](#) - 实用程序，用于在Go（Golang）中更好地记录日志，从而扩展了标准记录器。
- [logxi](#) - 快速的12要素应用程序记录器，让您感到高兴。
- [lumberjack](#) - 简单滚动记录，实施[io.WriteCloser](#)。
- [mlog](#) - 用于go的简单日志记录模块，具有5个级别，可选的旋转日志文件功能和[stdout /stderr](#)输出。
- [onelog](#) - Onelog是一个非常简单但非常有效的JSON记录器。在所有情况下，它都是最快的JSON记录器。而且，它是分配最低的记录器之一。
- [ozzo-log](#) - 高性能日志记录，支持日志严重性，分类和过滤。可以将过滤后的日志消息发送到各种目标（例如，控制台，网络，邮件）。
- [rollingwriter](#) - RollingWriter是[io.Writer](#)具有多个策略的自动旋转实现，可提供日志文件旋转。
- [seelog](#) - 具有灵活的分派，过滤和格式化功能的日志记录功能。
- [spew](#) - 为Go数据结构实现深层漂亮的打印机，以帮助调试。
- [stdlog](#) - Stdlog是一个面向对象的库，提供分级日志记录。这对于cron作业非常有用。
- [tail](#) - Go程序包，旨在模仿BSD [tail](#)程序的功能。
- [xlog](#) - Go的插件架构和灵活的日志系统，具有ctrl级，多个日志目标和自定义日志格式。
- [xlog](#) - 结构化记录器，可[net/context](#)通过灵活的调度来感知HTTP处理程序。
- [zap](#) - Go中快速，结构化，分级的日志记录。
- [zerolog](#) - 零分配JSON记录器。

机器学习

机器学习

机器学习图书馆。

- [bayesian](#) - 贝叶斯分类为Golang天真。
- [CloudForest](#) - 快速，灵活，多线程的决策树集合，用于纯Go中的机器学习。
- [eaopt](#) - 进化优化库。
- [evoli](#) - 遗传算法和粒子群优化库。
- [fonet](#) - 用Go编写的深度神经网络库。
- [go-cluster](#) - k模式和k-原型聚类算法的Go实现。
- [go-deep](#) - Go中功能丰富的神经网络库
- [go-fann](#) - 快速人工神经网络（FANN）库的Go绑定。
- [go-galib](#) - 用Go / golang编写的遗传算法库。
- [go-pr](#) - Go lang中的模式识别包。
- [gobrain](#) - 用go语言编写的神经网络
- [godist](#) - 各种概率分布及相关方法。
- [goga](#) - Go的遗传算法库。
- [GoLearn](#) - 用于Go的通用机器学习库。
- [golinear](#) - Go的liblinear绑定。
- [GoMind](#) - Go中的简单神经网络库。
- [goml](#) - Go中的在线机器学习。
- [Goptuna](#) - 用于Go语言编写的黑盒函数的贝叶斯优化框架。一切都会被优化。
- [goRecommend](#) - 用Go编写的推荐算法库。
- [gorgonia](#) - 基于图形的计算库，例如Theano for Go，它提供了用于构建各种机器学习和神经网络算法的原语。
- [gorse](#) - 基于Go编写的协作过滤的离线推荐系统后端。
- [goscore](#) - 用于PMML的Go Scoring API。
- [gossertext](#) - 使用Tesseract C ++库的OCR（光学字符识别）软件包。
- [libsvm](#) - 基于LIBSVM 3.14 libsvm的golang版本衍生作品。
- [neat](#) - 用于增强拓扑神经演化（NEAT）的即插即用，并行Go框架。
- [neural-go](#) - go-在Go中实现的多层感知器网络，通过反向传播进行训练。
- [ocrserver](#) - 一个简单的OCR API服务器，非常容易被Docker和Heroku部署。
- [onnx-go](#) - 转到开放神经网络交换（ONNX）的接口。
- [probab](#) - 概率分布函数。贝叶斯推断。用纯Go语言编写。
- [regomment](#) - 建议和协作过滤引擎。
- [shield](#) - 贝叶斯文本分类器，具有灵活的标记器和Go的存储后端。
- [tfgo](#) - 易于使用的Tensorflow绑定：简化了官方Tensorflow Go绑定的使用。在Go中定义计算图，加载并执行经过Python训练的模型。
- [Varis](#) - Golang神经网络。

实现消息传递

实现消息传递

实现消息传递系统的库。

- [APNs2](#) - 用于Go的HTTP / 2 Apple Push Notification提供程序—将推送通知发送到iOS, tvOS, Safari和OSX应用程序。
- [Beaver](#) - 实时消息服务器, 用于构建可扩展的应用内通知, 多人游戏, Web和移动应用中的聊天应用。
- [Benthos](#) - 一系列协议之间的消息流桥。
- [Bus](#) - 用于内部通信的极简消息总线实现。
- [Centrifugo](#) - Go中的实时消息传递 (Websockets或SockJS) 服务器。
- [Commander](#) - 由事件驱动的高级消费者/生产者, 支持各种“方言”, 例如Apache Kafka。
- [dbus](#) - D-Bus的本机Go绑定。
- [drone-line](#) -line-使用二进制, 泊坞窗或Drone CI 发送行通知。
- [emitter](#) - 使用Go方式发出事件, 并使用通配符, 谓词, 取消可能性和许多其他好成绩。
- [event](#) - 模式观察器的实现。
- [EventBus](#) - 具有异步兼容性的轻量级事件总线。
- [gaurun-client](#) - client-用Go编写的Gaurun客户端。
- [Glue](#) - 强大的围棋和Javascript的Socket库 (替代Socket.io)。
- [go-notify](#) - freedesktop通知规范的本地实现。
- [go-nsq](#) - NSQ的官方Go软件包。
- [go-socket.io](#) - 用于golang (实时应用程序框架)的socket.io库。
- [go-vitotrol](#) - Viessmann Vitotrol Web服务的客户端库。
- [Gollum](#) - 一个n: m多路复用器, 从不同的来源收集消息并将其广播到一组目标。
- [golongpoll](#) - 使Web pub-sub变得简单的HTTP longpoll服务器库。
- [gopush-cluster](#) - gopush-cluster是go push服务器群集。
- [gorush](#) - 使用APNs2和Google GCM的推送通知服务器。
- [guble](#) - 使用推送通知 (Google Firebase Cloud Messaging, Apple推送通知服务, SMS) 以及websockets, REST API的消息服务器, 具有分布式操作和消息持久性。
- [hub](#) - Go应用程序的消息/事件中心, 使用发布/订阅模式并支持如RabbitMQ交换之类的别名。
- [jazz](#) - 一个简单的RabbitMQ抽象层, 用于队列管理以及消息的发布和使用。
- [machinery](#) - 基于分布式消息传递的异步任务队列/作业队列。
- [mangos](#) - 具有传输互操作性的Nanomsg (“可伸缩协议”)的纯go实现。
- [melody](#) - 为处理WebSocket的会议简约的框架, 包括广播和自动平/乒乓处理。
- [Mercure](#) - 服务器和库, 用于使用Mercure协议 (建立在Server-Sent Events之上) 调度服务器发送的更新。
- [messagebus](#) - messagebus是一种Go简单的异步消息总线, 非常适合在进行事件源, CQRS和DDD时用作事件总线。
- [NATS Go Client](#) - 轻便, 高性能的发布订阅和分布式排队消息传递系统-这是Go库。
- [nsq-event-bus](#) - bus-围绕NSQ主题和渠道的小包装。
- [oplog](#) - REST API的通用oplog /复制系统。
- [pubsub](#) - 用于go的简单pubsub软件包。
- [rabbus](#) - amqp交换和队列的小型包装。

- [rabort](#) - RabbitMQ瑞士军刀cli应用程序。
- [RapidMQ](#) - RapidMQ是用于管理本地消息队列的轻量级且可靠的库。
- [redisqueue](#) - redisqueue提供使用Redis流的队列的生产者和消费者。
- [rmqconn](#) - RabbitMQ重新连接。包装在amqp.Connection和amqp.Dial上。在断开连接之前允许进行重新连接，然后强制关闭Close（）方法的调用。
- [sarama](#) - Apache Kafka的Go库。
- [Uniqush-Push](#) - Redis支持的统一推送服务，用于向移动设备的服务器端通知。
- [zmq4](#) -到ZeroMQ版本4的接口。也可用于版本3和版本2。

微软办公软件

微软办公软件

- [unioffice](#) - Pure Go库，用于创建和处理Office Word（.docx），Excel（.xlsx）和Powerpoint（.pptx）文档。

Microsoft Excel

用于使用Microsoft Excel的库。

- [excelize](#) - Golang库用于读取和写入Microsoft Excel™（XLSX）文件。
- [go-excel](#) - 一个简单而轻便的阅读器，可以将类似于related-db的excel读取为表格。
- [goxlsxwriter](#) - libxlsxwriter的Golang绑定，用于编写XLSX（Microsoft Excel）文件。
- [xlsx](#) - 用于简化在Go程序中读取Microsoft Excel最新版本使用的XML格式的库。
- [xlsx](#) - 在Go程序中快速/安全地读取/更新您现有的Microsoft Excel文件的方法。

依赖注入

依赖注入

用于依赖注入的库。

- [alice](#) - Golang的附加依赖项注入容器。
- [container](#) - 具有流利且易于使用的界面的强大IoC容器。
- [dig](#) - Go的基于反射的依赖注入工具包。
- [fx](#) - Go的基于依赖项注入的应用程序框架（建立在[dig](#)之上）。
- [gocontainer](#) - 简单依赖项注入容器。
- [inject](#) - 具有简单接口的基于反射的依赖项注入容器。
- [linker](#) - 基于反射的依赖项注入和具有组件生命周期支持的控制库反转。
- [wire](#) - Golang的严格运行时依赖注入。

项目布局

项目布局

用于组织项目的非官方模式集。

- [cookiecutter-golang](#) - Go应用样板模板，用于遵循生产最佳实践快速启动项目。
- [go-sample](#) - 具有实际代码的Go应用程序项目的示例布局。
- [golang-standards/project-layout](#) - Go生态系统中常见的历史和新兴项目布局模式集。
- [modern-go-application](#) - Go应用程序样板和应用现代实践的示例。
- [scaffold](#) - 脚手架生成入门Go项目布局。让您专注于已实现的业务逻辑。

