Memcache

sessions

App Engine Sessions - Three Options:

- Use Google logins and their session
- Use Gorilla sessions with manual routing
  - (either via http.ServeMux or some other router)
- Use a Session ID and Memcache
  - do a regular cookie
  - store in that cookie a session ID
  - use memcache for their session

memcache

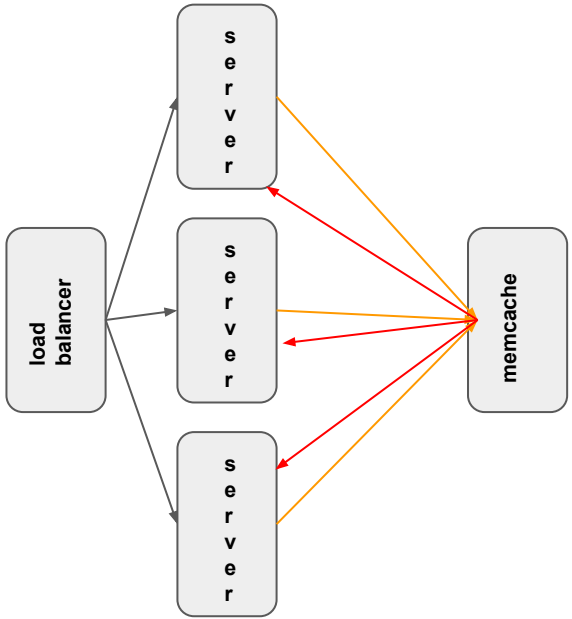
Memcache backstory

- servers weren’t using all of their capacity
  - eg, had 1GB RAM but only 100 MB used
    - memcache built to use it

Memcache

- a big map distributed across all of your machines
  - “a big distributed map”
- memcache is
  - very fast
  - you’re storing things in memory
- you can use memcache to improve data retrieval performance
  - cache database queries in memcache
    - is it in memcache?
      - yes - great, get it from memcache
      - no - do the datastore query, get it, then store it in memcache
    - this decreases load on database which increases performance
- you can use memcache for sessions
  - instead of storing user data in a cookie, store it in memcache

Why not just use a map instead of memcache?


load balancer
server
server
server
memcache

If you need something to absolutely be there, use a real database

memcache docs

Go

Go Runtime Environment

Python | Java | PHP | **Go**

Welcome to Google App Engine for Go. With App Engine, you can build web applications using the Go Programming Language. Your Go application runs on Google's scalable infrastructure and uses large-scale persistent storage and services.

> The Go runtime is also available in a Managed VM hosting environment.

Introduction

App Engine builds and executes Go application code using a safe "sandboxed" environment. Your app receives web requests, performs work, and sends responses by interacting with this environment.

The Go SDK provides an interface similar to the standard Go http package; writing Go App Engine apps is akin to writing stand-alone Go web servers.

The Go runtime environment uses Go version 1.4. The SDK includes the Go compiler and standard library, so it has no additional dependencies. As with the other runtimes, not all the standard library's functionality is available inside the sandbox. For example, attempts to open a socket or write to a file will return an `os.ErrPermission` error.

The SDK includes an automated build service to compile your app, so you'll never need to invoke the compiler yourself. And—as with

- OAuth
- Search
- Sockets
- Task Queues
- URL Fetch
- Users
- XMPP

- Configuration
- Tools

API Reference

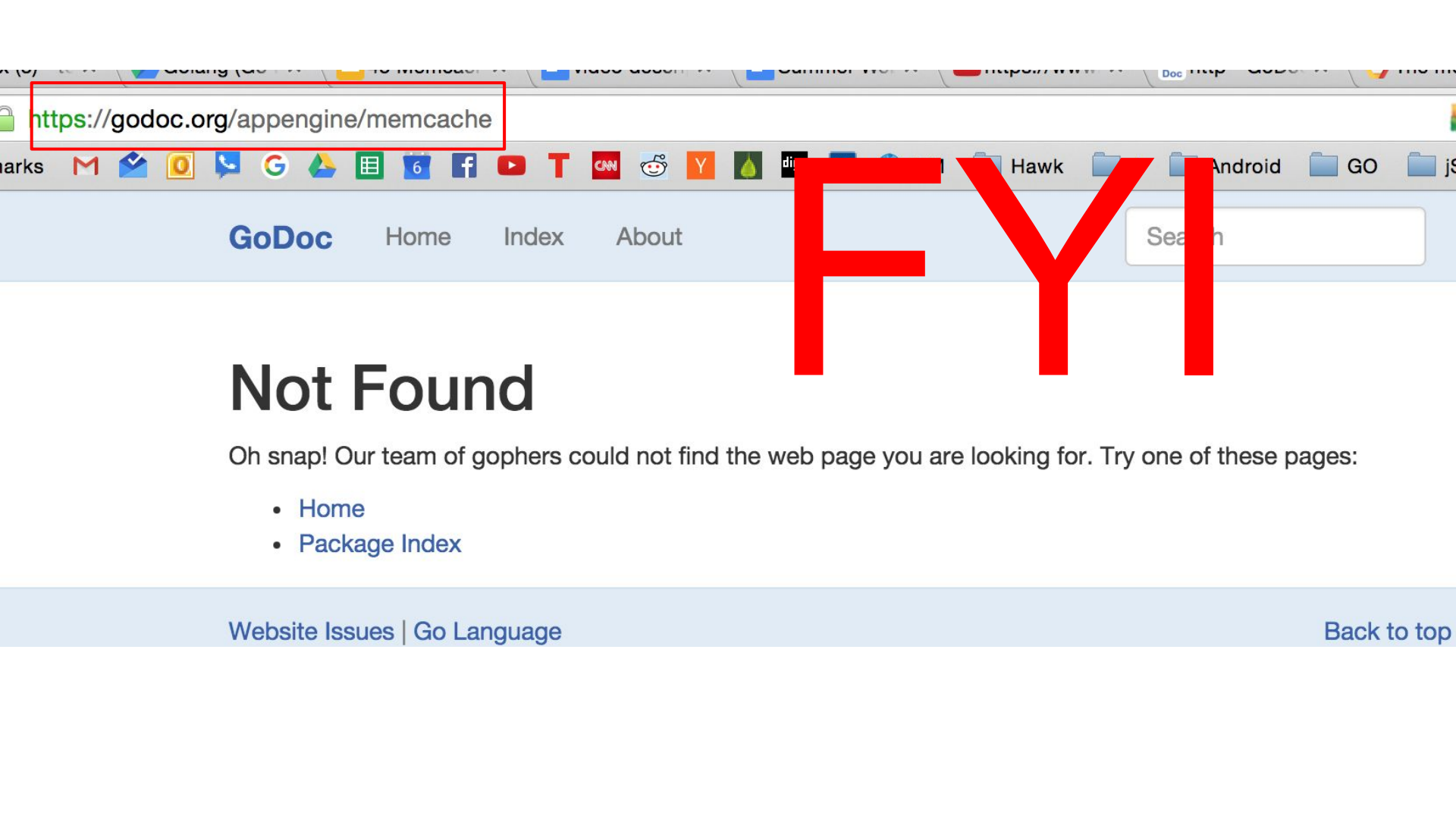
Release Notes

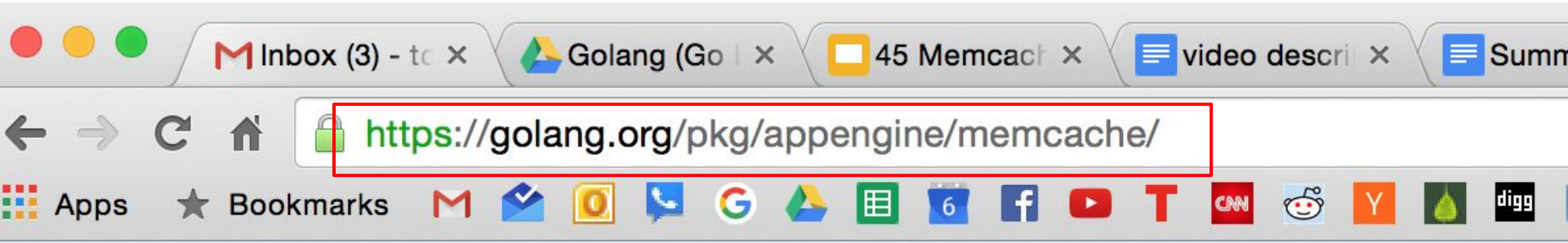
Index

type Item

```
type Item struct {
    // Key is the Item's key (250 bytes maximum).
    Key string
    // Value is the Item's value.
    Value []byte
    // Object is the Item's value for use with a Codec.
    Object interface{}
    // Flags are server-opaque flags whose semantics are entirely up to the
    // App Engine app.
    Flags uint32
    // Expiration is the maximum duration that the item will stay
    // in the cache.
    // The zero value means the Item has no expiration time.
    // Subsecond precision is ignored.
    // This is not set when getting items.
    Expiration time.Duration
    // contains filtered or unexported fields
}
```

Item is the unit of memcache gets and sets.


https://godoc.org/appengine/memcache
Hawk
Android
GO
GoDoc
Home
Index
About
FYI
Not Found
Oh snap! Our team of gophers could not find the web page you are looking for. Try one of these pages:
Home
Package Index
Website Issues | Go Language
Back to top



The Go Programming Language

Docu

FYI

File appengine/memcache

appengine/memcache

file not found: /go/src/appengine/memcache

GoDoc Home Index About Search

appengine: google.golang.org/appengine/memcache Index | Files

package memcache

```
import "google.golang.org/appengine/memcache"
```

Package memcache provides a client for App Engine's distributed in-memo[illegible] sm[illegible]hunks of arbitrary data.

The fundamental operations get and set items, keyed by a string.

```
item0, err := memcache.Get(c, "key")
if err != nil && err != memcache.ErrCacheMiss {
        return err
}
if err == nil {
        fmt.Fprintf(w, "memcache hit: Key=%q Val=[% x]\n", [illegible]m0.Key, item0.Value)
} else {
        fmt.Fprintf(w, "memcache miss\n")
}
```

and

```
item1 := &memcache.Item{
        Key:   "foo",
        Value: []byte("bar"),
}
if err := memcache.Set(c, item1); err != nil {
        return err
}
```

