李兆海

GO开发环境及工具

个人介绍

- 语言历程：C->C++->Python->Go
- 与Go结缘：
  - Go语言：互联网时代的C @ 2010 Beta技术沙龙
  - Golang初探 @ 2011《程序员》杂志
  - Go，下一代C @ 2015 天津GDG
  - Go in Action中文版翻译（即将出版）

演讲内容

- Go语言简介
- 开发环境目录结构
- 代码结构组织
- 编辑器
- 性能优化

演讲内容

- Go语言简介
- 开发环境目录结构
- 代码结构组织
- 编辑器
- 性能优化

GO语言简介——设计特点

- 源自Google
- 用于开发高并发，高性能的网络系统
- 简洁的语法
- 结合了工程上的最佳实践

GO语言简介——语言特点

- 静态语言
- GC
- 支持并发
- 非传统多态

演讲内容

- Go语言简介
- 开发环境目录结构
- 代码结构组织
- 编辑器
- 性能优化

开发环境目录结构

- 环境变量
- 目录结构
  - 公开库/内部共享库
  - 可执行项目
- 第三方依赖

环境变量

- GOROOT：默认为安装文件所在目录
- GOPATH：第三方包查找路径
- 包查找顺序：
  - $GOROOT/src/ ——一般是标准库
  - 按照顺序查找$GOPATH

目录结构 / 公开库，内部共享库

- 项目保存在$GOPATH/src之下
- 项目根目录直接保存代码
- 子目录保存拆分出来的子库
- 子目录保存例子程序
- 引用：go get github.com/xxx/xxx

```
$GOPATH
├── ...
└── src
    ├── github.com
    │   ├── 3rd_libraries
    │   └── your_account
    │       └── your_public_lib
    │           ├── Makefile
    │           ├── LICENSE
    │           ├── README.md
    │           ├── codes.go
    │           ├── codes_test.go
    │           ├── ...
    │           ├── example
    │           │   ├── asset
    │           │   │   └── ...
    │           │   ├── codes.go
    │           │   └── main.go
    │           └── sub_library
    │               ├── codes.go
    │               ├── codes_test.go
    │               └── ...
    └── your_private_lib
        └── ...
```

目录结构 / 公开库，内部共享库 - MAKEFILE

```
> cat Makefile

build:
   go build

install:
   go install

test:
   go test -v ./…

clean:
   go clean
```

```
$GOPATH
├── ...
└── src
    ├── github.com
    │   ├── 3rd_libraries
    │   └── your_account
    │       └── your_public_lib
    │           ├── Makefile
    │           ├── LICENSE
    │           ├── README.md
    │           ├── codes.go
    │           ├── codes_test.go
    │           ├── ...
    │           ├── example
    │           │   ├── asset
    │           │   │   └── ...
    │           │   ├── codes.go
    │           │   └── main.go
    │           └── sub_library
    │               ├── codes.go
    │               ├── codes_test.go
    │               └── ...
    └── your_private_lib
```

目录结构 / 可执行项目

- 项目根目录不直接保存代码
- 需要执行env.sh修改$GOPATH
- src子目录下保存代码
- 如果有多个执行文件，分目录保存在src/apps下

```
$GOPATH
├── env.sh
├── Makefile
├── ...
└── src
    ├── github.com
    │   └── 3rd_libraries
    ├── sub_libraries
    │   ├── README.go
    │   ├── codes.go
    │   ├── codes_test.go
    │   └── ...
    ├── examples
    │   ├── README.go
    │   ├── codes.go
    │   ├── codes_test.go
    │   └── ...
    └── apps
        ├── binary1
        │   ├── README.md
        │   ├── config.go
        │   ├── main.go
        │   └── ...
        ├── binary2
        │   ├── README.md
        │   ├── config.go
        │   ├── main.go
        │   └── ...
        └── ...
```

目录结构 / 可执行项目

```
> cat env.sh
export $GOPATH=`pwd`
```

```
> cat Makefile

build:
   go build your_project/...

install:
   go install your_project/...

test:
   go test -v your_project/...

clean:
   go clean
```

```
$GOPATH
├── env.sh
├── Makefile
├── ...
└── src
    ├── github.com
    │   └── 3rd_libraries
    └── your_project
        ├── sub_libraries
        │   ├── README.go
        │   ├── codes.go
        │   ├── codes_test.go
        │   └── ...
        ├── examples
        │   ├── README.go
        │   ├── codes.go
        │   ├── codes_test.go
        │   └── ...
        └── apps
            ├── binary1
            │   ├── README.md
            │   ├── config.go
            │   ├── main.go
            │   └── ...
            ├── binary2
            │   ├── README.md
            │   ├── config.go
            │   ├── main.go
            │   └── ...
            └── ...
```

演讲内容

- Go语言简介
- 开发环境目录结构
- 代码结构组织
- 编辑器
- 性能优化

代码结构组织

- README.md
  - 文档说明，库的逻辑流程，时序图
- protocol.go
  - 定义所有对外要用到的结构和接口
  - 通过这一个文件就能了解库的使用方法
- protocol_test.go
  - 针对对外接口的测试
  - 通过测试展示库的使用方法，测试最基本用法

```
library
├── README.md
├── protocol.go
├── protocol_test.go
└── ...
```

演讲内容

- Go语言简介
- 开发环境目录结构
- 代码结构组织
- 编辑器
- 性能优化

- 编辑器
- Atom
  - autocomplete-go
  - go-plus
- VS Code
  - Go for VS Code

DEMO

演讲内容

- Go语言简介
- 开发环境目录结构
- 代码结构组织
- 编辑器
- 性能优化

性能优化 / 测试

- 单元测试
  - 表组测试
  - testing/quick
- 压力测试
  - benchmem

性能优化 / 测试

- 执行测试：go test
- 执行压力测试：go test -bench .
- 内存测试：go test -bench . -benchmem
- 生成性能测试报告：
  - go test -c
  - ./strcat.test -test.bench . -test.cpuprofile cpu.prof -test.memprofile mem.prof

DEMO

性能优化 / 数据采集

- expvar
  - 内存分配
  - 自定义数据
  - 全局变量
  - http://localhost:8080/debug/vars
- net/http/pprof
  - 在线性能测试
  - 在线内存分配

DEMO

性能优化 / 展示

- go tool pprof
  - go tool pprof ./strcat.test cpu.prof
  - go tool pprof -seconds 2 http://localhost:8080/debug/pprof/profile
  - go tool pprof -alloc_objects ./strcat.test mem.prof
- github.com/uber/go-torch
  - 复制 https://raw.githubusercontent.com/brendangregg/FlameGraph/master/flamegraph.pl 到go-torch所在目录
  - go-torch ./strcat.test cpu.prof
  - go-torch –alloc_objects ./strcat.test cpu.prof
  - 配合httpprof：go-torch -t 2

DEMO

李兆海

Twitter/GitHub/微信：@googollee

Q&A