Strings

Strings

用于处理字符串的库

- [strutil](#) - 字符串实用程序。
- [xstrings](#) - 从其他语言移植的有用的字符串函数的集合。

其他

其他

将这些库放在此处是因为其他类别似乎都不适合。

- [anagent](#) - 具有依赖项注入功能的简约，可插入Golang evloop / timer处理程序。
- [antch](#) - 一个快速，强大且可扩展的Web爬网和抓取框架。
- [archiver](#) - 用于制作和提取.zip和.tar.gz档案的库和命令。
- [autoflags](#) - Go软件包可从struct字段自动定义命令行标志。
- [avgRating](#) - 根据Wilson得分方程计算平均得分和等级。
- [banner](#) - 将漂亮的标语添加到Go应用程序中。
- [base64Captcha](#) - Base64captcha支持数字，数字，字母，算术，音频和数字字母验证码。
- [battery](#) - 跨平台，标准化的电池信息库。
- [bitio](#) - 高度优化的Go语言位读写器。
- [browsercap_go](#) - GoLang L用于浏览器功能项目的 GoLang库。
- [captcha](#) - 验证码软件包为生成验证码提供了一个易于使用的，不受限制的API。
- [conv](#) - 包conv可在Go类型之间提供快速直观的转变。
- [datacounter](#) - 用于读者/作家/http.ResponseWriter的计数器。
- [ffmt](#) - 美化人类的数据显示。
- [ghorg](#) - 快速将整个组织/用户存储库克隆到一个目录中-支持GitHub，GitLab和Bitbucket。
- [go-commons-pool](#) - Golang的通用对象池。
- [go-openapi](#) - 解析和利用open-api模式的软件包集合。
- [go-resiliency](#) - golang的弹性模式。
- [go-unarr](#) - RAR，TAR，ZIP和7z档案的解压缩库。
- [gofakeit](#) - 用go编写的随机数据生成器。
- [gommit](#) - 分析git commit消息以确保它们遵循定义的模式。
- [gopsutil](#) - 跨平台库，用于检索进程和系统利用率（CPU，内存，磁盘等）。
- [gosh](#) - 提供Go统计处理程序，结构，度量方法。
- [gosms](#) - Go中您自己的本地SMS网关，可用于发送SMS。
- [gotoprom](#) - 官方Prometheus客户端的类型安全度量构建器包装器库。
- [gountries](#) - 公开国家和细分数据的软件包。
- [health](#) - 易于使用的可扩展健康检查库。
- [healthcheck](#) - 用于RESTful服务的经过审慎并发的运行状况检查HTTP处理程序。
- [hostutils](#) - 用于打包和解压缩FQDN列表的golang库。
- [indigo](#) - 使用Sonyflake并由Base58编码的分布式唯一ID生成器。
- [lk](#) - 一个简单的牌库golang。
- [llvm](#) - 用于在纯Go语言中与LLVM IR交互的库
- [metrics](#) - 指标检测和Prometheus展示的库。
- [morse](#) - 莫尔斯电码的来回转换库。
- [numa](#) - NUMA是一个实用的库，用go编写。它有助于我们编写一些NUMA-AWARED代码。
- [pdfgen](#) - HTTP服务，用于从Json请求生成PDF

- [persian](#) - go中波斯语语言的一些实用程序
- [sandid](#) - 地球上的每一粒沙都有自己的ID。
- [shellwords](#) - 一个Golang库，用于根据UNIX Bourne shell的单词解析规则来操作字符串。
- [shortid](#) - 分布式生成的超短，唯一，非顺序，URL友好的ID。
- [stateless](#) - 用于创建状态机的流畅库。
- [stats](#) - 监视Go MemStats和系统统计信息（例如内存，交换和CPU），并通过UDP发送到您想要记录的任何位置等。
- [turtle](#) - Go表情符号。
- [url-shortener](#) - 具有mysql支持的现代，强大而强大的URL缩短器微服务。
- [VarHandler](#) - 生成样板http输入和输出处理。
- [xdg](#) - FreeDesktop.org（XDG）规格的围棋实现。
- [xkg](#) - X键盘抓取器。
- [manba](#) - 轻量级网关

自然语言处理

自然语言处理

用于使用人类语言的图书馆。

- [getlang](#) - 快速自然语言检测程序包。
- [go-i18n](#) - 用于处理本地化文本的软件包和一个随附工具。
- [go-mystem](#) - CGo与Yandex.Mystem的绑定-俄罗斯形态分析仪。
- [go-nlp](#) - 用于处理离散概率分布的实用程序和其他可用于执行NLP工作的工具。
- [go-pinyin](#) - CN Hanzi至Hanyu拼音转换器。
- [go-stem](#) - 搬运程序阻止算法的实现。
- [go-unidecode](#) - Unicode文本的ASCII音译。
- [go2vec](#) - 用于word2vec嵌入的阅读器 and 实用程序功能。
- [gojieba](#) - 这是一个围棋实施解霸其中中国分词算法。
- [golibstemmer](#) - 雪球库libstemmer库的绑定，包括porter 2。
- [gotokenizer](#) - 基于字典和Goram语言的Bigram语言模型的标记器。（现在仅支持中文细分）
- [gounidecode](#) - Go的Unicode音译器（也称为unidecode）。
- [gse](#) - 进行有效的文本分割；支持英语，中文，日语等。
- [icu](#) - CGO结合为ICU4C C库检测和转换功能。保证与版本50.1兼容。
- [kagome](#) - 用纯Go语言编写的JP形态分析仪。
- [libtextcat](#) - libtextcat C库的Cgo绑定。保证与2.2版兼容。
- [MMSEGO](#) - 这是MMSEG的GO实现，它是中文分词算法。
- [nlp](#) - 从字符串中提取值，并用nlp填充您的结构。
- [nlp](#) - 支持LSA（潜在语义分析）的自然语言处理库。
- [paicehusk](#) - Paice / Husk提取算法的Golang实现。
- [petrovich](#) - 彼得罗维奇（Petrovich）是图书馆，在给定的语法情况下使用俄语名称。
- [porter](#) - 这是Martin Porter的Porter干算法的C实现的相当简单的移植。
- [porter2](#) - 非常快的Porter 2 提取器。
- [prose](#) - 用于文本处理的库，支持标记化，词性标记，命名实体提取等。仅限英语。
- [RAKE.go](#) - 快速自动关键字提取算法（RAKE）的Go端口。
- [segment](#) - 用于执行Unicode标准附件# 29中所述的Unicode文本分段的Go库
- [sentences](#) - 句子标记器：将文本转换为句子列表。
- [shamoji](#) - shamoji是用Go编写的单词过滤程序包。
- [snowball](#) - Go的雪球茎端口（cgo包装器）。提供单词词干提取功能Snowball本机。
- [stemmer](#) - 用于Go编程语言的Stemmer软件包。包括英语和德语词干。
- [textcat](#) -Go软件包，用于基于n-gram的文本分类，并支持utf-8和原始文本。
- [whatlanggo](#) - Go的自然语言检测程序包。支持84种语言和24种脚本（书写系统，例如拉丁语，西里尔字母等）。
- [when](#) - 自然EN和RU语言日期/时间分析器具有可插拔的规则。

网络

网络

与网络各层配合使用的库。

- [arp](#) - 包arp实现ARP协议，如RFC 826中所述。
- [buffstreams](#) - 通过TCP流化协议缓冲区数据变得容易。
- [canopus](#) - CoAP客户端/服务器实施（RFC 7252）。
- [cidranger](#) - Go的快速IP到CIDR查找。
- [dhcp6](#) - 软件包dhcp6实现了DHCPv6服务器，如RFC 3315中所述。
- [dns](#) - 使用DNS的Go库。
- [ether](#) - 用于发送和接收以太网帧的跨平台Go软件包。
- [ethernet](#) - 程序包ethernet实施IEEE 802.3以太网II帧和IEEE 802.1Q VLAN标签的封装处理。
- [fasthttp](#) - 软件包fasthttp是Go的一种快速HTTP实现，比net / http快10倍。
- [fortio](#) - 负载测试库和命令行工具，高级回显服务器和Web UI。允许指定设置的每秒查询负载，并记录延迟直方图和其他有用的统计数据并对其进行图形化。Tcp, Http, gRPC。
- [ftp](#) 程序包ftp实现RFC 959中所述的FTP客户端。
- [gev](#) - gev是基于Reactor模式的轻量级，快速，无阻塞的TCP网络库。
- [gmqtt](#) - Gmqtt是一个灵活的高性能MQTT代理库，它完全实现了MQTT协议V3.1.1。
- [gnet](#) - gnet是一个高性能的，用纯围棋轻便，非阻塞，事件循环网络库。
- [gNxI](#) - 使用gNMI和gNOI协议的网络管理工具的集合。
- [go-getter](#) - Go库，用于使用URL从各种来源下载文件或目录。
- [go-powerdns](#) - Golang的 PowerDNS API绑定。
- [go-stun](#) - STUN客户端的Go实现（RFC 3489和RFC 5389）。
- [gobgp](#) - 使用Go编程语言实现的BGP。
- [golibwireshark](#) - 软件包golibwireshark使用libwireshark库来解码pcap文件并分析解剖数据。
- [gopacket](#) - Go库，用于使用libpcap绑定进行数据包处理。
- [gopcap](#) - libpcap的包装器。
- [goshark](#) - 软件包goshark使用tshark解码IP数据包并创建数据结构以分析数据包。
- [gosnmp](#) - 用于执行SNMP操作的本地Go库。
- [gosocsvr](#) - 套接字服务器变得简单。
- [gotcp](#) - 用于快速编写tcp应用程序的Go软件包。
- [grab](#) - 用于管理文件下载的软件包。
- [graval](#) - 实验性FTP服务器框架。
- [HTTPLab](#) - HTTPLabs可让您检查HTTP请求并伪造响应。
- [iplib](#) - 受python ipaddress和ruby ipaddr启发而使用IP地址（net.IP, net.IPNet）的库
- [jazigo](#) - Jazigo是用Go语言编写的工具，用于检索多个网络设备的配置。
- [kcp-go](#) - KCP-快速可靠的ARQ协议。
- [kcptun](#) - 基于KCP协议的极其简单和快速的udp隧道。
- [lhttp](#) - 强大的websocket框架，可更轻松地构建IM服务器。
- [linkio](#) - 用于读取器/写入器接口的网络链接速度模拟。

- [llb](#) - 这是代理服务器的非常简单但快速的后端。对于零内存分配和快速响应的快速重定向到预定义域很有用。
- [mdns](#) - Golang中的简单mDNS（多播DNS）客户端/服务器库。
- [mqttPaho](#) - Paho Go客户端提供了一个MQTT客户端库，用于通过TCP，TLS或WebSockets连接到MQTT代理。
- [NFF-Go](#) - 用于快速开发云和裸机（以前的YANFF）的高性能网络功能的框架。
- [packet](#) - 通过TCP和UDP发送数据包。如果需要，它可以缓冲消息和热交换连接。
- [peerdiscovery](#) - Pure Go库，用于使用UDP多播的跨平台本地对等发现。
- [portproxy](#) - 简单的TCP代理，它将不支持它的API添加到CORS支持中。
- [publicip](#) - 软件包publicip返回您的面向公众的IPv4地址（互联网出口）。
- [quic-go](#) - 在纯Go中实现QUIC协议。
- [raw](#) - 包raw允许在设备驱动程序级别为网络接口读取和写入数据。
- [sftp](#) - 程序包sftp实现SSH文件传输协议，如<https://filezilla-project.org/specs/draft-ietf-secsh-filexfer-02.txt>
- [ssh](#) - 用于构建SSH服务器的高级API（包装crypto / ssh）。
- [sslb](#) - 这是一个超级简单的负载均衡器，只是一个实现某种性能的小项目。
- [stun](#) - 实施RFC 5389 STUN协议。
- [tcp_server](#) - 用于更快地构建tcp服务器的Go库。
- [tspool](#) - TCP库使用工作池来提高性能并保护您的服务器。
- [utp](#) - 围棋UTP微传输协议的实现。
- [water](#) - 简单的TUN / TAP库。
- [webrtc](#) - WebRTC API的纯Go实现。
- [winrm](#) - 进入WinRM客户端以在Windows计算机上远程执行命令。
- [xtcp](#) - 具有同步全双工通信，安全关闭，自定义协议的TCP Server Framework。

HTTP客户端

HTTP客户端

用于发出HTTP请求的库。

- [gentleman](#) - 功能齐全的插件驱动的HTTP客户端库。
- [grequests](#) - 著名的Requests库的Go克隆。
- [heimdall](#) - 具有重试和hystrix功能的增强型http客户端。
- [pester](#) - 通过重试，退避和并发进行HTTP客户端调用。
- [resty](#) - 受Ruby rest-client启发的Go的简单HTTP和REST客户端。
- [rq](#) - golang stdlib HTTP客户端的更好接口。
- [sling](#) - Sling是用于创建和发送API请求的Go HTTP客户端库。
- [sreq](#) - Go的简单，用户友好和并发安全的HTTP请求库。