FYI

The Go Programming Language

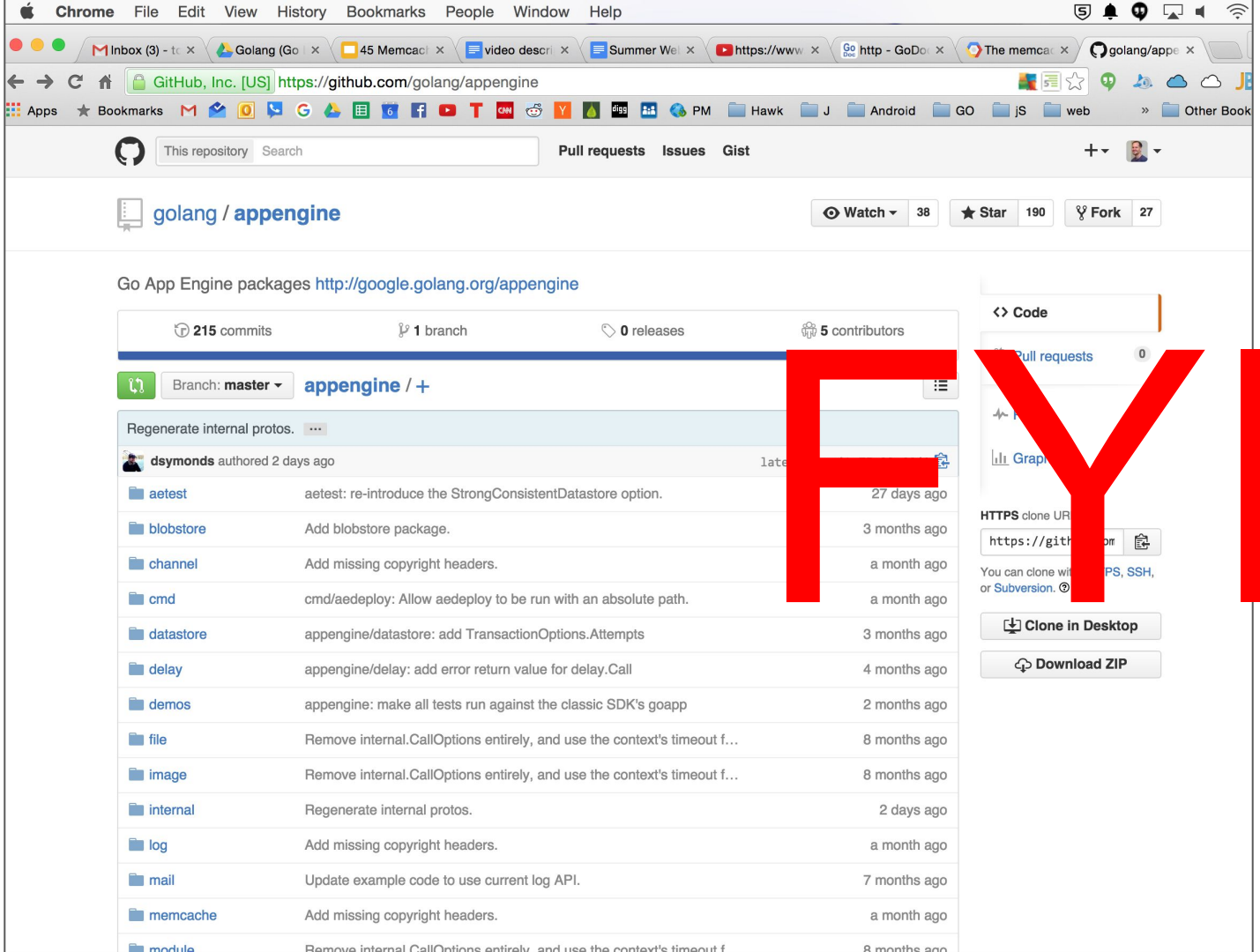
Documents | Packages | The Project | He

FYI

File google.golang.org/appengine/memcache

google.golang.org/appengine/memcache

file not found: /go/src/google.golang.org/appengine/memcache


Chrome File Edit View History Bookmarks People Window Help
https://github.com/golang/appengine
This repository Search
Pull requests Issues Gist
golang / appengine
Watch 38 Star 190 Fork 27
Go App Engine packages http://google.golang.org/appengine
215 commits 1 branch 0 releases 5 contributors
Branch: master appengine / +
Regenerate internal protos.
dsymonds authored 2 days ago
aetest aetest: re-introduce the StrongConsistentDatastore option. 27 days ago
blobstore Add blobstore package. 3 months ago
channel Add missing copyright headers. a month ago
cmd cmd/aedeploy: Allow aedeploy to be run with an absolute path. a month ago
datastore appengine/datastore: add TransactionOptions.Attempts 3 months ago
delay appengine/delay: add error return value for delay.Call 4 months ago
demos appengine: make all tests run against the classic SDK's goapp 2 months ago
file Remove internal.CallOptions entirely, and use the context's timeout f… 8 months ago
image Remove internal.CallOptions entirely, and use the context's timeout f… 8 months ago
internal Regenerate internal protos. 2 days ago
log Add missing copyright headers. a month ago
mail Update example code to use current log API. 7 months ago
memcache Add missing copyright headers. a month ago
Code
Pull requests 0
HTTPS clone URL
Clone in Desktop
Download ZIP
FYI

set, get, memcache.Item

The memcache package

```
import "appengine/memcache"
```

Introduction

Package memcache provides a client for App Engine's distributed in-memory key-value store for small chunks of arbitrary data.

The fundamental operations get and set items, keyed by a string.

```
item0, err := memcache.Get(c, "key")
if err != nil && err != memcache.ErrCacheMiss {
    return err
}
if err == nil {
    fmt.Fprintf(w, "memcache hit: Key=%q Val=[% x]\n", item0.Key, item0.Value)
} else {
    fmt.Fprintf(w, "memcache miss\n")
}
```

and

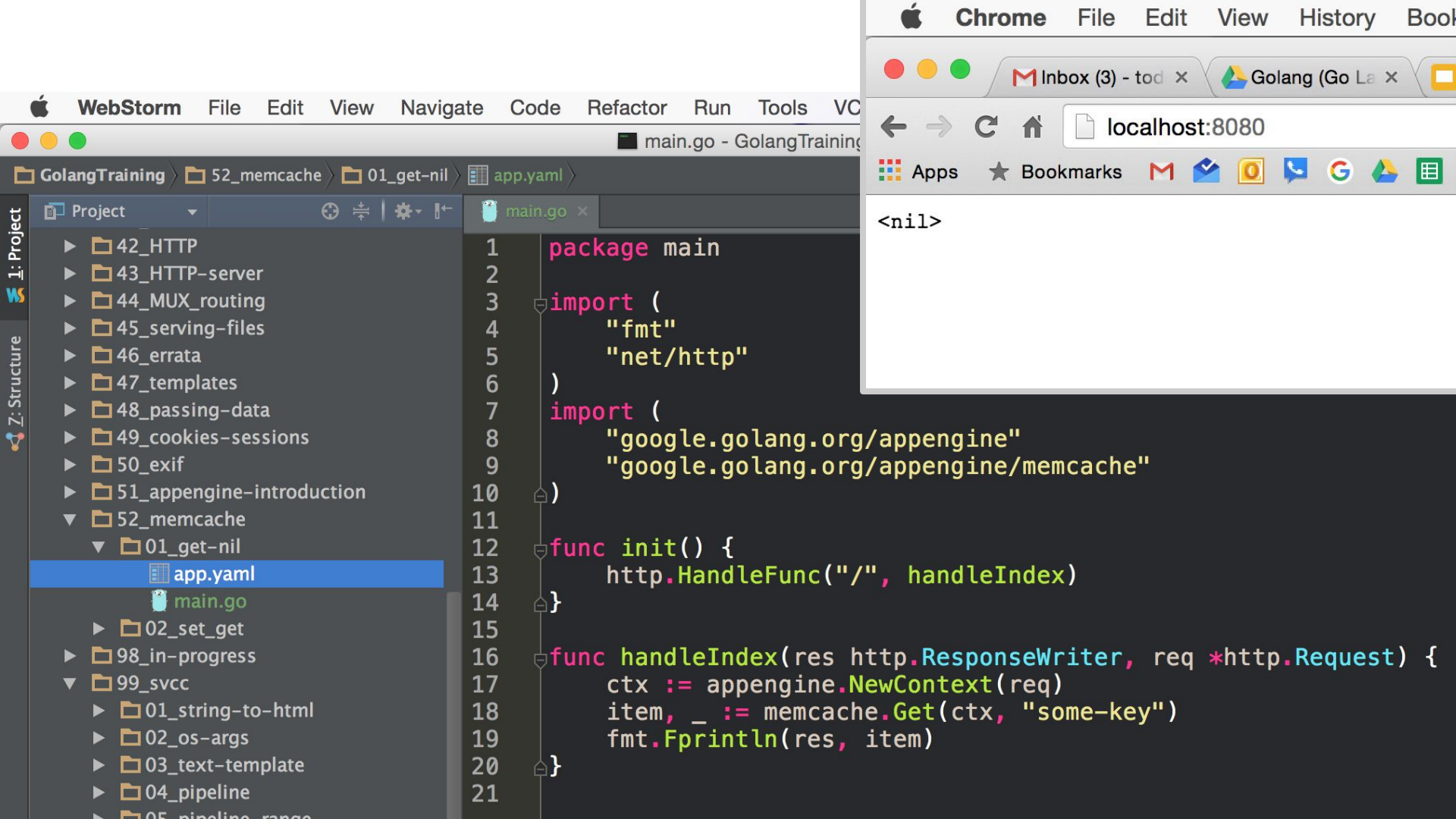
```
item1 := &memcache.Item{
    Key:   "foo",
    Value: []byte("bar"),
}
if err := memcache.Set(c, item1); err != nil {
    return err
}
```

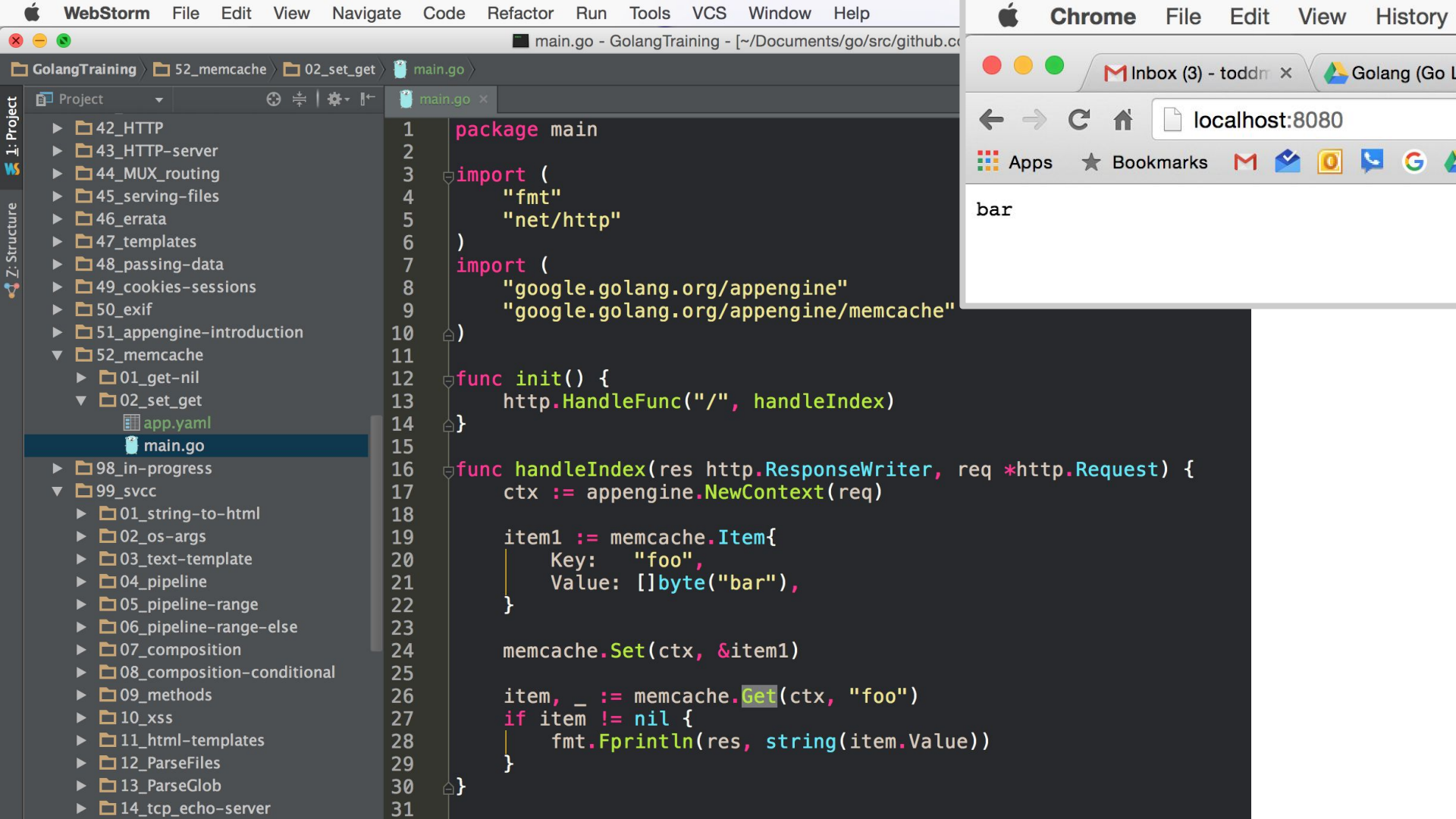
Index

```
item0, err := memcache.Get(c, "key")
if err != nil && err != memcache.ErrCacheMiss {
    return err
}
if err == nil {
    fmt.Fprintf(w, "memcache hit: Key=%q Val=[% x]\n", item0.Key, item0.Value)
} else {
    fmt.Fprintf(w, "memcache miss\n")
}
```

and

```
item1 := &memcache.Item{
    Key:   "foo",
    Value: []byte("bar"),
}
if err := memcache.Set(c, item1); err != nil {
    return err
}
```


Chrome File Edit View History Book
Inbox (3) - tod
Golang (Go La
localhost:8080
Apps
Bookmarks
<nil>
WebStorm File Edit View Navigate Code Refactor Run Tools VC
main.go - GolangTraining
GolangTraining
52_memcache
01_get-nil
app.yaml
Project
main.go
42_HTTP
43_HTTP-server
44_MUX_routing
45_serving-files
46_errata
47_templates
48_passing-data
49_cookies-sessions
50_exif
51_appengine-introduction
52_memcache
01_get-nil
app.yaml
main.go
02_set_get
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
1: Project
Z: Structure
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6)
7 import (
8 "google.golang.org/appengine"
9 "google.golang.org/appengine/memcache"
10)
11
12 func init() {
13 http.HandleFunc("/", handleIndex)
14 }
15
16 func handleIndex(res http.ResponseWriter, req *http.Request) {
17 ctx := appengine.NewContext(req)
18 item, _ := memcache.Get(ctx, "some-key")
19 fmt.Fprintln(res, item)
20 }
21


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.c
GolangTraining 52_memcache 02_set_get main.go
Project
1: Project
Z: Structure
42_HTTP
43_HTTP-server
44_MUX_routing
45_serving-files
46_errata
47_templates
48_passing-data
49_cookies-sessions
50_exif
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
app.yaml
main.go
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
13_ParseGlob
14_tcp_echo-server
main.go
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6)
7 import (
8 "google.golang.org/appengine"
9 "google.golang.org/appengine/memcache"
10)
11
12 func init() {
13 http.HandleFunc("/", handleIndex)
14 }
15
16 func handleIndex(res http.ResponseWriter, req *http.Request) {
17 ctx := appengine.NewContext(req)
18
19 item1 := memcache.Item{
20 Key: "foo",
21 Value: []byte("bar"),
22 }
23
24 memcache.Set(ctx, &item1)
25
26 item, _ := memcache.Get(ctx, "foo")
27 if item != nil {
28 fmt.Fprintln(res, string(item.Value))
29 }
30 }
31
Chrome File Edit View History
Inbox (3) - toddm
Golang (Go L
localhost:8080
Apps
Bookmarks
bar

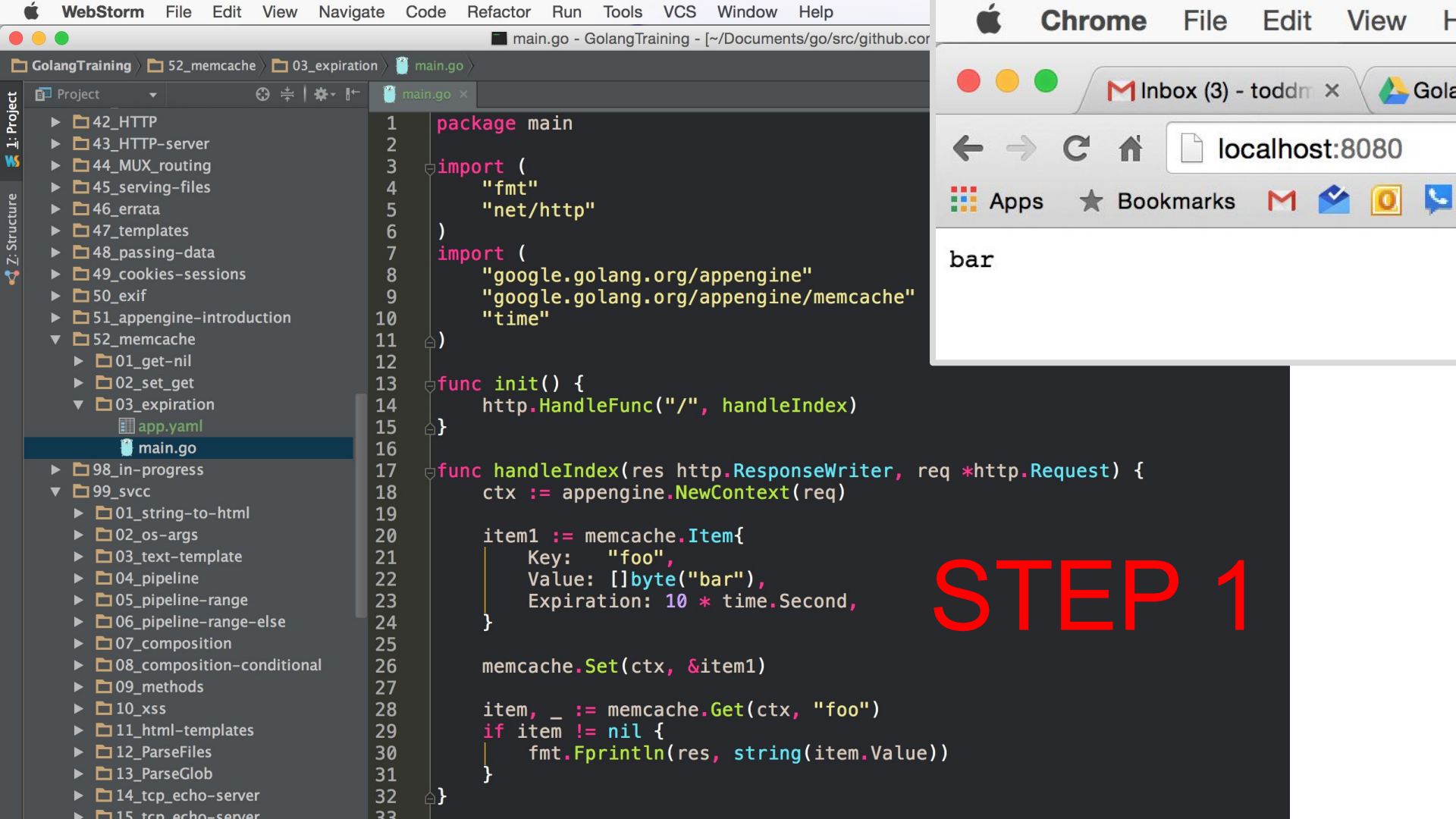
add

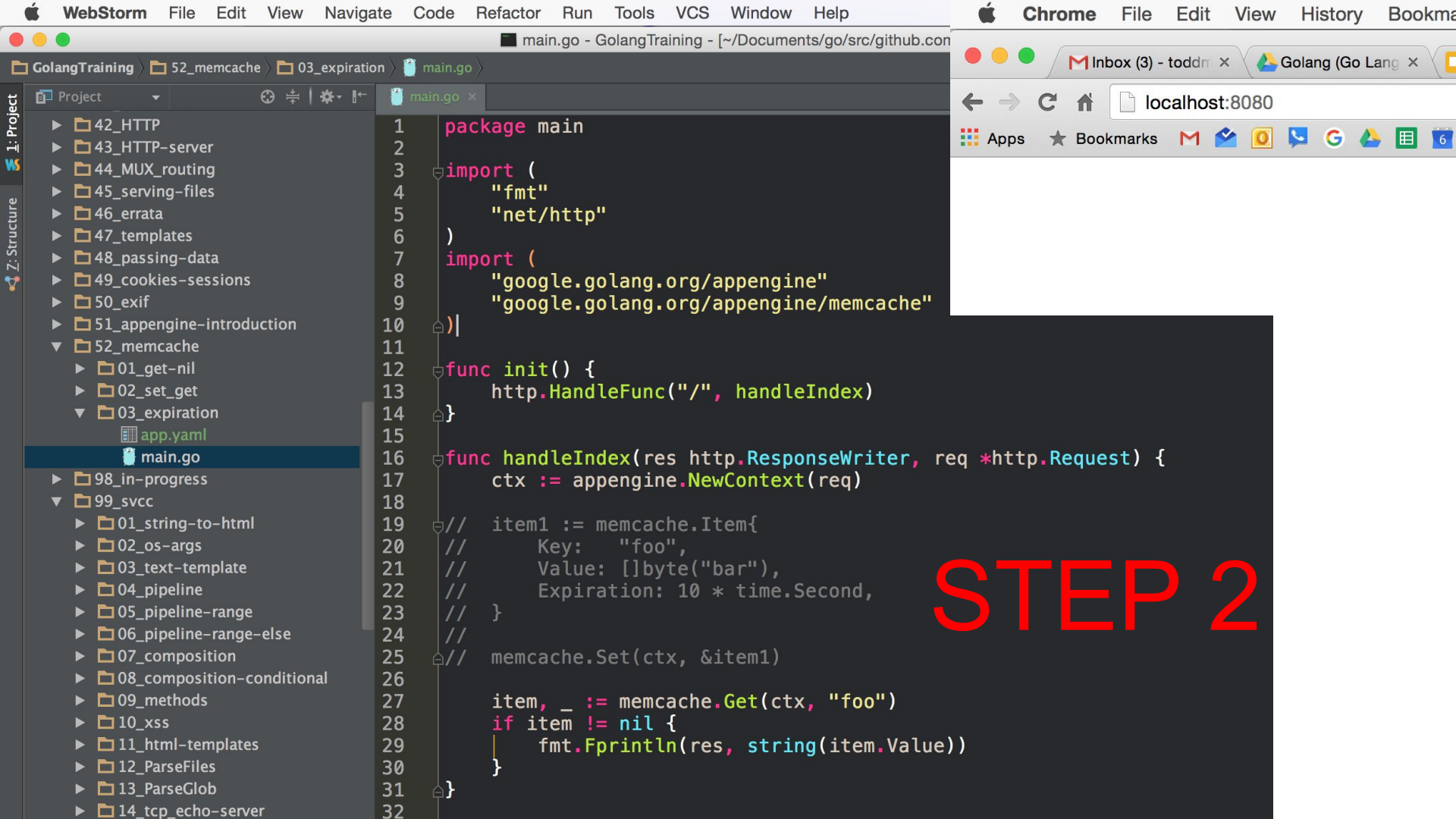
func Add

```
func Add(c appengine.Context, item *Item) error
```

Add writes the given item, if no value already exists for its key. ErrNotStored is returned if that condition is not met.

expiration


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.com
GolangTraining 52_memcache 03_expiration main.go
Project
42_HTTP
43_HTTP-server
44_MUX_routing
45_serving-files
46_errata
47_templates
48_passing-data
49_cookies-sessions
50_exif
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
app.yaml
main.go
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
13_ParseGlob
14_tcp_echo-server
main.go
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6)
7 import (
8 "google.golang.org/appengine"
9 "google.golang.org/appengine/memcache"
10 "time"
11)
12
13 func init() {
14 http.HandleFunc("/", handleIndex)
15 }
16
17 func handleIndex(res http.ResponseWriter, req *http.Request) {
18 ctx := appengine.NewContext(req)
19
20 item1 := memcache.Item{
21 Key: "foo",
22 Value: []byte("bar"),
23 Expiration: 10 * time.Second,
24 }
25
26 memcache.Set(ctx, &item1)
27
28 item, _ := memcache.Get(ctx, "foo")
29 if item != nil {
30 fmt.Fprintln(res, string(item.Value))
31 }
32 }
Chrome File Edit View
Inbox (3) - toddn
Gola
localhost:8080
Apps
Bookmarks
bar
STEP 1


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.com
GolangTraining 52_memcache 03_expiration main.go
Project
1: Project
Z: Structure
42_HTTP
43_HTTP-server
44_MUX_routing
45_serving-files
46_errata
47_templates
48_passing-data
49_cookies-sessions
50_exif
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
app.yaml
main.go
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
13_ParseGlob
14_tcp_echo-server
main.go
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6)
7 import (
8 "google.golang.org/appengine"
9 "google.golang.org/appengine/memcache"
10)
11
12 func init() {
13 http.HandleFunc("/", handleIndex)
14 }
15
16 func handleIndex(res http.ResponseWriter, req *http.Request) {
17 ctx := appengine.NewContext(req)
18
19 // item1 := memcache.Item{
20 // Key: "foo",
21 // Value: []byte("bar"),
22 // Expiration: 10 * time.Second,
23 // }
24 //
25 // memcache.Set(ctx, &item1)
26
27 item, _ := memcache.Get(ctx, "foo")
28 if item != nil {
29 fmt.Fprintln(res, string(item.Value))
30 }
31 }
32
Chrome File Edit View History Bookma
Inbox (3) - toddn
Golang (Go Lang
localhost:8080
Apps
Bookmarks
STEP 2