OpenGL

OpenGL

在Go中使用OpenGL的库。

- [gl](#) - 用于OpenGL的绑定（通过glow生成）。
- [glfw](#) - GLFW 3的绑定。
- [goxjs/gl](#) - 跨平台的OpenGL绑定（OS X, Linux, Windows, 浏览器, iOS, Android）。
- [goxjs/glfw](#) - 跨平台的glfw库，用于创建OpenGL上下文并接收事件。
- [mathgl](#) - 受GLM启发，专门用于3D数学的Pure Go数学软件包。

ORM

ORM

实现对象关系映射或数据映射技术的库。

- [beego orm](#) - 强大的orm框架。支持: pq / mysql / sqlite3。
- [go-firebase](#) - Google / Firebase Cloud Firestore的简单ORM。
- [go-pg](#) - PostgreSQL ORM, 专注于PostgreSQL的特定功能和性能。
- [go-queryset](#) - 具有代码生成和基于GORM的MySQL, PostgreSQL, Sqlite3, SQL Server支持的100%类型安全的ORM。
- [go-sqlbuilder](#) - 灵活而强大的SQL字符串构建器库以及零配置ORM。
- [go-store](#) - 用于Go的简单快速的Redis支持的键值存储库。
- [GORM](#) - 这些巨大的ORM库Golang, 目的是开发者友好。
- [gorp](#) - Go Relational Persistence, 用于Go的ORM类库。
- [grimoire](#) - Grimoire是数据库访问层和golang验证。(支持: MySQL, PostgreSQL和SQLite3)。
- [lore](#) - 用于Go的简单轻量级的伪ORM /伪结构映射环境。
- [Marlow](#) - 从项目结构生成的ORM, 用于编译时安全保证。
- [pop/soda](#) - MySQL, PostgreSQL和SQLite的数据库迁移, 创建, ORM等。
- [QBS](#) - 代表按结构查询。Go ORM。
- [reform](#) - 基于非空接口和代码生成的更好的ORM for Go。
- [SQLBoiler](#) - ORM生成器。生成针对您的数据库模式量身定制的功能强大且快速的ORM。
- [upper.io/db](#) - 用于使用包装成熟的数据库驱动程序的适配器与不同数据源进行交互的单个接口。
- [Xorm](#) - Go的简单而强大的ORM。
- [Zoom](#) - 超快数据存储和查询引擎建立在Redis的。

包管理

包管理

依赖和程序包管理的官方工具

- [go modules](#) - 模块是源代码交换和版本控制的单元。`go`命令直接支持使用模块，包括记录和解决对其他模块的依赖关系。

用于软件包管理的官方实验工具

- [dep](#) - Go依赖工具。
- [vgo](#) - 版本化的Go。

用于包和依赖项管理的非官方库。

- [gigo](#) - 用于golang的类似于PIP的依赖项工具，支持私有存储库和哈希。
- [glide](#) - 轻松管理您的golang供应商和供应商软件包。受到Maven，Bundler和Pip等工具的启发。
- [godep](#) - go的依赖工具，`godep`通过修复程序包的依赖关系可重复构建软件包。
- [gom](#) - Go Manager-go捆绑包。
- [goop](#) - 简单的依赖经理围棋（golang），通过捆扎机的启发。
- [gop](#) - 在GOPATH之外构建和管理Go应用程序。
- [gopm](#) - 转到软件包管理器。
- [govendor](#) - Go软件包管理器。使用标准供应商文件的Go供应商工具。
- [gpm](#) - 准系统Go的准系统依赖管理器。
- [johnny-deps](#) - 使用Git的最小依赖版本。
- [mvn-golang](#) - 插件，用于自动加载Golang SDK，依赖管理并在Maven项目基础架构中启动构建环境。
- [nut](#) - Vendor Go依赖项。
- [VenGO](#) - 创建和管理可导出的隔离go虚拟环境。

性能

性能

- [jaeger](#) - 分布式跟踪系统。
- [profile](#) - Go的简单配置支持包。
- [tracer](#) - 简单，轻便的跟踪。

查询语言

查询语言

- [gojsonq](#) - 一个用于查询JSON数据的简单Go包。
- [graphql](#) - graphql解析器和实用程序。
- [graphql](#) - 专注于易用性的GraphQL服务器。
- [graphql-go](#) - GraphQL for Go的实现。
- [jsonql](#) - Golang中的JSON查询表达式库。
- [jsonslice](#) - 具有高级过滤器的Jsonpath查询。
- [rql](#) - REST API的资源查询语言。
- [straf](#) - 轻松将Golang结构转换为GraphQL对象。

资源嵌入

资源嵌入

- [esc](#) - 嵌入功能文件到电脑围棋程序，并提供[http.FileSystem](#)接口给他们。
- [fileb0x](#) - 嵌入文件的简单工具，重点放在“自定义”上，易于使用。
- [go-embed](#) - 生成go代码以将资源文件嵌入到您的库或可执行文件中。
- [go-resources](#) - Go中嵌入的多余资源。
- [go.rice](#) - [go.rice](#)是一个Go程序包，可轻松处理html, js, css, 图像和模板等资源。
- [packr](#) - 将静态文件嵌入Go二进制文件的简便方法。
- [statics](#) - 将静态资源嵌入go文件中以进行单个二进制编译+与[http.FileSystem](#) +符号链接一起使用。
- [statik](#) - 将静态文件嵌入Go可执行文件。
- [templify](#) - 将外部模板文件嵌入Go代码中以创建单个文件二进制文件。
- [vfsgen](#) - 生成一个[vfsgen.go](#)文件，该文件静态实现给定的虚拟文件系统。

科学与数据分析

科学与数据分析

用于科学计算和数据分析的包。

- [assocentity](#) - 包关联性返回单词到给定实体的平均距离。
- [bradleyterry](#) - 提供Bradley-Terry模型进行成对比较。
- [chart](#) - Go的简单图表绘图库。支持多种图形类型。
- [dataframe-go](#) - 用于机器学习和统计的数据框（类似于熊猫）。
- [evaler](#) - 简单的浮点算术表达式评估器。
- [ewma](#) - 指数加权移动平均值。
- [geom](#) - golang的2D几何。
- [go-dsp](#) - Go的数字信号处理。
- [go-gt](#) - 用“Go”语言编写的图论算法。
- [goent](#) - 熵测度的GO实施。
- [gohistogram](#) - 数据流的近似直方图。
- [gonum](#) - Gonum是Go编程语言的一组数字库。它包含用于矩阵，统计信息，优化等的库。
- [gonum/plot](#) - gonum / plot提供了一个用于在Go中构建和绘制图的API。
- [goraph](#) - Pure Go图论库（数据结构，算法可视化）。
- [gosl](#) - Go科学图书馆，提供线性代数，FFT，几何，NURBS，数值方法，概率，优化，微分方程等。
- [GoStats](#) - GoStats是用于数学统计的开源GoLang库，主要用于机器学习领域，它涵盖了大多数统计度量功能。
- [graph](#) - 基本图形算法库。
- [ode](#) - 常微分方程（ODE）求解器，支持扩展状态和基于通道的迭代停止条件。
- [orb](#) - 二维几何类型与剪裁外，GeoJSON和Mapbox矢量砖支撑。
- [pagerank](#) - 在Go中实现的加权PageRank算法。
- [piecwiselinear](#) - 微小的线性插值库。
- [PiHex](#) - 十六进制数Pi的“Bailey-Borwein-Plouffe”算法的实现。
- [rootfinding](#) - 用于查找二次函数根的寻根算法库。
- [sparse](#) - 用于线性代数的Go稀疏矩阵格式，支持科学和机器学习应用程序，与gonum矩阵库兼容。
- [stats](#) - Golang标准库中缺少常用功能的统计信息包。
- [streamtools](#) - 用于处理数据流的通用图形工具。
- [TextRank](#) - Golang中的TextRank实现，具有可扩展的功能（摘要，加权，短语提取）和多线程（goroutine）支持。
- [triangolatte](#) - 2D三角剖分库。允许将线和多边形（均基于点）转换为GPU的语言。

安全

安全

用于帮助您您的应用程序更安全的库。

- [acmetool](#) - A具有自动续订功能的ACME（让我们加密）客户端工具。
- [acra](#) - 网络加密代理，可保护基于数据库的应用程序免遭数据泄漏：强大的选择性加密，SQL注入预防，入侵检测系统。
- [argon2pw](#) - 具有恒定时间密码比较的Argon2密码哈希生成。
- [autocert](#) - 自动配置让我们加密证书并启动TLS服务器。
- [BadActor](#) - 内存中的，由应用程序驱动的监牢程序，本着fail2ban的精神构建。
- [Cameradar](#) - 从远程监控摄像机远程破解RTSP流的工具和库。
- [certificates](#) - 用于生成tls证书的经过验证的工具。
- [go-generate-password](#) - 密码生成器，可以在cli上使用或用作库。
- [go-yara](#) - YARA的 Go Bindings ，“用于恶意软件研究人员（和其他所有人）的模式匹配瑞士刀”。
- [goArgonPass](#) - Argon2密码哈希和验证，旨在与现有的Python和PHP实现兼容。
- [goSecretBoxPassword](#) - 一个可能偏执的软件包，用于安全地哈希和加密密码。
- [Interpol](#) - 基于规则的数据生成器，用于模糊测试和渗透测试。
- [lego](#) - Pure Go ACME客户端库和CLI工具（与Let's Encrypt一起使用）。
- [memguard](#) - 用于处理内存中敏感值的纯Go库。
- [nacl](#) - NaCL API集的实现。
- [passlib](#) - 永不过时的密码哈希库。
- [secure](#) - 用于Go的HTTP中间件，有助于快速赢得安全性。
- [simple-scrypt](#) - scrypt-具有简单，显而易见的API和内置自动成本校准功能的Scrypt软件包。
- [ssh-vault](#) - vault-使用ssh密钥加密/解密。
- [sslmgr](#) - 围绕acme / autocert的高级包装使SSL证书变得容易。

序列化

序列化

库和二进制序列化工具。

- [asn1](#) - 用于golang的Asn.1 BER和DER编码库。
- [bambam](#) - 开始使用Cap'n Proto模式的生成器。
- [bel](#) - 生成围棋结构/接口打字稿接口。对JSON RPC有用。
- [binstruct](#) - Golang二进制解码器，用于将数据映射到结构中。
- [cbor](#) - 小型，安全和容易的CBOR编码和解码库。
- [colfer](#) - Colfer二进制格式的代码生成。
- [csvutil](#) - 对本地Go结构的高性能，惯用CSV记录编码和解码。
- [fwencoder](#) - Go的固定宽度文件解析器（编码和解码库）。
- [go-capnproto](#) - Cap'n Proto库和go解析器。
- [go-codec](#) - 用于msgpack, cbor和json的高性能，功能丰富，惯用的编码，解码和rpc库，具有基于运行时的OR代码生成支持。
- [gogoprotobuf](#) - Go with Gadgets的协议缓冲区。
- [goprotobuf](#) - 以库和协议编译器插件的形式提供对Google协议缓冲区的支持。
- [jsoniter](#) - 以库和协议编译器插件的形式提供对Google协议缓冲区的支持。
- [mapstructure](#) - Go库，用于将通用地图值解码为本地Go结构。
- [php_session_decoder](#) - GoLang库，用于使用PHP会话格式和PHP序列化/反序列化功能。
- [pletter](#) - 为消息代理包装原始消息的一种标准方法。
- [structomap](#) - 从静态结构轻松动态地生成地图的库。