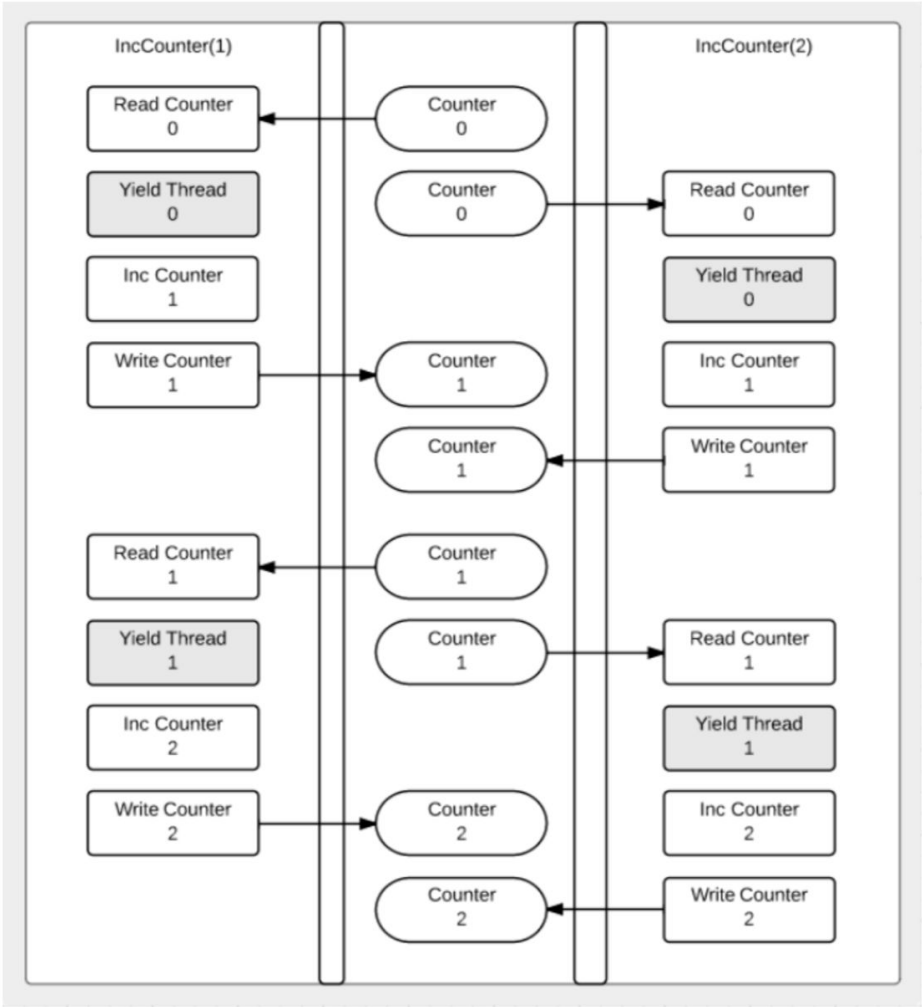
increment

func Increment

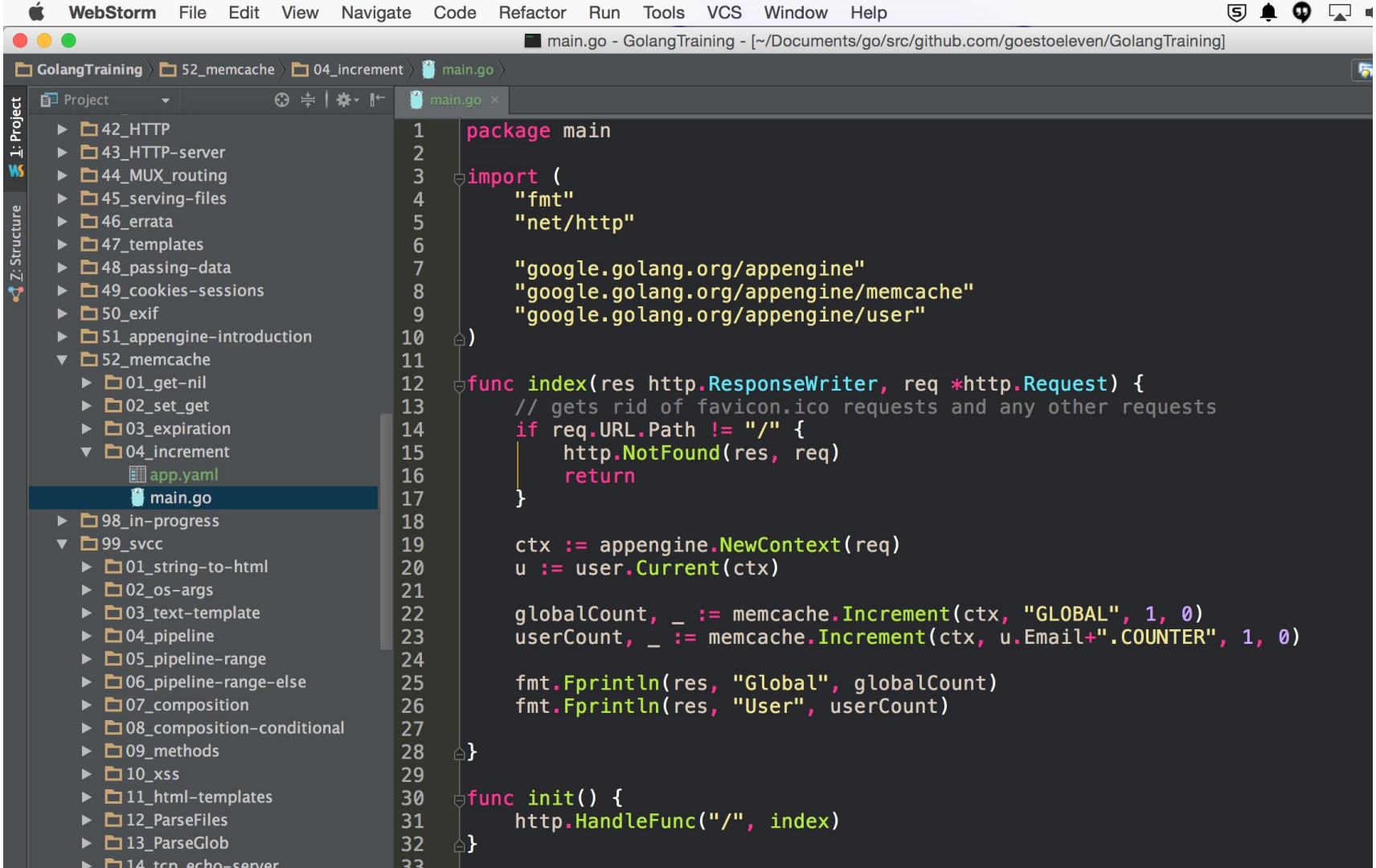
```
func Increment(c appengine.Context, key string, delta int64, initialValue uint64) (newValue uint64
```

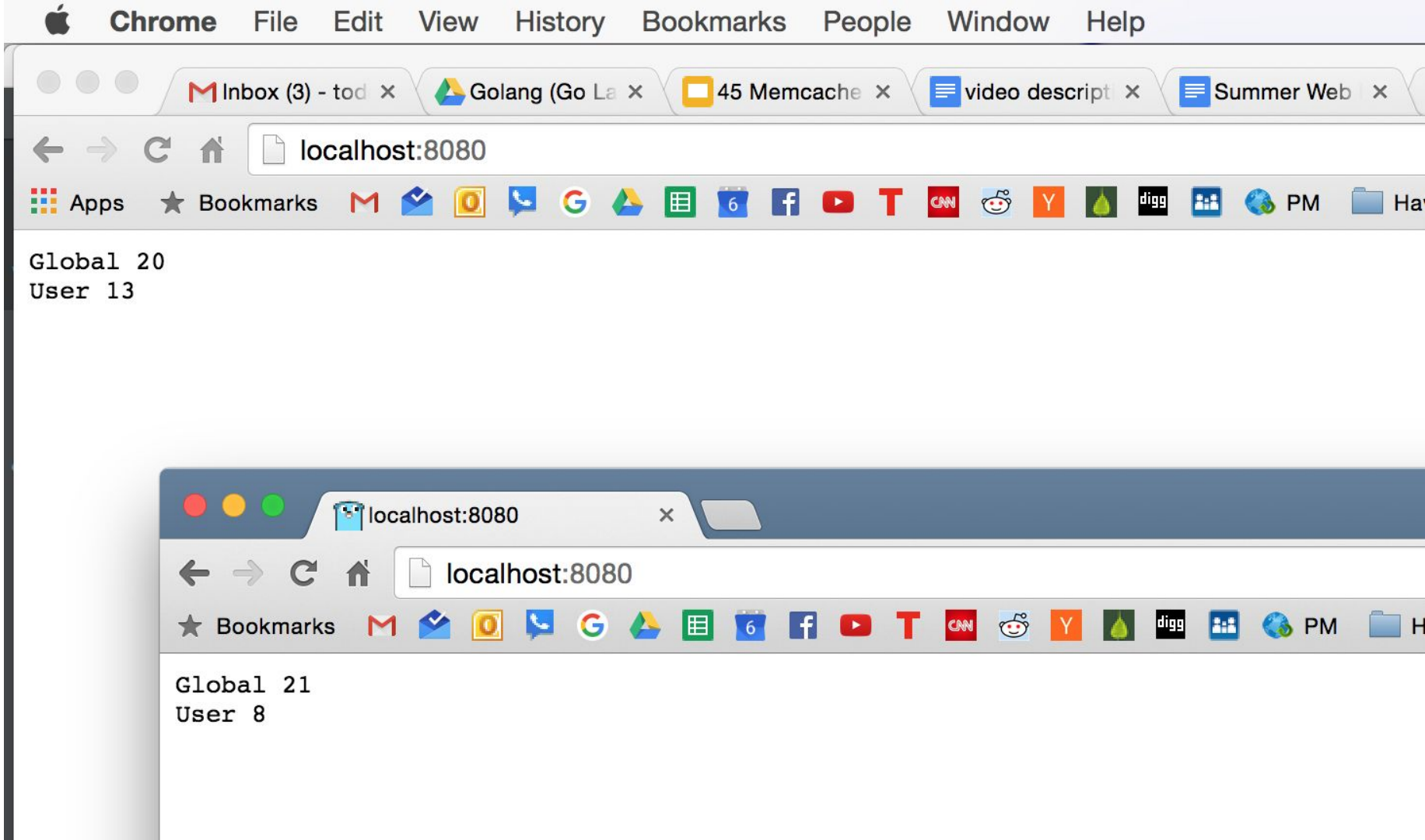
Increment atomically increments the decimal value in the given key by delta and returns the new value. The value must fit in a uint64. Overflow wraps around, and underflow is capped to zero. The provided delta may be negative. If the key doesn't exist in memcache, the provided initial value is used to atomically populate it before the delta is applied. The key must be at most 250 bytes in length.

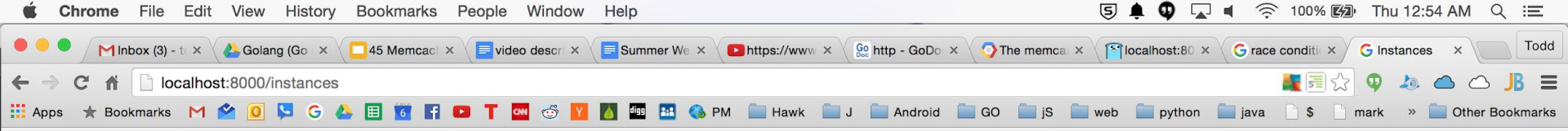
```
194
195 // Increment atomically increments the decimal value in the given key
196 // by delta and returns the new value. The value must fit in a uint64.
197 // Overflow wraps around, and underflow is capped to zero. The
198 // provided delta may be negative. If the key doesn't exist in
199 // memcache, the provided initial value is used to atomically
200 // populate it before the delta is applied.
201 // The key must be at most 250 bytes in length.
202 func Increment(c context.Context, key string, delta int64, initialValue uint64) (newValue uint64,
     err error) {
203     return incr(c, key, delta, &initialValue)
204 }
205
206 // IncrementExisting works like Increment but assumes that the key
207 // already exists in memcache and doesn't take an initial value.
208 // IncrementExisting can save work if calculating the initial value is
209 // expensive.
210 // An error is returned if the specified item can not be found.
211 func IncrementExisting(c context.Context, key string, delta int64) (newValue uint64, err error) {
212     return incr(c, key, delta, nil)
213 }
214
215 func incr(c context.Context, key string, delta int64, initialValue *uint64) (newValue uint64, err
     error) {
216     req := &pb.MemcacheIncrementRequest{
217         Key:          []byte(key),
218         InitialValue: initialValue,
219     }
220     if delta >= 0 {
```



source: William Kennedy, Go In Action


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining 52_memcache 04_increment main.go
Project
main.go
42_HTTP
43_HTTP-server
44_MUX_routing
45_serving-files
46_errata
47_templates
48_passing-data
49_cookies-sessions
50_exif
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
04_increment
app.yaml
main.go
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
13_ParseGlob
1: Project
Z: Structure
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6
7 "google.golang.org/appengine"
8 "google.golang.org/appengine/memcache"
9 "google.golang.org/appengine/user"
10)
11
12 func index(res http.ResponseWriter, req *http.Request) {
13 // gets rid of favicon.ico requests and any other requests
14 if req.URL.Path != "/" {
15 http.NotFound(res, req)
16 return
17 }
18
19 ctx := appengine.NewContext(req)
20 u := user.Current(ctx)
21
22 globalCount, _ := memcache.Increment(ctx, "GLOBAL", 1, 0)
23 userCount, _ := memcache.Increment(ctx, u.Email+".COUNTER", 1, 0)
24
25 fmt.Fprintln(res, "Global", globalCount)
26 fmt.Fprintln(res, "User", userCount)
27
28 }
29
30 func init() {
31 http.HandleFunc("/", index)
32 }