服务器应用

服务器应用

- [algeron](#) - HTTP / 2 Web服务器，内置对Lua，Markdown，GCSS和Amber的支持。
- [Caddy](#) - Caddy是易于配置和使用的备用HTTP / 2 Web服务器。
- [consul](#) - Consul是用于服务发现，监视和配置的工具。
- [devd](#) - 开发人员的本地网络服务器。
- [discovery](#) - 用于弹性中间层负载均衡和故障转移的注册表。
- [dudeldu](#) - 一个简单的SHOUTcast服务器。
- [etcd](#) - 高度可用的键值存储，用于共享配置和服务发现。
- [Fider](#) - Fider是一个开放平台，用于收集和组织的客户反馈。
- [Flagr](#) - Flagr是开源功能标记和A / B测试服务。
- [flipt](#) - 用Go和Vue.js编写的自包含功能标志解决方案
- [jackal](#) - 用Go编写的XMPP服务器。
- [minio](#) - Minio是分布式对象存储服务器。
- [nginx-prometheus](#) - Nginx日志解析器和Prometheus的导出器。
- [nsq](#) - 实时分布式消息传递平台。
- [psql-streamer](#) - 将数据库事件从PostgreSQL流化到Kafka。
- [riemann-relay](#) - 中继以实现负载均衡Riemann事件和/或将其转换为Carbon。
- [RoadRunner](#) - 高性能PHP应用程序服务器，负载均衡器和进程管理器。
- [SFTPGo](#) - 功能齐全且可高度配置的SFTP服务器软件。
- [yakvs](#) - 小型网络内存中键值存储。

流处理

流处理

用于流处理和反应式编程的库和工具。

- [go-streams](#) - Go流处理库。

模板引擎

模板引擎

用于模板化和词汇化的库和工具。

- [ace](#) - Ace是Go语言的HTML模板引擎，其灵感来自Slim和Jade。王牌是黄金的提炼。
- [amber](#) - 琥珀色是Go编程语言的优雅模板引擎，其灵感来自HAML和Jade。
- [damsel](#) - 标记语言，具有通过css-selectors进行html概述的功能，可通过pkg html / template等扩展。
- [ego](#) - 轻量级的模板语言，可让您在Go中编写模板。模板被翻译成Go并被编译。
- [extemplate](#) - 围绕html / template的微型包装器，可轻松实现基于文件的模板继承。
- [fasttemplate](#) - 简单快速的模板引擎。替换模板占位符的速度比text/template快10倍。
- [gofpdf](#) - 具有高级支持的文本，图形和图像的PDF文档生成器。
- [gospin](#) - 文章旋转和spintax / spinning语法引擎，适用于A / B，测试文本/文章以及创建更自然的对话。
- [goview](#) - Goview是一个基于golang html / template的轻量级，极简且惯用的模板库，用于构建Go Web应用程序。
- [hero](#) - Hero是一个方便，快速且强大的go模板引擎。
- [jet](#) - Jet模板引擎。
- [kasia.go](#) - HTML和其他文本文档的模板系统-执行。
- [liquid](#) - Shopify Liquid模板的实现。
- [maroto](#) - 一种创建PDF的maroto方法。Maroto受Bootstrap启发，并使用gofpdf。快速简单。
- [mustache](#) - Mustache模板语言的实现。
- [pongo2](#) - Go的类似Django的模板引擎。
- [quicktemplate](#) - 快速，强大但易于使用的模板引擎。将模板转换为Go代码，然后进行编译。
- [raymond](#) - Go中完整的车把实现。
- [Razor](#) - Golang的Razor视图引擎。
- [Soy](#) - 遵循官方规格的 Go 大豆封闭模板（也称为大豆模板）。
- [velvet](#) - Go中完整的车把实现。

测试

测试

用于测试代码库和生成测试数据的库。

- 测试框架
 - [apitest](#) - 简单和可扩展的行为测试库，用于基于REST的服务或HTTP处理程序，支持模拟外部http调用和序列图的呈现。
 - [assert](#) - 与本地本机go测试一起使用的基本断言库，带有用于自定义断言的构建块。
 - [badio](#) - Go testing/iotest包的扩展。
 - [baloo](#) - 表达性和通用性的端到端HTTP API测试变得容易。
 - [biff](#) - 分叉测试框架，与BDD兼容。
 - [charlatan](#) - 生成用于测试的虚假接口实现的工具。
 - [commander](#) - 用于在Windows，Linux和osx上测试cli应用程序的工具。
 - [cupaloy](#) - 适用于您的测试框架的简单快照测试插件。
 - [dbcleaner](#) - 受database_cleanerRuby启发，清理数据库以进行测试。
 - [dsunit](#) - SQL，NoSQL，结构化文件的数据存储区测试。
 - [embedded-postgres](#) - Postgres-作为另一个Go应用程序或测试的一部分，在Linux，OSX或Windows上本地运行真实的Postgres数据库。
 - [endly](#) - 声明式端到端功能测试。
 - [flute](#) - HTTP客户端测试框架。
 - [frisby](#) - REST API测试框架。
 - [ginkgo](#) - Go的BDD测试框架。
 - [go-carpet](#) - 用于在终端中查看测试覆盖率的工具。
 - [go-cmp](#) - 用于比较测试中Go值的软件包。
 - [go-mutesting](#) - Go源代码的变异测试。
 - [go-testdeep](#) - 极其灵活的golang深度比较，扩展了go测试包。
 - [go-vcr](#) - 记录并重放您的HTTP交互，以进行快速，确定性和准确的测试。
 - [goblin](#) - 摩卡（Mocha）像Go的测试框架。
 - [gocheck](#) - 比getest更高级的测试框架。
 - [GoConvey](#) - 具有Web UI和实时重载的BDD样式的框架。
 - [gocrest](#) - Go断言的可组合的类似于hamcrest的匹配器。
 - [godog](#) - Cucumber或Behat像Go的BDD框架。
 - [gofight](#) - Golang路由器框架的API处理程序测试。
 - [gogiven](#) - Go的类似YATSPEC的BDD测试框架。
 - [gomatch](#) - 创建用于针对模式测试JSON的库。
 - [gomega](#) - Rspec，例如匹配器/断言库。
 - [GoSpec](#) - 用于Go编程语言的BDD样式测试框架。
 - [gospecify](#) - 这提供了BDD语法来测试您的Go代码。使用过rspec之类的库的任何人都应该熟悉。
 - [gosuite](#) - testing利用Go1.7的子测试，使带有设置/拆卸功能的轻量级测试套件成为可能。

- [gotest.tools](#) - 一组软件包的集合，以增强go测试软件包并支持常见模式。
 - [Hamcrest](#) - 用于声明式Matcher对象的流利框架，将其应用于输入值时，会产生自描述结果。
 - [httpexpect](#) - 简洁，声明性且易于使用的端到端HTTP和REST API测试。
 - [jsonassert](#) - 用于验证JSON有效负载已正确序列化的软件包。
 - [restit](#) - 一个微框架，可帮助编写RESTful API集成测试。
 - [schema](#) - 在请求和响应中使用JSON模式方便快捷则表达式匹配。
 - [testcase](#) - 行为驱动开发的惯用测试框架。
 - [testfixtures](#) - Rails的测试夹具，用于测试数据库应用程序。
 - [Testify](#) - 对标准go测试包的神圣扩展。
 - [testmd](#) - 将markdown代码片段转换为可测试的go代码。
 - [testsql](#) - 在测试之前从SQL文件生成测试数据，并在完成后将其清除。
 - [trial](#) - 快速简单的可扩展断言，无需引入太多样板。
 - [Tt](#) - 简单而丰富多彩的测试工具。
 - [wstest](#) - WebSocket客户端，用于对websocket http.Handler进行单元测试。
- Mock
 - [counterfeiter](#) - 生成独立模拟对象的工具。
 - [go-sqlmock](#) - 模拟SQL驱动程序，用于测试数据库交互。
 - [go-txdb](#) - 基于单事务的数据库驱动程序，主要用于测试目的。
 - [gock](#) - 多种HTTP 模拟变得容易。
 - [gomock](#) - Go编程语言的模拟框架。
 - [govcr](#) - 用于Golang的HTTP模拟：记录和重放HTTP交互以进行脱机测试。
 - [hoverfly](#) - HTTP (S) 代理，用于通过可扩展的中间件和易于使用的CLI记录和模拟REST / SOAP API。
 - [httpmock](#) - 轻松模拟来自外部资源的HTTP响应。
 - [minimock](#) - Go接口的模拟生成器。
 - [mockhttp](#) - Go http.ResponseWriter的模拟对象。
 - Fuzzing and delta-debugging/reducing/shrinking.
 - [go-fuzz](#) - 随机测试系统。
 - [gofuzz](#) - 用于填充带有随机值的go对象的库。
 - [Tavor](#) - 通用的模糊测试和增量调试框架。
 - Selenium and browser control tools.
 - [cdp](#) - Chrome调试协议的类型安全绑定，可与实现该协议的浏览器或其他调试目标一起使用。
 - [chromedp](#) - 一种驱动/测试Chrome, Safari, Edge, Android Webview和其他支持Chrome调试协议的浏览器的方法。
 - [ggr](#) - 轻量级服务器，将Selenium WebDriver请求路由和代理到多个Selenium集线器。
 - [selenoid](#) - 替代的Selenium集线器服务器，可在容器内启动浏览器。
 - Fail injection
 - [failpoint](#) - failpoint -的实现failpoints为Golang。

文字处理

文字处理

用于解析和处理文本的库。

- 特定格式
 - [align](#) - 对齐文本的通用应用程序。
 - [allot](#) - CLI工具和漫游器的占位符和通配符文本解析。
 - [bbConvert](#) - 将bbCode转换为HTML，使您可以添加对自定义bbCode标记的支持。
 - [blackfriday](#) - Go中的Markdown处理器。
 - [bluemonday](#) - HTML清理工具。
 - [codetree](#) - 解析缩进的代码（python, pixy, scarlet等）并返回树结构。
 - [colly](#) - Gophers的快速优雅的刮取框架。
 - [commonregex](#) - Go的通用正则表达式的集合。
 - [dataflowkit](#) - Web抓取框架，用于将网站转换为结构化数据。
 - [did](#) - Go中的DID（分散标识符）解析器和Stringer。
 - [doi](#) - Go中的文档对象标识符（doi）解析器。
 - [editorconfig-core-go](#) - Go的Editorconfig文件解析器和操纵器。
 - [enca](#) - libenca的最小cgo绑定。
 - [encdec](#) - 程序包为编码器和解码器提供通用接口。
 - [genex](#) - 计算正则表达式并将其扩展为所有匹配的字符串。
 - [github_flavored_markdown](#) - GitHub Flavored Markdown渲染器（使用blackfriday），带有围栏代码块突出显示，可单击的标题锚链接。
 - [go-fixedwidth](#) - 固定宽度的文本格式（带反射的编码器/解码器）。
 - [go-humanize](#) - 将时间，数字和内存大小的格式化程序转换为可读格式。
 - [go-nmea](#) - 用于Go语言的NMEA解析器库。
 - [go-runewidth](#) - 获取固定宽度的字符或字符串的函数。
 - [go-slugify](#) - 通过多种语言支持，让您轻松自如。
 - [go-toml](#) - 具有查询支持和方便的cli工具的TOML格式的Go库。
 - [go-vcard](#) - 解析和格式化vCard。
 - [go-zero-width](#) - Go的零宽度字符检测和删除。
 - [gofeed](#) - 在Go中解析RSS和Atom提要。
 - [gographviz](#) - 解析Graphviz DOT语言。
 - [gommon/bytes](#) - 将字节格式化为字符串。
 - [gonameparts](#) - 将人的名字解析成单独的名字部分。
 - [goq](#) - 使用带有jQuery语法的struct标签的HTML声明式解封（使用GoQuery）。
 - [GoQuery](#) - GoQuery为Go语言带来了一种类似于jQuery的语法和一组功能。
 - [goregen](#) - 用于从正则表达式生成随机字符串的库。
 - [goribot](#) - 一个简单的golang爬虫/抓取框架，分3行构建一个蜘蛛。
 - [gotext](#) - Go的GNU gettext实用程序。

- [guesslanguage](#) - 确定unicode文本的自然语言的功能。
 - [htmlquery](#) - 用于HTML的XPath查询包，可让您通过XPath表达式从HTML文档中提取数据或评估。
 - [inject](#) - 包注入提供基于反射的注入器。
 - [ltsv](#) - 用于Go的高性能LTSV（标签制表符分隔值）阅读器。
 - [mxj](#) - 将XML编码/解码为JSON或map [string] interface {}; 提取带点符号路径和通配符的值。替换x2j和j2x软件包。
 - [sdp](#) - SDP: 会话描述协议[RFC 4566]。
 - [sh](#) - Shell解析器和格式化程序。
 - [slug](#) - URL友好slugify多语言支持。
 - [Slugify](#) - 去处理字符串的Slugify应用程序。
 - [syndfeed](#) - Atom 1.0和RSS 2.0的联合供稿。
 - [toml](#) - TOML配置格式（带反射的编码器/解码器）。
- Utility
 - [gofuckyourself](#) - 用于Go的基于消毒的发誓过滤器。
 - [gotabulate](#) - 使用Go轻松漂亮地打印表格数据。
 - [kace](#) - 常见情况转换，涵盖常见的缩写。
 - [parseargs-go](#) - 理解引号和反斜杠的字符串参数解析器。
 - [parth](#) - URL路径分段解析。
 - [radix](#) - 快速字符串排序算法。
 - [Tagify](#) - 从给定的来源产生一组标签。
 - [textwrap](#) - textwrap从Python 实现模块。
 - [TySug](#) - 关于键盘布局的替代建议。
 - [xj2go](#) - 将xml或json转换为struct。
 - [xurls](#) - 从文本中提取URL。

第三方API

第三方API

用于访问第三方API的库。

- [amazon-product-advertising-api](#) - 用于Amazon Product Advertising API的 Go客户端库。
- [anaconda](#) - Twitter 1.1 API的Go客户端库。
- [aws-sdk-go](#) - 用于Go编程语言的官方AWS开发工具包。
- [brewerydb](#) -用于访问BreweryDB API的Go库。
- [cachet](#) - 用于Cachet（开放源代码状态页面系统）的客户端库。
- [circleci](#) - Go客户端库，用于与CircleCI的API进行交互。
- [clarifai](#) - Go客户端库，用于与Clarifai API接口。
- [codeship-go](#) - Go客户端库，用于与Codeship的API v2进行交互。
- [coinpaprika-go](#) - 用于与Coinpaprika的API交互的客户端库。
- [discordgo](#) - Discord Chat API的绑定。
- [ethrpc](#) - 以太坊JSON RPC API的绑定。
- [facebook](#) - 支持Facebook Graph API的Go库。
- [fcm](#) - 用于Firebase云消息传递的库。
- [gads](#) - Google Adwords非官方API。
- [gami](#) - Asterisk Manager界面的Go库。
- [gcm](#) - 用于Google Cloud Messaging的Go库。
- [geo-golang](#) - 访问Google Maps, MapQuest, Nominatim, OpenCage, Bing, Mapbox和OpenStreetMap地理编码/反向地理编码API的图书馆。
- [github](#) - Go库，用于访问GitHub REST API v3。
- [githubql](#) - 用于访问GitHub GraphQL API v4的Go库。
- [go-chronos](#) - 用于与Chronos Job Scheduler 交互的Go库
- [go-hacknews](#) - HackerNews API的Tiny Go客户端。
- [go-here](#) - 围绕基于HERE位置的API的客户端库。
- [go-imgur](#) - 用于imgur的客户端库
- [go-jira](#) - Atlassian JIRA的 Go客户库
- [go-marathon](#) - 与Mesosphere的Marathon PAAS交互的Go库。
- [go-myanimelist](#) - 用于访问MyAnimeList API的 Go客户端库。
- [go-sophos](#) - 具有零依赖关系的Sophos UTM REST API的 Go客户端库。
- [go-sptrans](#) -SPTrans Olho Vivo API的Go客户端库。
- [go-telegraph](#) - 电报发布平台API客户端。
- [go-trending](#) - Go库，用于访问Github上的热门存储库和开发人员。
- [go-twitch](#) - Go客户端，用于与Twitch v3 API进行交互。
- [go-twitter](#) - Twitter v1.1 API的Go客户端库。
- [go-unsplash](#) - 围棋客户端库的Unsplash.com API。
- [go-xkcd](#) - xkcd API的Go客户端。
- [golyrics](#) - Golyrics是一个Go库，用于从Wikia网站获取音乐歌词数据。

- [gomalshare](#) - Go库MalShare API malshare.com
- [GoMusicBrainz](#) - Go MusicBrainz WS2客户端库。
- [google](#) - 为Go自动生成的Google API。
- [google-analytics](#) - 简单的包装器，可轻松生成Google Analytics（分析）报告。
- [google-cloud](#) - Google Cloud API Go客户端库。
- [google-email-audit-api](#) - Google G Suite电子邮件审核API的客户端库。
- [gostorm](#) - GoStorm是一个Go库，实现了在Go中编写与Storm外壳进行通信的Storm喷口和螺栓所需的通信协议。
- [hipchat](#) - 该项目为Hipchat API实现了golang客户端库。
- [hipchat \(xmpp\)](#) - 用于通过XMPP与HipChat进行通信的golang程序包。
- [igdb](#) - Internet游戏数据库API的 Go客户端。
- [lastpass-go](#) - 用于LastPass API的Go客户端库。
- [libgoffi](#) - 用于本机libffi集成的库适配器工具箱
- [Medium](#) - Golang SDK为中等的了OAuth2 API。
- [megos](#) - 用于访问Apache Mesos集群的客户端库。
- [minio-go](#) - 用于Amazon S3兼容云存储的Minio Go库。
- [mixpanel](#) - Mixpanel是一个用于跟踪事件并从go应用程序向Mixpanel发送Mixpanel配置文件更新的库。
- [patreon-go](#) - 用于Patreon API的Go库。
- [paypal](#) - PayPal付款API的包装。
- [playlyfe](#) - Playlyfe Rest API Go SDK。
- [pushover](#) - Pushover API的包装器。
- [rrdaclient](#) - 转到库以访问statdns.com API，后者又是RRDA API。通过HTTP的DNS查询
- [shopify](#) - 转到库向Shopify API发出CRUD请求。
- [simples3](#) - 使用REST和Go语言编写的V4签名，使AWS S3库变得简单而简洁。
- [slack](#) - Go中的松弛API。
- [smite](#) - Go包，用于包装对Smite游戏API的访问。
- [spotify](#) - 转到库以访问Spotify WEB API。
- [steam](#) - 转到Library与Steam游戏服务器进行交互。
- [stripe](#) - Stripe API的客户端。
- [textbelt](#) - 消息传递API的Go客户端。
- [translate](#) - 在线翻译包。
- [Trello](#) - Trello API的包装器。
- [TripAdvisor](#) - TripAdvisor API的包装器。
- [tumblr](#) - Tumblr v2 API的包装器。
- [uptimerobot](#) - Uptime Robot v2 API的包装器和命令行客户端。
- [vl-go](#) - 围绕VerifiD身份验证层API的Go客户端库。
- [webhooks](#) - GitHub和Bitbucket的Webhook接收器。
- [wit-go](#) - 围棋客户端wit.ai HTTP API。
- [ynab](#) - YNAB API的包装器。
- [zooz](#) - Go Zooz API的Go客户端。

实用工具

实用工具

通用工具和工具使您的生活更轻松。

- [apm](#) - 用于Golang应用程序与HTTP API过程管理器。
- [backscanner](#) - 一种类似于bufio.Scanner的扫描仪，但是它以相反的顺序读取和返回行，从给定位置开始并向后移动。
- [beyond](#) - Go工具将带您进入AOP世界！
- [blank](#) - 验证或删除字符串中的空格和空格。
- [boilr](#) - 十分快速的CLI工具，用于从样板模板创建项目。
- [chyle](#) - 使用git存储库的Changelog生成器，具有多种配置可能性。
- [circuit](#) - 高效和完整的功能堪比去执行断路器的格局。
- [circuitbreaker](#) - 断路器在Go。
- [clockwork](#) - 用于golang的简单假时钟。
- [cmd](#) - 用于在osx, windows和linux上执行shell命令的库。
- [command](#) - 命令模式的围棋与线程安全的串行和并行调度。
- [copy-pasta](#) - 通用多工作站剪贴板，使用后端等S3进行存储。
- [ctop](#) - Top-like 顶状接口（例如HTOP），用于容器的指标。
- [ctxutil](#) - A 上下文的实用程序功能的集合。
- [dbt](#) - 一种用于从中央可信存储库运行自更新签名二进制文件的框架。
- [Death](#) - 通过信号管理go应用程序关闭。
- [Deeppcopier](#) - Go的简单结构复制。
- [delve](#) - Go 调试器。
- [dlog](#) - 编译时控制的记录器，可在不删除调试调用的情况下缩小发行版。
- [ergo](#) - 简化在不同端口上运行的多个本地服务的管理。
- [evaluator](#) - 根据s表达式动态求值。它很容易扩展。
- [filetype](#) - 小包，用于推断检查魔术数字签名的文件类型。
- [filler](#) - 使用“填充”标签的小工具来填充结构。
- [filter](#) - 提供过滤，消毒，和转到数据的转换。
- [fzf](#) - 用Go编写的命令行模糊查找器。
- [gaper](#) - 在Go项目崩溃或某些观看的文件更改时生成并重新启动Go项目。
- [generate](#) - 运行go在指定的路径或环境变量上递归生成，并且可以通过正则表达式进行过滤。
- [ghokin](#) - 并行化的格式化程序，没有黄瓜的外部依赖关系（黄瓜，behat ...）。
- [git-time-metric](#) - Git的简单，无缝，轻量级时间跟踪。
- [go-astitodo](#) - 在您的GO代码中解析TODO。
- [go-bind-plugin](#) - go: generate工具，用于包装由golang插件导出的符号（仅限1.8）。
- [go-bsdiff](#) - 纯Go bsdiff和bspatch库以及CLI工具。
- [go-dry](#) - Go 的干燥包装（请勿重复）。
- [go-funk](#) - 现代Go实用程序库，提供了帮助程序（映射，查找，包含，过滤器，块，反向等）。
- [go-health](#) - 运行状况软件包简化了将运行状况检查添加到服务中的方式。
- [go-httpheader](#) - 用于将结构编码为Header字段的Go库。