Chrome File Edit View History Bookmarks People Window Help
Inbox (3) - tod
Golang (Go La
45 Memcache
video descript
Summer Web
localhost:8080
Apps
Bookmarks
PM
Global 20
User 13
localhost:8080
localhost:8080
Bookmarks
PM
Global 21
User 8



Google App Engine

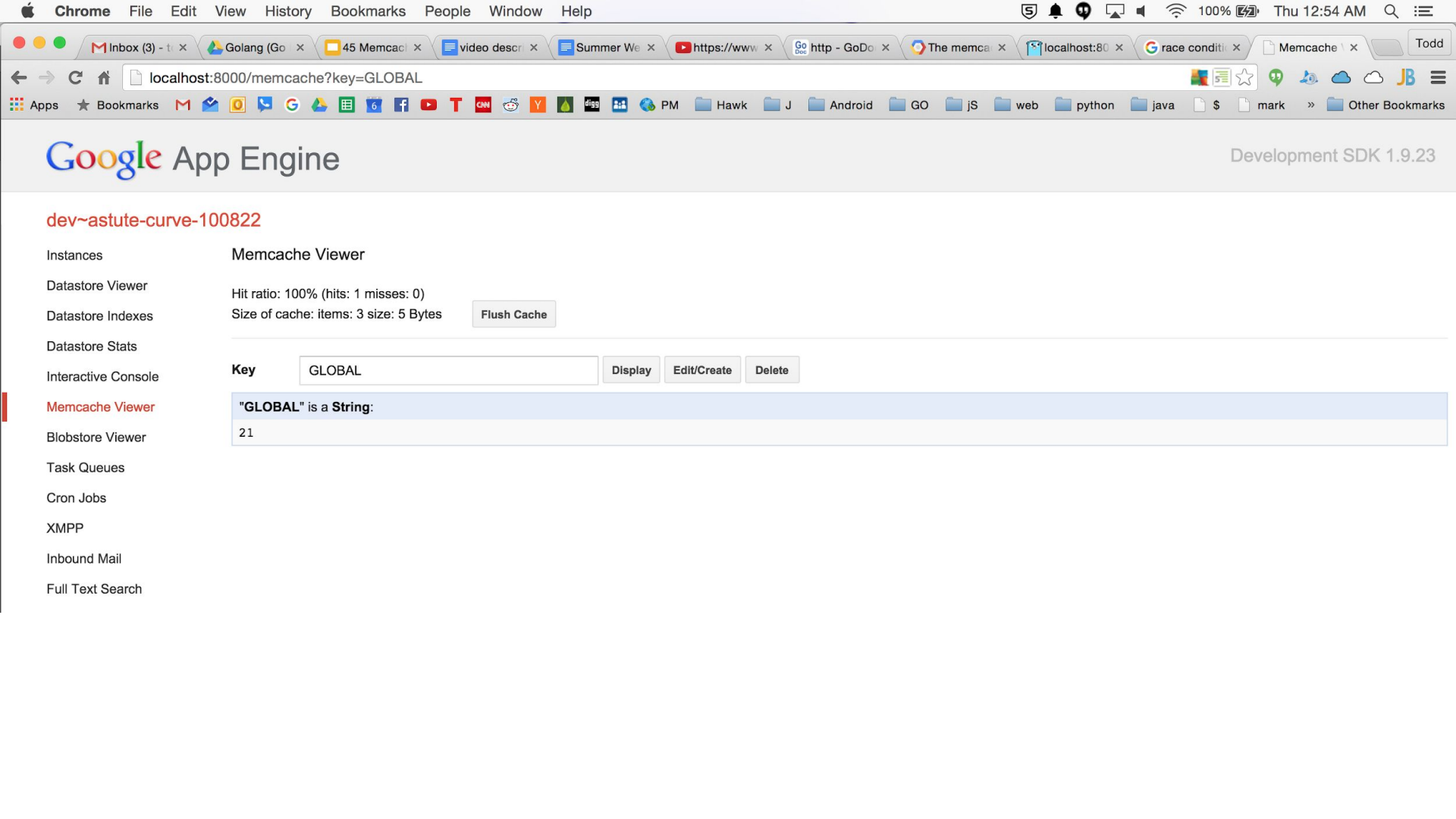
Development SDK 1.9.23

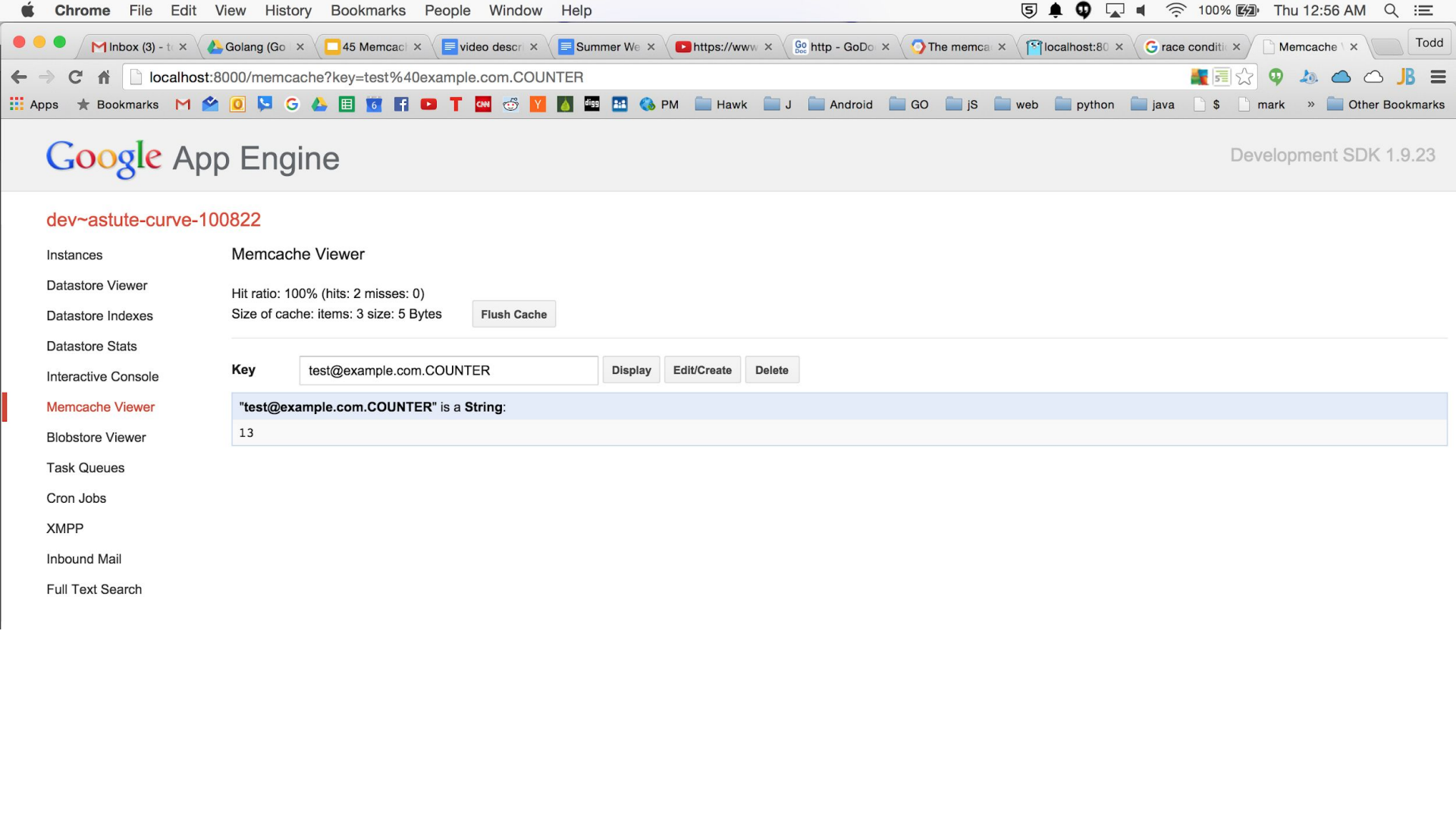
dev~astute-curve-100822

- Instances
- Datastore Viewer
- Datastore Indexes
- Datastore Stats
- Interactive Console
- Memcache Viewer
- Blobstore Viewer
- Task Queues
- Cron Jobs
- XMPP
- Inbound Mail
- Full Text Search

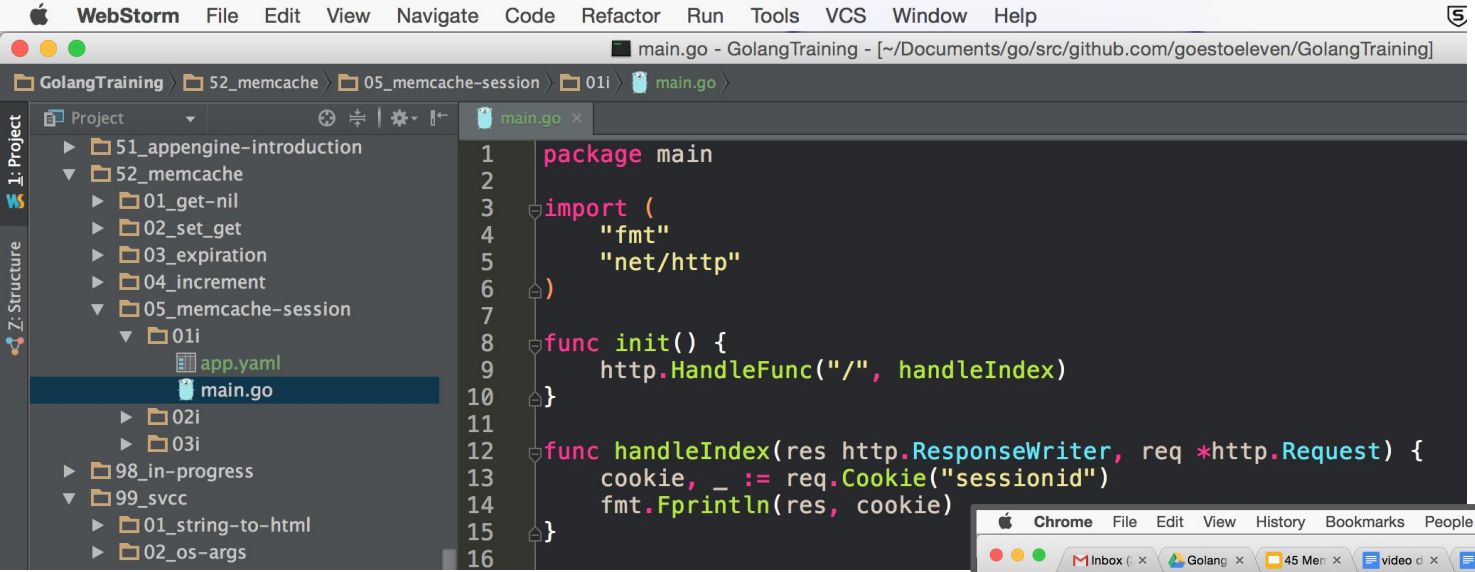
Instances

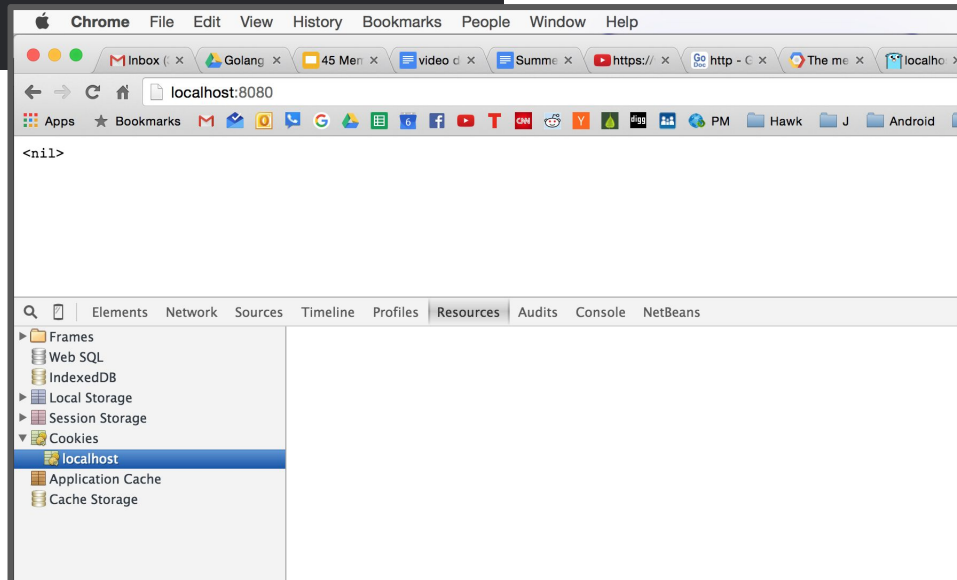
	Latency (ms)	QPS	Total Requests	Logs	Runtime
default					go
ecbf999b31fcbb7ec10a8a293dcce2f6b596	0.0	0.00	0		