- [go-problemdetails](#) - 用于处理“问题详细信息”的Go软件包。
- [go-rate](#) - Go的定时速率限制器。
- [go-sitemap-generator](#) - generator-用Go编写的XML Sitemap生成器。
- [go-trigger](#) - Go-lang全局事件触发程序，使用ID注册事件并从项目的任何位置触发事件。
- [goback](#) - 使用简单的指数补偿程序包。
- [godaemon](#) - 编写守护程序的实用程序。
- [godropbox](#) - 用于从Dropbox编写Go服务/应用程序的通用库。
- [gohper](#) - 各种工具/模块有助于开发。
- [golarm](#) - 具有系统事件的火灾警报。
- [golog](#) - 简单，轻量级的CLI工具，可定时跟踪您的任务。
- [gopencils](#) - 小型简单的程序包，可轻松使用REST API。
- [goplaceholder](#) - 一个小的golang库，用于生成占位符图像。
- [goreadability](#) - 使用Facebook Open Graph和arc90的可读性的网页摘要提取器。
- [goreleaser](#) - 尽可能快速，轻松地交付Go二进制文件。
- [goreporter](#) - Golang工具，可进行静态分析，单元测试，代码审查并生成代码质量报告。
- [goseaweedfs](#) - 具有几乎全部功能的SeaweedFS客户端库。
- [gostrutils](#) - 字符串操作和转换函数的集合。
- [gotenv](#) - 从Go .env或任何io.ReaderGo中加载环境变量。
- [gpath](#) - 使用Go的反射形式简化访问结构字段的库。
- [gubrak](#) - Golang实用程序库，带有语法糖。就像lodash，但对于golang。
- [handy](#) - 字符串处理程序/格式化程序和验证程序等许多实用程序和帮助程序。
- [htcat](#) - 并行和流水线HTTP GET实用程序。
- [hub](#) - 将git命令包装为具有其他功能的功能，以便从终端与github进行交互。
- [hystrix-go](#) - 实现程序员定义的备用（又称为断路器）的Hystrix模式。
- [immortal](#) - `* nix` 跨平台（不可知操作系统）主管。
- [intrinsic](#) - 使用x86 SIMD无需编写任何汇编代码。
- [jump](#) - 跳跃可帮助您通过学习习惯来更快地导航。
- [koazee](#) - 受到懒惰评估和函数式编程启发的库，消除了使用数组的麻烦。
- [limiters](#) - Golang中分布式应用程序的速率限制器，具有可配置的后端和分布式锁。
- [lrserver](#) - Go的LiveReload服务器。
- [mc](#) - Minio Client提供了最少的工具来与Amazon S3兼容的云存储和文件系统一起使用。
- [mergo](#) - 在Golang中合并结构和地图的助手。对于配置默认值很有用，避免了混乱的if语句。
- [mimemagic](#) - Pure Go高性能MIME嗅探库/实用程序。
- [mimesniffer](#) - Go的MIME类型嗅探器。
- [mimetype](#) - 用于基于幻数的MIME类型检测的软件包。
- [minify](#) - HTML, CSS, JS, XML, JSON和SVG文件格式的快速缩小器。
- [minquery](#) - 支持高效分页的MongoDB / mgo.v2查询（光标继续列出我们停下来的文档）。
- [mmake](#) - 现代品牌。
- [moldova](#) - 用于基于输入模板生成随机数据的实用程序。
- [mole](#) - cli应用程序，可轻松创建ssh隧道。
- [mssqlx](#) - 数据库客户端库，任何主从服务器，主服务器结构的代理。轻巧且自动平衡。
- [multitick](#) - 用于对齐的行情的多路复用器。
- [myhttp](#) - 使用超时支持发出HTTP GET请求的简单API。
- [okrun](#) - 运行错误压路机。

- [olaf](#) - Go中实现的Twitter Snowflake。
- [onecache](#) - 缓存库，支持多个后端存储（Redis, Memcached, 文件系统等）。
- [panicparse](#) - 将相似的goroutine分组并为堆栈转储着色。
- [peco](#) - 简化的交互式过滤工具。
- [pgo](#) - PHP社区的便捷功能。
- [pm](#) - 过程（即够程）与HTTP API管理器。
- [rclient](#) - REST API的可读，灵活，易于使用的客户端。
- [realize](#) - 使用文件监视程序构建系统并实时重新加载。使用自定义路径运行，构建和监视文件更改。
- [repeat](#) - 实施不同的退避策略，对重试操作和心跳很有用。
- [request](#) - Go HTTP Requests for Humans™。
- [rerate](#) - Go的基于Redis的速率计数器和速率限制器。
- [rerun](#) - 当源更改时，重新编译并重新运行go应用。
- [rest-go](#) - 一个软件包，提供了许多使用rest api的有用方法。
- [retry](#) - 最先进的功能机制，可以重复执行操作直到成功。
- [retry](#) - 用于Go的简单但高度可配置的重试包。
- [retry](#) - 用于Go的简单重试机制包。
- [retry](#) - 一个非常简单的库，以确保完成您的工作。
- [retry-go](#) - golang的重试变得简单容易。
- [robustly](#) - 灵活运行功能，捕获并重新启动紧急情况。
- [scan](#) - 扫描golang sql.Rows直接结构，条带或基本类型。
- [serve](#) - 您需要的任何位置的静态http服务器。
- [shutdown](#) - 应用关机钩子进行os.Signal处理。
- [silk](#) - 读取丝绸netflow文件。
- [slice](#) - 常见的Go slice操作的类型安全函数。
- [sliceconv](#) - 基本类型之间的切片转换。
- [slicer](#) - 使切片操作更加轻松。
- [spinner](#) - Go程序包，可轻松为终端Spinner提供选项。
- [sqlx](#) - 在出色的内置数据库/ sql软件包的基础上提供了一组扩展。
- [sslice](#) - 创建一个始终排序的切片。
- [Storm](#) - 简单和BoltDB强大的工具包。
- [structs](#) - 实现简单的功能来操纵结构。
- [Task](#) - 简单的“制作”替代方案。
- [toolbox](#) - 切片，映射，多图，结构，函数，数据转换实用程序。服务路由器，宏评估程序，令牌生成器。
- [ugo](#) - ugo是切片工具箱，具有Go的简洁语法。
- [UNIS](#) - Go中字符串实用程序的Common Architecture™。
- [usql](#) - usql是SQL数据库的通用命令行界面。
- [util](#) - 有用的实用程序功能的集合。（字符串，并发，操作等）。
- [wuzz](#) - 用于HTTP检查的交互式cli工具。
- [xferspd](#) - Xferspd在golang中提供了二进制差异和补丁库。

UUID

UUID

用于处理UUID的库。

- [goid](#) - 生成并解析符合RFC4122的V4 UUID。
- [nanoid](#) - 微型高效的Go唯一字符串ID生成器。
- [sno](#) - 具有嵌入式元数据的紧凑，可排序和快速的唯一ID。
- [ulid](#) - ULID（通用唯一词典分类排序标识符）的实现。
- [uniq](#) - 使用命令无需麻烦，安全，快速的唯一标识符。
- [uuid](#) - 使用快速或加密质量的随机节点标识符生成，编码和解码UUID v1。
- [uuid](#) - 通用唯一标识符（UUID）的实现。支持创建和解析UUID。积极维护satori uuid的叉子。
- [wuid](#) - 一种非常快速的唯一数字生成器，比UUID快10-135倍。

验证方式

验证方式

用于验证的库。

- [checkdigit](#) - 提供校验位算法（卢恩，维尔赫夫，达姆）和计算器（ISBN，EAN，JAN，UPC，等等）。
- [govalidator](#) - 用于字符串，数字，切片和结构的验证器和消毒剂。
- [govalidator](#) - 使用简单规则验证Golang请求数据。受到Laravel请求验证的极大启发。
- [jio -joi](#)类似于JSON模式验证淳佳。
- [ozzo-validation](#) - 支持使用普通代码结构中指定的可配置和可扩展验证规则（而不是struct标签）验证各种数据类型（结构，字符串，映射，切片等）。
- [terraform-validator](#) - Terraform的规范和约定验证器。
- [validate](#) - 用于数据验证和过滤的Go软件包。支持验证Map，Struct，Request（Form，JSON，url.Values，Uploaded Files）数据和更多功能。
- [validate](#) - 该软件包提供了一个框架，用于编写Go应用程序的验证。
- [validator](#) - 进行结构和字段验证，包括交叉场，交叉结构，地图，切片和阵列潜水。

版本控制

版本控制

用于版本控制的库。

- [gh](#) - 脚本化服务器和网络/ HTTP中间件GitHub的网络挂接。
- [git2go](#) - 转到libgit2的绑定。
- [go-git](#) - 在纯Go中高度可扩展的Git实现。
- [go-vcs](#) - 在Go中操纵和检查VCS存储库。
- [hercules](#) - 从Git存储库历史中获得高级见解。
- [hgo](#) - Hgo是Go软件包的集合，提供对本地Mercurial存储库的读取访问。

视频

视频

用于处理视频的库。

- [gmf](#) - `FFmpeg av *` 库的绑定。
- [go-astisub](#) - 在GO中处理字幕（.srt, .stl, .ttml, .webvtt, .ssa / .ass, 图文电视, .smi等）。
- [go-astits](#) - 在GO中本地解析和解复用MPEG传输流（.ts）。
- [go-m3u8](#) - Apple m3u8播放列表的解析器和生成器库。
- [goav](#) - FFmpeg的综合Go绑定。
- [gst](#) - GStreamer的绑定。
- [libgosubs](#) - go的字幕格式支持。支持.srt, .ttml和.ass。
- [libvlc-go](#) - libvlc 2.X / 3.X / 4.X的绑定（由VLC媒体播放器使用）。
- [m3u8](#) - Apple HLS的M3U8播放列表的解析器和生成器库。
- [v4l](#) - 用Go编写的Linux视频捕获库。

Web框架

Web框架

全栈Web框架。

- [aah](#) - Go的可扩展，高性能，快速开发Web框架。
- [Aero](#) - 用于Go的高性能Web框架，在Lighthouse中得分最高。
- [Air](#) - Go的理想完善的Web框架。
- [Banjo](#) - 非常简单，快速的Go Web框架。
- [Beego](#) - beego是一种用于Go编程语言的开源高性能Web框架。
- [Buffalo](#) - 带动Rails的生产力！
- [Echo](#) - 高性能，简约的Go Web框架。
- [Fireball](#) - 更具“自然”感觉的Web框架。
- [Flamingo](#) - 可插拔Web项目的框架。包括模块概念并提供DI, Configareas, i18n, 模板引擎, graphql, 可观察性, 安全性, 事件, 路由和反向路由等功能。
- [Flamingo Commerce](#) - 使用干净的体系结构（如DDD以及端口和适配器）提供电子商务功能，可用于构建灵活的电子商务应用程序。
- [Gin](#) - Gin是用Go! 编写的Web框架。它具有类似于martini的API, 性能更高, 速度最高可提高40倍。如果您需要性能和良好的生产率。
- [Ginrpc](#) - Gin参数自动绑定工具, gin rpc工具。
- [Gizmo](#) - 微服务工具包使用的纽约时报。
- [go-json-rest](#) - 设置RESTful JSON API的快速简便方法。
- [go-rest](#) - Go的小巧的REST框架。
- [Goa](#) - Goa提供了用于在Go中开发远程API和微服务的整体方法。
- [goa](#) - goa就像golang的koajs一样, 它是一个基于中间件的灵活, 轻便, 高性能和可扩展的Web框架。
- [Golax](#) - 一个非Sinatra快速HTTP框架, 支持Google自定义方法, 深度拦截器, 递归等。
- [Golf](#) - 高尔夫是一种用于Go的快速, 简单, 轻便的微型Web框架。它具有强大的功能, 除了Go标准库外没有其他依赖项。
- [Gondola](#) - 用于更快地编写站点的Web框架。
- [gongular](#) - 具有输入映射/验证和（DI）依赖注入的Fast Go Web框架。
- [hiboot](#) - hiboot是具有自动配置和依赖项注入支持的高性能Web应用程序框架。
- [Macaron](#) - Macaron是Go中的一种高效且模块化的Web设计框架。
- [mango](#) - 芒果是Go的模块化Web应用程序框架, 其灵感来自Rack和PEP333。
- [Microservice](#) - 为创建微服务的, 写在Golang框架。
- [neo](#) - Neo是最小且快速的Go Web Framework, 具有极其简单的API。
- [patron](#) - Patron是遵循最佳云实践的微服务框架, 其重点是生产力。
- [Resourcea](#) - 用于快速编写基于资源的服务的REST框架。
- [REST Layer](#) - 用于在数据库之上构建REST / GraphQL API的框架, 主要通过代码进行配置。
- [Revel](#) - Go语言的高效Web框架。
- [rex](#) - Rex是一个基于大猩猩/ mux进行模块化开发的库, 与.x完全兼容net/http。
- [rux](#) - 用于构建golang HTTP应用程序的简单快速的Web框架。
- [tango](#) - 用于Go的微型和可插入Web框架。

Web框架

- [tigertonic](#) - 受Dropwizard启发而构建JSON Web服务的Go框架。
- [uAdmin](#) - 受Django启发的功能齐全的Golang Web框架。
- [utron](#) - 用于Go (Golang) 的轻量级MVC框架。
- [vox](#) - 一个golang人类web框架，由兴亚启发严重。
- [WebGo](#) - 用于构建Web应用程序的微框架；与处理程序链接，中间件和上下文注入。使用符合标准库的HTTP处理程序（即http.HandlerFunc）。
- [YARF](#) - 快速的微框架，旨在以快速，简单的方式构建REST API和Web服务。

中间件

中间件

实际的中间件

- [client-timing](#) - Server-Timing标头的HTTP客户端。
- [CORS](#) - 轻松向您的API添加CORS功能。
- [formjson](#) - 透明地将JSON输入作为标准表单POST处理。
- [go-server-timing](#) - 添加/解析服务器计时标头。
- [Limiter](#) - Go的失效简单速率限制中间件。
- [In-paywall](#) - Go中间件，用于通过Lightning Network（比特币）按请求在API上获利。
- [Tollbooth](#) - 速率限制HTTP请求处理程序。
- [XFF](#) - 处理X-Forwarded-For标头和朋友。

用于创建HTTP中间件的库

- [alice](#) - Go的无痛中间件链接。
- [catena](#) - http.Handler包装器类别（与“链”相同的API）。
- [chain](#) - 与作用域数据（基于net / context的“中间件”）的处理程序包装器链。
- [go-wrap](#) - net / http的小型中间件软件包。
- [gores](#) - 处理HTML，JSON，XML等响应的Go包。对于RESTful API很有用。
- [interpose](#) - 用于golang的极简网络/ http中间件。
- [muxchain](#) - 用于net / http的轻量级中间件。
- [negroni](#) - Golang的惯用HTTP中间件。
- [render](#) - Go包，用于轻松呈现JSON，XML和HTML模板响应。
- [renderer](#) - Go的简单，轻量级和更快的响应（JSON，JSONP，XML，YAML，HTML，文件）渲染包。
- [rye](#) - 微型围棋中间件库（罐头中间件），支持智威汤逊，CORS，Statsd和Go 1.7上下文。
- [stats](#) - Go中间件，用于存储有关Web应用程序的各种信息。

路由器

路由器

- [alien](#) - 来自外层空间的轻巧快速的http路由器。
- [bellt](#) - 一个简单的Go HTTP路由器。
- [Bone](#) - 闪电快速HTTP多路复用器。
- [Bxog](#) - Go的简单快速HTTP路由器。它适用于难度，长度和嵌套不同的路线。而且他知道如何根据接收到的参数创建URL。
- [chi](#) - 基于net / context构建的小型，快速且富有表现力的HTTP路由器。
- [fasthttprouter](#) - 高性能的路由器httprouter。第一个适合的路由器fasthttp。
- [FastRouter](#) - 用Go编写的快速，灵活的HTTP路由器。
- [gocraft/web](#) - Go中的Mux和中间件软件包。
- [Goji](#) - Goji是一种简约且灵活的HTTP请求多路复用器，支持net/context。
- [goroute](#) - 简单但功能强大的HTTP请求多路复用器。
- [GoRouter](#) - GoRouter是服务器/ API微型框架，HTTP请求路由器，多路复用器，多路复用器，可为请求路由器提供中间件支持net/context。
- [gowww/router](#) - 与net / http.Handler接口完全兼容的快速闪电HTTP路由器。
- [httprouter](#) - 高性能路由器。使用它和标准的http处理程序可以形成一个非常高性能的Web框架。
- [httpreemux](#) - 用于Go的高速，基于树的灵活HTTP路由器。来自httprouter的启发。
- [lars](#) - Go的轻量级，快速且可扩展的零分配HTTP路由器，用于创建可自定义的框架。
- [mux](#) - 强大的URL路由和golang调度。
- [ozzo-routing](#) - 一种非常快速的Go（golang）HTTP路由器，支持正则表达式路由匹配。附带对构建RESTful API的全面支持。
- [pure](#) - 是一种轻量级HTTP路由器，坚持使用std“ net / http”实现。
- [Siesta](#) - 可组合的框架来写的中间件和处理程序。
- [vestigo](#) - 用于go Web应用程序的高性能，独立的，HTTP兼容的URL路由器。
- [violetear](#) - 转到HTTP路由器。
- [xmux](#) - 基于高性能复用器httprouter与net/context支持。
- [xujiajun/gorouter](#) - Go的简单快速HTTP路由器。

视窗

视窗

- [d3d9](#) - Direct3D9的绑定。
- [go-ole](#) - 用于golang的Win32 OLE实现。
- [gosddl](#) - 从SDDL字符串转换为用户友好的JSON。SDDL由四部分组成：所有者，主要组，DACL，SACL。

XML格式

XML格式

用于处理XML的库和工具。

- [XML-Comp](#) - 简单的命令行XML比较器，生成文件夹，文件和标签的差异。
- [xml2map](#) - 用Golang编写的XML到MAP转换器。
- [xmlwriter](#) - 基于libxml2的xmlwriter模块的过程XML生成API。
- [xpath](#) - Go的XPath软件包。
- [xquery](#) - XQuery使您可以使用XPath表达式从HTML / XML文档中提取数据。
- [zek](#) - 从XML生成Go结构。

代码分析

代码分析

- [apicompat](#) - 检查Go项目的最新更改是否存在向后不兼容的更改。
- [dupl](#) - 代码克隆检测工具。
- [errcheck](#) - Errcheck是用于检查Go程序中未经检查的错误的程序。
- [gcvis](#) - 实时可视化Go程序GC跟踪数据。
- [go-checkstyle](#) - checkstyle是样式检查工具，例如java checkstyle。该工具的灵感来自java checkstyle golint。该样式涉及“Go Code评论注释”中的某些要点。
- [go-cleanarch](#) - 创建go-cleanarch是为了验证Clean Architecture规则，例如The Dependency Rule和Go项目中程序包之间的交互。
- [go-critic](#) - 源代码linter，它带来了当前在其他linter中未实现的检查。
- [go-mod-outdated](#) - 查找Go项目的过时依赖项的简便方法。
- [go-outdated](#) - 显示过期软件包的控制台应用程序。
- [goast-viewer](#) - 基于Web的Golang AST可视化工具。
- [GoCover.io](#) - GoCover.io提供任何golang软件包即服务的代码覆盖率。
- [goimports](#) - 自动修复（添加，删除）Go导入的工具。
- [GolangCI](#) - GolangCI是针对GitHub拉取请求的自动化Golang代码审查服务。服务是开源的，对于开源项目是免费的。
- [GoLint](#) - Golint是Go源代码的皮特。
- [Golint online](#) - 使用golint软件包在线上GitHub，Bitbucket和Google Project Hosting上在线Go源文件。
- [GoPlantUML](#) - 生成文本plantump类图的库和CLI，该类图包含有关结构和接口以及它们之间的关系的信息。
- [goreturns](#) - 添加零值返回语句以匹配func返回类型。
- [gosimple](#) - gosimple是Go源代码的linter，专门研究简化代码。
- [gostatus](#) - 命令行工具，显示包含Go软件包的存储库的状态。
- [lint](#) - 作为go测试的一部分运行棉绒。
- [php-parser](#) - 用Go编写的PHP解析器。
- [staticcheck](#) - staticcheck go vet用于类固醇，对ReSharper for C# 等工具应用了大量的静态分析检查。
- [tarp](#) - tarp在Go源代码中查找没有直接单元测试的函数和方法。
- [tickgit](#) - CLI和go软件包，用于显示代码注释TODO（以任何语言显示）并应用git blame标识作者。
- [unconvert](#) - 从Go源代码中删除不必要的类型转换。
- [unused](#) - 未使用的检查将代码用于未使用的常量，变量，函数和类型。
- [validate](#) - 自动验证带有标签的结构域。

编辑器插件

编辑器插件

- [Go plugin for JetBrains IDEs](#) - 用于JetBrains IDE的 Go插件- 用于JetBrains IDE的 Go插件。
- [go-language-server](#) - server-将VSCode go扩展名转换为支持language-server-protocol的语言服务器的包装器。
- [go-mode](#) - GNU / Emacs的Go模式。
- [go-plus](#) - plus-用于Atom的Go（Golang）软件包，可添加自动完成，格式设置，语法检查，整理和审核。
- [gocode](#) - Go编程语言的自动完成守护程序。
- [goprofiling](#) - 此扩展在VS Code中添加了对Go语言的基准分析支持。
- [GoSublime](#) - 文本编辑器SublimeText 3的Golang插件集合，提供代码完成和其他类似IDE的功能。
- [gounit-vim](#) - Vim插件，用于基于函数或方法的签名生成Go测试。
- [theia-go-extension](#) - Theia IDE的Go语言支持。
- [vim-compiler-go](#) - Vim插件，突出显示保存时的语法错误。
- [vim-go](#) - go-用于Vim的Go开发插件。
- [vscode-go](#) - Visual Studio Code（VS Code）扩展，提供对Go语言的支持。
- [Watch](#) - 在文件更改时以极致优势运行命令。

硬件

Go + 硬件

此存储库是工具，软件包和教程的目录，可让您在硬件项目中介绍Go。

为什么选go?