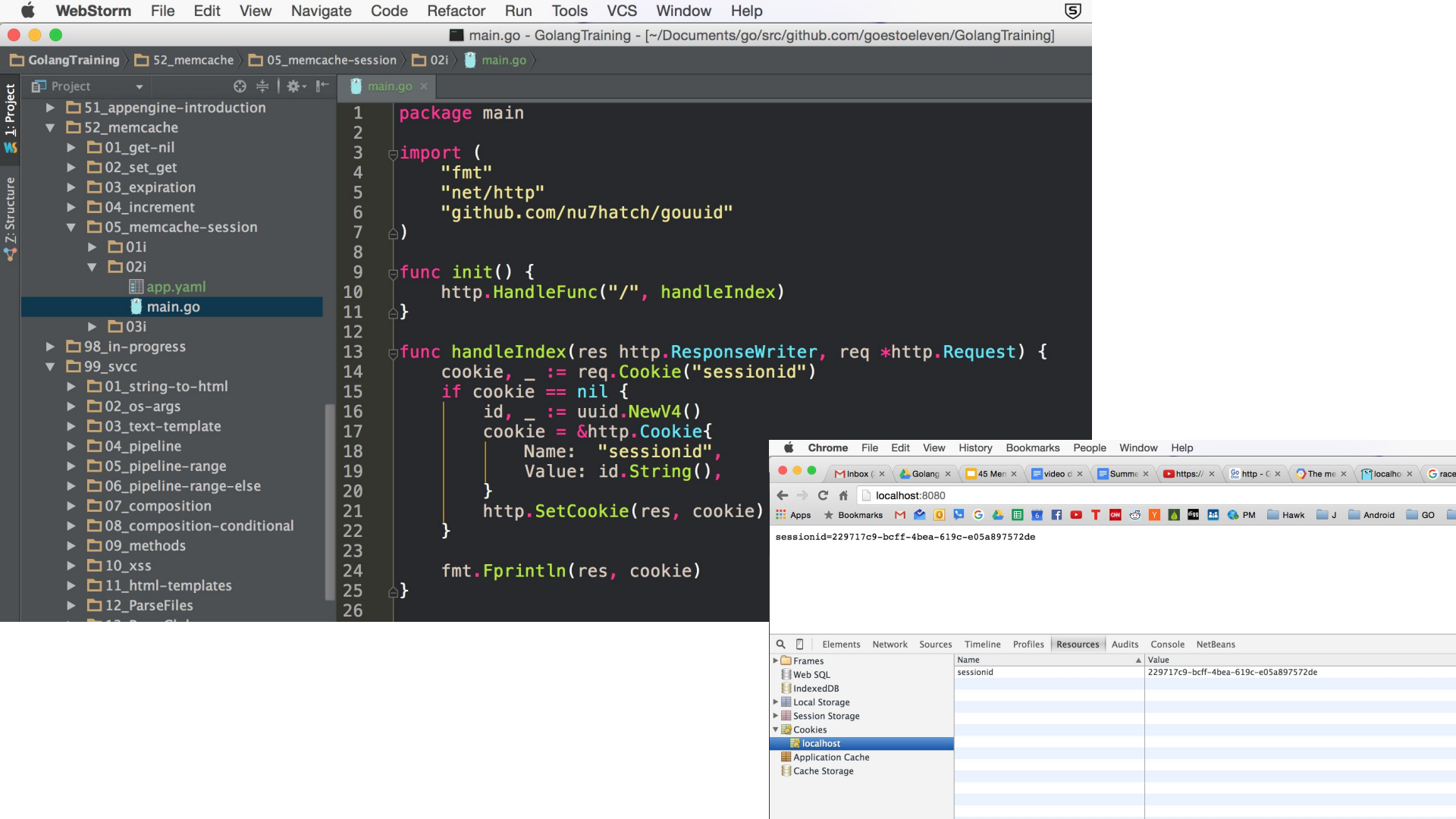

Chrome File Edit View History Bookmarks People Window Help
100% Thu 12:54 AM
Todd
localhost:8000/memcache?key=GLOBAL
Apps Bookmarks PM Hawk J Android GO jS web python java $ mark Other Bookmarks
Google App Engine
Development SDK 1.9.23
dev~astute-curve-100822
Instances
Datastore Viewer
Datastore Indexes
Datastore Stats
Interactive Console
Memcache Viewer
Blobstore Viewer
Task Queues
Cron Jobs
XMPP
Inbound Mail
Full Text Search
Memcache Viewer
Hit ratio: 100% (hits: 1 misses: 0)
Size of cache: items: 3 size: 5 Bytes
Flush Cache
Key
GLOBAL
Display
Edit/Create
Delete
"GLOBAL" is a String:
21

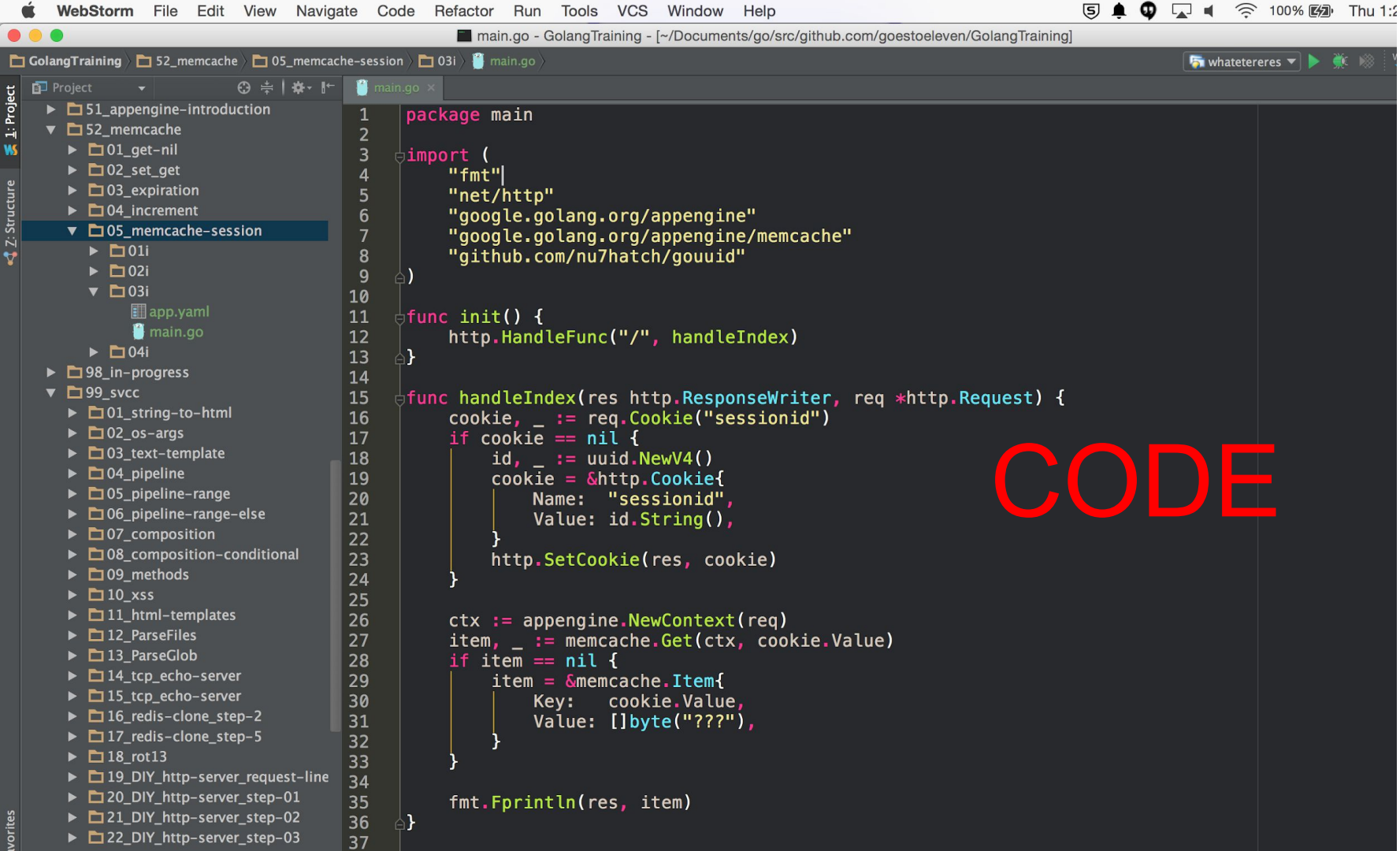

Chrome File Edit View History Bookmarks People Window Help
100% Thu 12:56 AM
Todd
localhost:8000/memcache?key=test%40example.com.COUNTER
Apps Bookmarks PM Hawk J Android GO jS web python java $ mark Other Bookmarks
Google App Engine
Development SDK 1.9.23
dev~astute-curve-100822
Instances
Datastore Viewer
Datastore Indexes
Datastore Stats
Interactive Console
Memcache Viewer
Blobstore Viewer
Task Queues
Cron Jobs
XMPP
Inbound Mail
Full Text Search
Memcache Viewer
Hit ratio: 100% (hits: 2 misses: 0)
Size of cache: items: 3 size: 5 Bytes
Flush Cache
Key
test@example.com.COUNTER
Display
Edit/Create
Delete
"test@example.com.COUNTER" is a String:
13

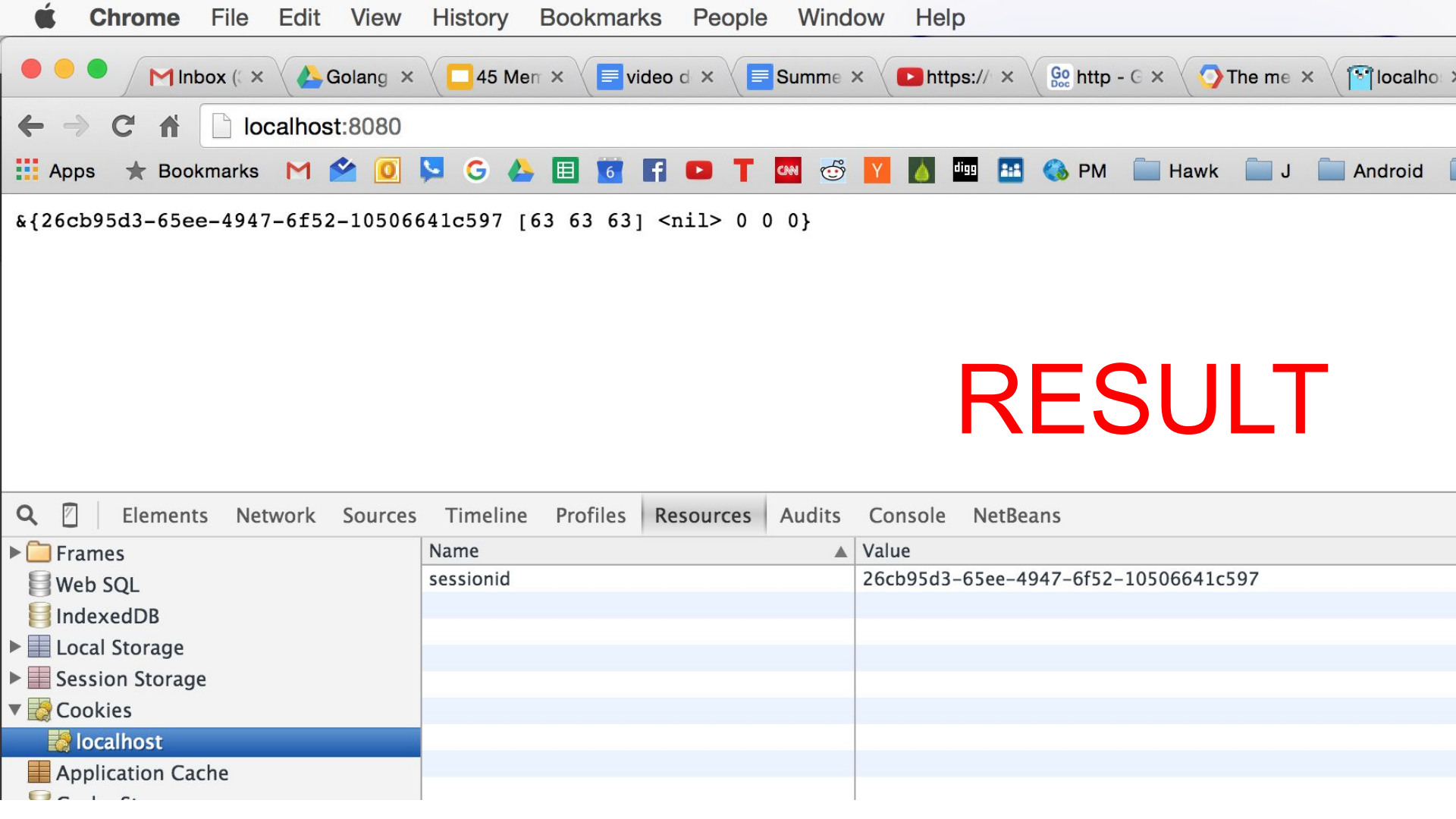
sessions with memcache

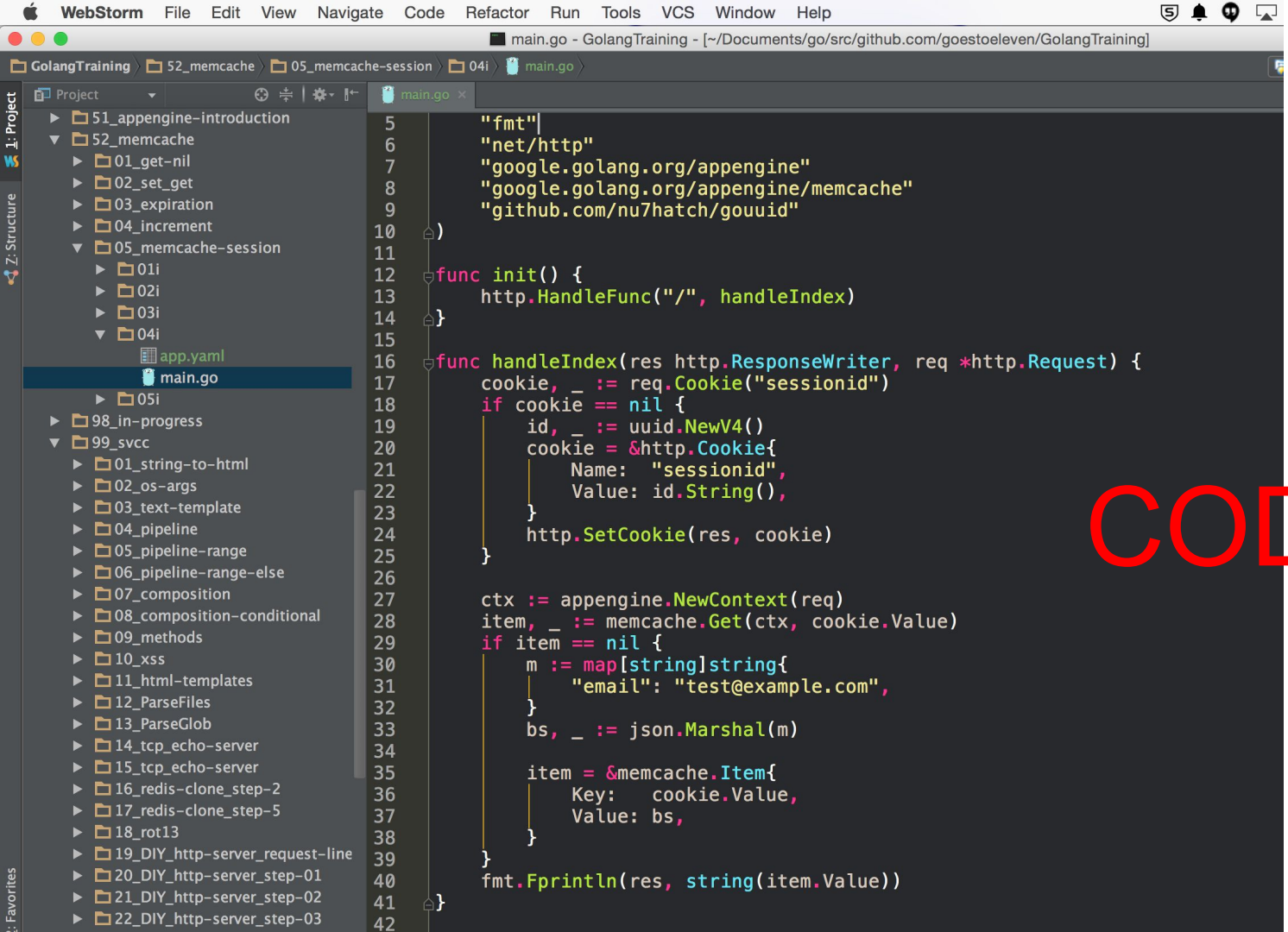

WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining 52_memcache 05_memcache-session 01i main.go
Project
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
04_increment
05_memcache-session
01i
app.yaml
main.go
02i
03i
98_in-progress
99_svcc
01_string-to-html
02_os-args
main.go
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6)
7
8 func init() {
9 http.HandleFunc("/", handleIndex)
10 }
11
12 func handleIndex(res http.ResponseWriter, req *http.Request) {
13 cookie, _ := req.Cookie("sessionid")
14 fmt.Fprintln(res, cookie)
15 }
16

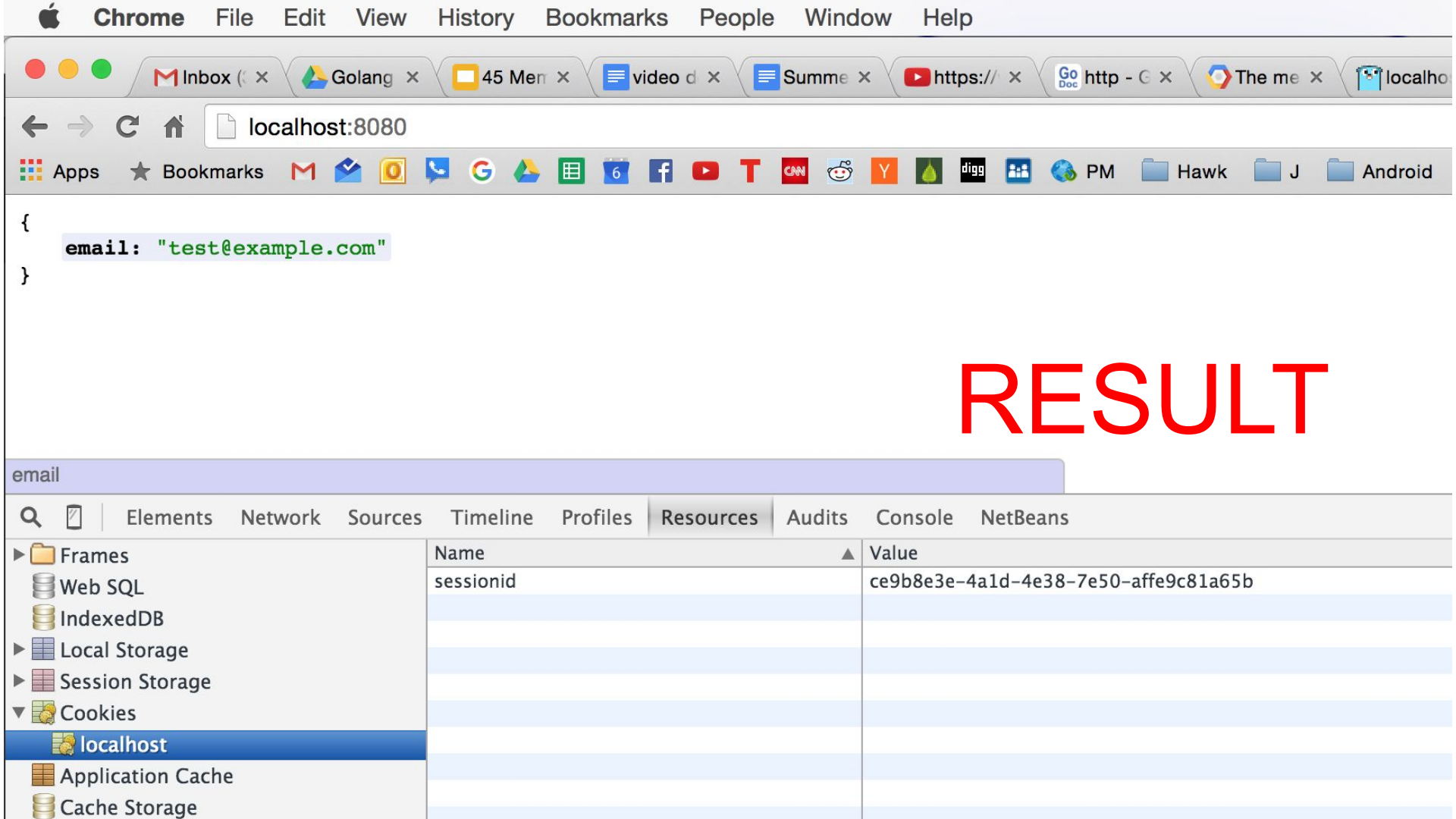

Chrome File Edit View History Bookmarks People Window Help
localhost:8080
Apps Bookmarks PM Hawk J Android
<nil>
Elements Network Sources Timeline Profiles Resources Audits Console NetBeans
Frames
Web SQL
IndexedDB
Local Storage
Session Storage
Cookies
localhost
Application Cache
Cache Storage


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining 52_memcache 05_memcache-session 02i main.go
Project
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
04_increment
05_memcache-session
01i
02i
app.yaml
main.go
03i
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
main.go
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6 "github.com/nu7hatch/gouuid"
7)
8
9 func init() {
10 http.HandleFunc("/", handleIndex)
11 }
12
13 func handleIndex(res http.ResponseWriter, req *http.Request) {
14 cookie, _ := req.Cookie("sessionid")
15 if cookie == nil {
16 id, _ := uuid.NewV4()
17 cookie = &http.Cookie{
18 Name: "sessionid",
19 Value: id.String(),
20 }
21 http.SetCookie(res, cookie)
22 }
23
24 fmt.Fprintln(res, cookie)
25 }
26
Chrome File Edit View History Bookmarks People Window Help
localhost:8080
Apps Bookmarks PM Hawk Android GO
sessionid=229717c9-bcff-4bea-619c-e05a897572de
Elements Network Sources Timeline Profiles Resources Audits Console NetBeans
Frames
Web SQL
IndexedDB
Local Storage
Session Storage
Cookies
localhost
Application Cache
Cache Storage
Name
Value
sessionid
229717c9-bcff-4bea-619c-e05a897572de


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
100% Thu 1:2
main.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining 52_memcache 05_memcache-session 03i main.go
whatetereres
Project
main.go
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
04_increment
05_memcache-session
01i
02i
03i
app.yaml
main.go
04i
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
13_ParseGlob
14_tcp_echo-server
15_tcp_echo-server
16_redis-clone_step-2
17_redis-clone_step-5
18_rot13
19_DIY_http-server_request-line
20_DIY_http-server_step-01
21_DIY_http-server_step-02
22_DIY_http-server_step-03
1: Project
7: Structure
vorites
1 package main
2
3 import (
4 "fmt"
5 "net/http"
6 "google.golang.org/appengine"
7 "google.golang.org/appengine/memcache"
8 "github.com/nu7hatch/gouuid"
9)
10
11 func init() {
12 http.HandleFunc("/", handleIndex)
13 }
14
15 func handleIndex(res http.ResponseWriter, req *http.Request) {
16 cookie, _ := req.Cookie("sessionid")
17 if cookie == nil {
18 id, _ := uuid.NewV4()
19 cookie = &http.Cookie{
20 Name: "sessionid",
21 Value: id.String(),
22 }
23 http.SetCookie(res, cookie)
24 }
25
26 ctx := appengine.NewContext(req)
27 item, _ := memcache.Get(ctx, cookie.Value)
28 if item == nil {
29 item = &memcache.Item{
30 Key: cookie.Value,
31 Value: []byte("???"),
32 }
33 }
34
35 fmt.Fprintln(res, item)
36 }
37
CODE