- Go可以针对主要在非实时嵌入式操作系统范围内的平台和体系结构。
 - 对Linux的ARMv5, ARMv6, ARMv7和ARMv8支持
 - ARMv7, ARMv8, x86对Android的支持
 - 实验性OpenWRT
- 开箱即用的交叉编译故事。
 - 例如，运行`GOOS=linux GOARCH=arm go build`以从Mac或Windows机器为ARM / linux板构建二进制文件。不需要其他配置。
- Go中内置的并发原语使得编写并发程序变得更加容易。
 - 在实施硬件控制器方面的强大优势。
 - 在实施网络设备方面的强大优势。
- Go是垃圾收集语言，但是自Go 1.4起，垃圾收集器的占用空间已得到显著改善。即使对于大堆，暂停时间也被定为10ms或更短，并且在非实时操作系统上并不是一个明显的缺点-OS调度程序的抢占性质比暂停时间更是一个主要问题。
- Go的网络堆栈质量高且维护良好。网络是物联网的核心组件。
- Go提供了现成的HTTP, HTTPS和HTTP / 2客户端/服务器实现。
 - 标准库可能缺少P2P协议的更高级别的实现，但是社区正在努力弥补这一空白。
 - 社区还投入时间来实施更新的以物联网为目标的网络协议，例如COAP。
- 与其他高级编程语言（如Python和Java）不同，使用Go语言在Go中编写C绑定非常简单。从Go上下文中依赖现有的C / C++库要容易得多。
 - 例如，[portmidi](#)绑定。请参阅源代码，以了解如何在同一个Go文件中混合和匹配Go和C代码。
- 类似于C的语法使现有的IoT /嵌入式编程人员（大多熟练使用C语言）可以在无需过多了解该语言的情况下读写Go。
- Go程序可编译为静态二进制文件，并且不需要主机（例如VM）上的运行时。部署是复制/粘贴二进制文件。
- Go拥有强大的社区人物，从早期开始就从事硬件方面的工作。
 - Gobot在第一届GopherCon上首次亮相。

- 社区已经提供了许多围绕硬件和机器人技术的库和框架。可以在[go硬件](#)上找到列表。
- Go高效，快速且内存占用少。
- 服务器和客户端（连接的设备或移动设备）之间的代码重用。
 - 例如，gRPC Go在服务器，Android，iOS和嵌入式Linux上开箱即用。

讲解

- (版本1.5及更高版本) [@rakyll/go-1-5-cross-compilation-488092ba44ec"target="_blank">交叉编译指南](#)
- (执行1.4及更低版本) [交叉编译指南](#)
- [查阅ARM技巧, 已知问题和成功案例](#)

配套

控制器

- [devices](#) - Go的高级设备/传感器软件包的存储库。
- [gobot](#) - 具有各种平台API的高级Go软件包，可与流行的驱动程序进行通信，例如Arduino和Raspberry Pi上的GPIO，Leap Motion或Pebble。
- [go-gpio](#) - Go的GPIO，不需要cgo。
- [embed](#) - Go的嵌入式编程框架。
- [fadecandy](#) - 基于OPC的抖动LED控制器
- [godrone](#) - 用Go编写的高级Parrot AR Drone 2.0框架。
- [launchpad](#) - Novation Launchpad的高级控制器库。
- [littlebits](#) - 用于Go的littleBits控制器。它需要USB I/O 模块。
- [piCamera](#) - 在GoLang中为Raspberry Pi相机捕获图像流
- [hwio](#) - 松散地基于Arduino的ARM板的硬件库，例如Raspberry Pi和BeagleBone。
- [go-lepton](#) - 将通过连接到Raspberry Pi SPI端口的FLIR Lepton上拍摄的图像通过WebSockets通过嵌入式HTTP服务器进行流式传输。
- [go-embedded](#) - 对I2C, SPI, PWM, GPIO, ADC的嵌入式Linux支持
- [go-beaglebone](#) - BeagleBone开源硬件的Go包
- [go-mavlink](#) - Go的MAVLink协议实现（MAVLink用于控制无人机）
- [joystick](#) - 一种轮询的API，用于读取连接的操纵杆的状态。
- [emgo](#) - 一种用于对嵌入式系统（例如STM32 MCU）进行编程的类语言。
- [go-rpi-rgb-led-matrix](#) - matrix-使用Raspberry Pi GPIO控制多达三个链的32x32或16x32 RGB LED显示
- [go-rpi-ws281x](#) - 用于WS281X LED的Raspberry Pi PWM库的绑定
- [ghw](#) - Golang硬件发现/检查库
- [sysinfo](#) - 一个纯的Go库，提供Linux OS /内核/硬件系统信息。

信号处理，计算机视觉，图形，媒体

- [go-gl](#) - Go的OpenGL绑定。
- [go-opc](#) - Go的开放像素控件绑定。

- [go-opencv](#) - Go的OpenCV绑定。
- [go-sox](#) - Go的SoX绑定。
- [portaudio](#) - Go的PortAudio绑定。
- [portmidi](#) - Go的PortMidi绑定。
- [go-osc](#) - 为Go打开声音控制（OSC）绑定。
- [opengl](#) - 包装OpenGL API的2D图形库
- [lirc](#) - Linux红外遥控器的Go Client。允许发送和接收IR命令。
- [gocv](#) - Go的OpenCV 3+绑定。还支持OpenVINO。

网络，外围通讯

- [mdns](#) - Go中的mDNS服务器和客户端实现。组播DNS可用于在本地网络上发现服务和消息，而无需使用权威DNS服务器。
- [gatt](#) - 提供蓝牙低功耗GATT实施。
- [go.hid](#) - 提供与USB人机接口设备的通信。
- [goble](#) - Go的蓝牙低功耗（HM10）模块
- [serial](#) - Go的串行端口通讯程序包
- [firmata](#) - Go的Firmata客户
- [periph](#) - 外设I / O在Go
- [go-ckb](#) - Go中的Corsair ckb-daemon通信
- [ble](#) - 适用于Linux / macOS的低功耗蓝牙通信包装器。

移动版（1.5或更高版本）

- 本机应用程序（Android和iOS）
- SDK应用程序

go生成工具

go生成工具

- [generic](#) - Go的灵活数据类型。
- [genny](#) - Go的优雅仿制药。
- [gocontracts](#) - 通过使代码与文档同步来将按合同设计引入Go。
- [gonerics](#) - Go中的惯用泛型。
- [gotests](#) - 从您的源代码生成Go测试。
- [gounit](#) - 使用您自己的模板生成Go测试。
- [hasgo](#) - 为您的切片生成Haskell启发的函数。
- [re2dfa](#) - 将正则表达式转换为有限状态机并输出Go源代码。
- [TOML-to-Go](#) - 在浏览器中立即将TOML转换为Go类型。

go工具

go工具

- [colorgo](#) - go用于彩色go build输出的命令周围的包装器。
- [depth](#) - 任何包通过分析进口的依赖可视化树。
- [gb](#) - Go编程语言的易于使用的基于项目的构建工具。
- [generator-go-lang](#) - 一个Yeoman生成器，用于启动新的Go项目。
- [gilbert](#) - Go项目的构建系统和任务运行器。
- [go-callvis](#) - 使用点格式可视化Go程序的调用图。
- [go-james](#) - Go项目框架创建者，无需手动设置即可构建和测试项目。
- [go-pkg-complete](#) - go和wgo的Bash完成。
- [go-swagger](#) - Go的 Swagger 2.0实现。Swagger是RESTful API的简单但功能强大的表示形式。
- [godbg](#) - Rustsdbg!宏的实现，可在开发过程中快速轻松地进行调试。
- [gothanks](#) - GoThanks自动为您的go.mod github依赖项加注星标，以这种方式向其维护者发送一些爱意。
- [OctoLinker](#) - 使用GitHub的OctoLinker浏览器扩展可以高效地浏览go文件。
- [richgo](#) - go test使用文本装饰丰富输出。
- [rts](#) - RTS: 对struct的响应。从服务器响应生成Go结构。

DevOps工具

DevOps工具

- [aptly](#) - 适当地是Debian仓库管理工具。
- [aurora](#) - 基于Web的跨平台Beanstalkd队列服务器控制台。
- [awsenv](#) - 小型二进制文件，用于为配置文件加载Amazon（AWS）环境变量。
- [Blast](#) - 用于API负载测试和批处理作业的简单工具。
- [bombardier](#) - 快速的跨平台HTTP基准测试工具。
- [bosun](#) - 时间序列警报框架。
- [DepCharge](#) - 帮助协调跨大型项目中许多依赖项的命令执行。
- [Dockerfile-Generator](#) - 一个go库和一个可执行文件，可使用各种输入通道生成有效的Dockerfile。
- [dogo](#) - 监视源文件中的更改，并自动编译和运行（重新启动）。
- [drone-jenkins](#) - 使用二进制，泊坞窗或Drone CI触发下游Jenkins作业。
- [drone-scp](#) - 使用二进制文件，docker或Drone CI通过SSH复制文件和工件。
- [Dropship](#) - 通过CDN部署代码的工具。
- [easyssh-proxy](#) - Golang软件包，可通过SSH和SCP下载通过轻松地远程执行ProxyCommand。
- [fac](#) - 修复git合并冲突的命令行用户界面。
- [gaia](#) - 使用任何编程语言构建功能强大的管道。
- [Gitea](#) - Gogs的叉子，完全由社区驱动。
- [gitea-github-migrator](#) - 将所有GitHub存储库，问题，里程碑和标签迁移到Gitea实例。
- [go-furnace](#) - 用Go编写的托管解决方案。在AWS，GCP或DigitalOcean上轻松部署应用程序。
- [go-selfupdate](#) - 使Go应用程序能够自我更新。
- [gobrew](#) - 使用 gobrew，您可以轻松地在多个版本的go之间切换。
- [godbg](#) - 基于Web的gdb前端应用程序。
- [Gogs](#) - Go编程语言中的自托管Git服务。
- [gonative](#) - 创建Go的构建的工具，该工具可以交叉编译到所有平台，同时仍使用支持Cgo的stdlib软件包版本。
- [govvv](#) - “构建”包装器可轻松将版本信息添加到Go二进制文件中。
- [gox](#) - 简单易用，没有多余的装饰Go交叉编译工具。
- [goxc](#) - Go的构建工具，着重于交叉编译和打包。
- [grapes](#) - 轻量级工具，旨在通过ssh轻松分发命令。
- [GVM](#) - GVM提供了管理Go版本的界面。
- [Hey](#) - Hey 这是个很小的程序，它将一些负载发送到Web应用程序。
- [jcli](#) - Jenkins CLI使您可以轻松管理Jenkins。
- [kala](#) - 简单，现代和高性能的工作计划程序。
- [kcli](#) - 用于检查kafka主题/分区/消息的命令行工具。
- [kubernetes](#) - Google的Container Cluster Manager。
- [Istags](#) - 用于在不同注册表之间同步Docker映像的工具和API。
- [lwc](#) - UNIX wc命令的实时更新版本。
- [manssh](#) - manssh是用于轻松管理ssh别名配置的命令行工具。
- [Moby](#) - 容器生态系统的协作项目，用于组装基于容器的系统。
- [Mora](#) - 用于访问MongoDB文档和元数据的REST服务器。

- [ostent](#) - 收集并显示系统指标，并可选地中继到Graphite和/或InfluxDB。
- [Packer](#) - Packer是一个工具，可从一个源配置为多个平台创建相同的机器映像。
- [Pewpew](#) - 灵活的HTTP命令行压力测试仪。
- [Pomerium](#) - Pomerium是一个身份识别访问代理。
- [Rodent](#) - Rodent 可帮助您管理Go版本，项目和跟踪依赖关系。
- [s3gof3r](#) - 经过优化的小型实用程序/库，用于将大型对象高速传输到Amazon S3和从Amazon S3传输出来。
- [Scaleway-cli](#) - cli-从命令行管理BareMetal服务器（与Docker一样容易）。
- [script](#) - 轻松在Go中为DevOps和系统管理任务编写类似shell的脚本。
- [sg](#) - 对一组HTTP端点（例如ab）进行基准测试，并可能根据每次响应之间的先前响应，在每次调用之间使用响应代码和数据来处理特定的服务器压力。
- [skm](#) - SKM是一个简单而强大的SSH密钥管理器，它可以帮助您轻松管理多个SSH密钥！
- [StatusOK](#) - 监视您的网站和REST API。当服务器关闭或响应时间超出预期时，通过Slack，电子邮件通知。
- [terraform-provider-openapi](#) - Terraform提供程序插件，可在运行时根据包含公开API定义的OpenAPI文档（以前称为swagger文件）动态配置自身。
- [traefik](#) - 反向代理和负载均衡器，支持多个后端。
- [uTask](#) - 自动化引擎，用于建模和执行yami中声明的业务流程。
- [Vegeta](#) - HTTP负载测试工具和库。超过9000！
- [webhook](#) - 工具，允许用户创建在服务器上执行命令的HTTP端点（挂钩）。
- [Wide](#) - 使用Golang的团队基于Web的IDE。
- [winrm-cli](#) - 在Windows计算机上远程执行命令的Cli工具。