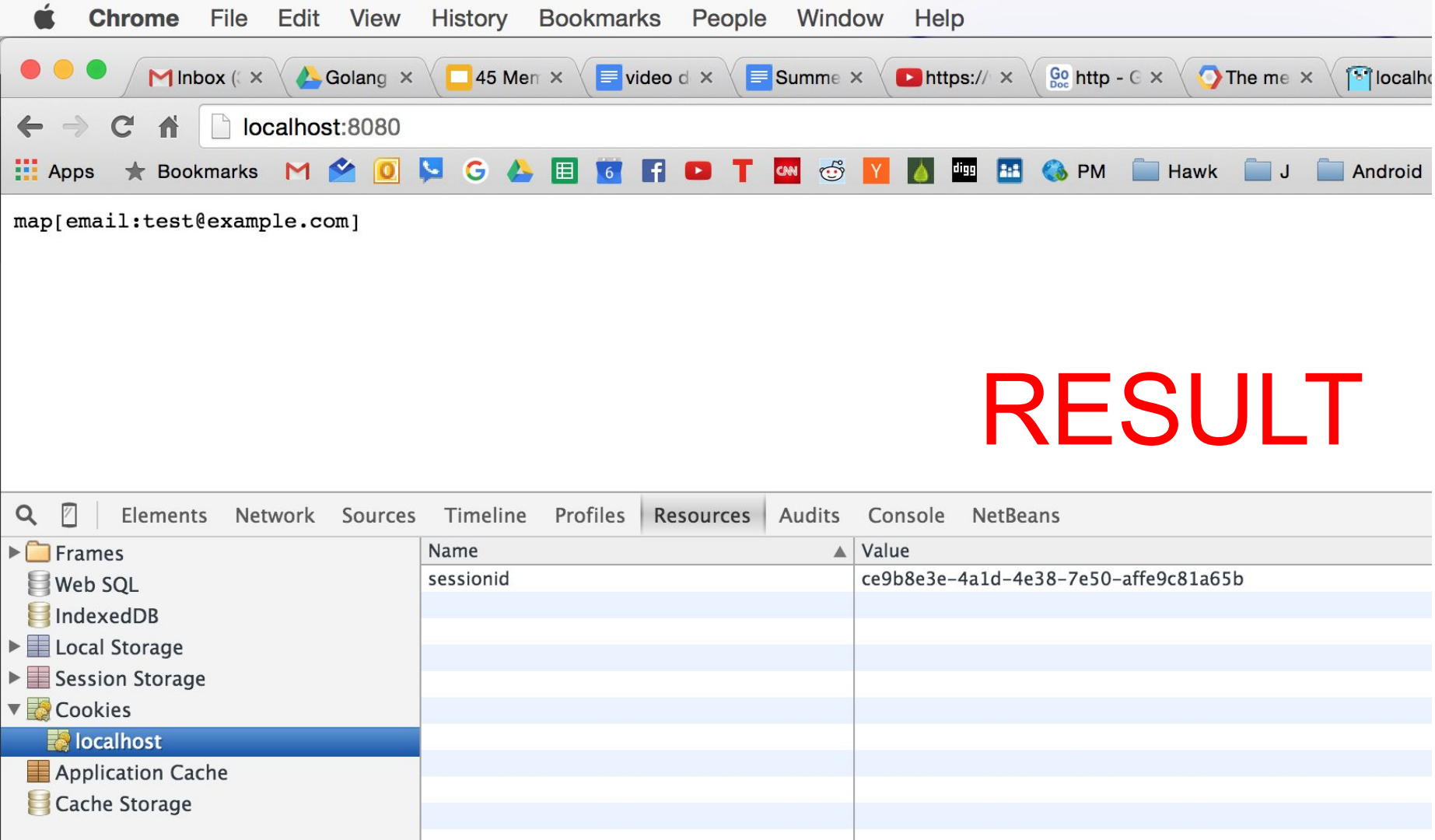

Chrome File Edit View History Bookmarks People Window Help
Inbox (
Golang
45 Men
video d
Summe
https://
http - G
The me
localho
localhost:8080
Apps
Bookmarks
PM
Hawk
J
Android
&{26cb95d3-65ee-4947-6f52-10506641c597 [63 63 63] <nil> 0 0 0}
RESULT
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
NetBeans
Frames
Web SQL
IndexedDB
Local Storage
Session Storage
Cookies
localhost
Application Cache
Name
Value
sessionid
26cb95d3-65ee-4947-6f52-10506641c597

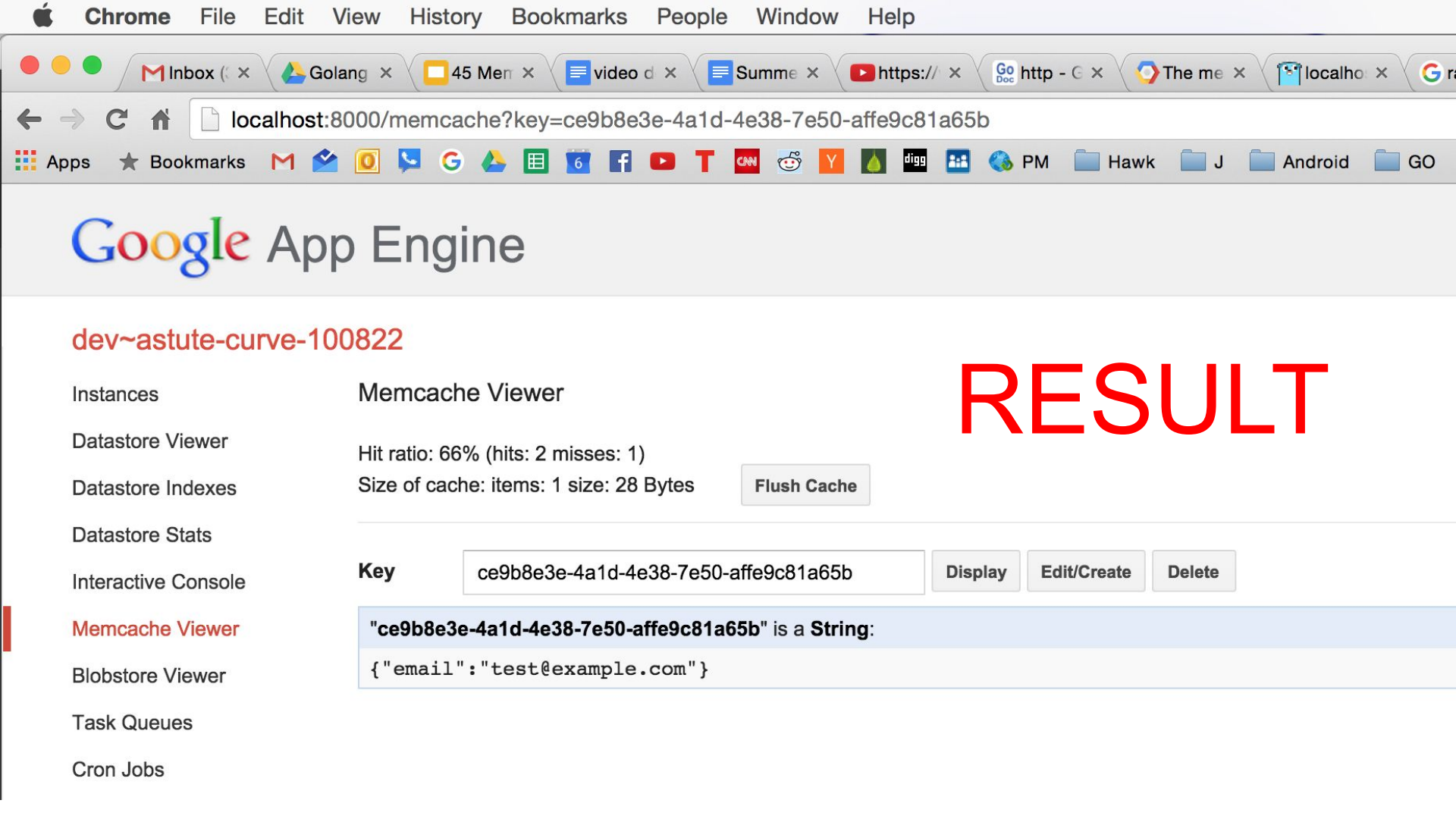

WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
main.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining 52_memcache 05_memcache-session 04i main.go
Project
main.go
51_appengine-introduction
52_memcache
01_get-nil
02_set_get
03_expiration
04_increment
05_memcache-session
01i
02i
03i
04i
app.yaml
main.go
05i
98_in-progress
99_svcc
01_string-to-html
02_os-args
03_text-template
04_pipeline
05_pipeline-range
06_pipeline-range-else
07_composition
08_composition-conditional
09_methods
10_xss
11_html-templates
12_ParseFiles
13_ParseGlob
14_tcp_echo-server
15_tcp_echo-server
16_redis-clone_step-2
17_redis-clone_step-5
18_rot13
19_DIY_http-server_request-line
20_DIY_http-server_step-01
21_DIY_http-server_step-02
22_DIY_http-server_step-03
"fmt"
"net/http"
"google.golang.org/appengine"
"google.golang.org/appengine/memcache"
"github.com/nu7hatch/gouuid"
)
func init() {
http.HandleFunc("/", handleIndex)
}
func handleIndex(res http.ResponseWriter, req *http.Request) {
cookie, _ := req.Cookie("sessionid")
if cookie == nil {
id, _ := uuid.NewV4()
cookie = &http.Cookie{
Name: "sessionid",
Value: id.String(),
}
http.SetCookie(res, cookie)
}
ctx := appengine.NewContext(req)
item, _ := memcache.Get(ctx, cookie.Value)
if item == nil {
m := map[string]string{
"email": "test@example.com",
}
bs, _ := json.Marshal(m)
item = &memcache.Item{
Key: cookie.Value,
Value: bs,
}
}
fmt.Fprintln(res, string(item.Value))
}
CODE


Chrome File Edit View History Bookmarks People Window Help
Inbox (Golang 45 Mer video d Summe https:// http - G The me localho
localhost:8080
Apps Bookmarks PM Hawk J Android
{
email: "test@example.com"
}
RESULT
email
Elements Network Sources Timeline Profiles Resources Audits Console NetBeans
Frames
Web SQL
IndexedDB
Local Storage
Session Storage
Cookies
localhost
Application Cache
Cache Storage
Name
Value
sessionid
ce9b8e3e-4a1d-4e38-7e50-affe9c81a65b

```go
func init() {
    http.HandleFunc("/", handleIndex)
}

func handleIndex(res http.ResponseWriter, req *http.Request) {
    cookie, _ := req.Cookie("sessionid")
    if cookie == nil {
        id, _ := uuid.NewV4()
        cookie = &http.Cookie{
            Name:  "sessionid",
            Value: id.String(),
        }
        http.SetCookie(res, cookie)
    }

    ctx := appengine.NewContext(req)
    item, _ := memcache.Get(ctx, cookie.Value)
    if item == nil {
        m := map[string]string{
            "email": "test@example.com",
        }
        bs, _ := json.Marshal(m)

        item = &memcache.Item{
            Key:   cookie.Value,
            Value: bs,
        }
        memcache.Set(ctx, item)
    }

    var m map[string]string
    json.Unmarshal(item.Value, &m)

    fmt.Fprintln(res, m)

}
```

CODE


Chrome File Edit View History Bookmarks People Window Help
Inbox (
Golang
45 Men
video d
Summe
https://
http - G
The me
localho
localhost:8080
Apps
Bookmarks
PM
Hawk
J
Android
map[email:test@example.com]
RESULT
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
NetBeans
Frames
Web SQL
IndexedDB
Local Storage
Session Storage
Cookies
localhost
Application Cache
Cache Storage
Name
Value
sessionid
ce9b8e3e-4a1d-4e38-7e50-affe9c81a65b


Chrome File Edit View History Bookmarks People Window Help
Inbox
Golang
45 Mer
video d
Summe
https://
http - G
The me
localho
localhost:8000/memcache?key=ce9b8e3e-4a1d-4e38-7e50-affe9c81a65b
Apps
Bookmarks
PM
Hawk
J
Android
GO
Google App Engine
dev~astute-curve-100822
Instances
Datastore Viewer
Datastore Indexes
Datastore Stats
Interactive Console
Memcache Viewer
Blobstore Viewer
Task Queues
Cron Jobs
Memcache Viewer
RESULT
Hit ratio: 66% (hits: 2 misses: 1)
Size of cache: items: 1 size: 28 Bytes
Flush Cache
Key
ce9b8e3e-4a1d-4e38-7e50-affe9c81a65b
Display
Edit/Create
Delete
"ce9b8e3e-4a1d-4e38-7e50-affe9c81a65b" is a String:
{"email":"test@example.com"}

photoblog with memcache