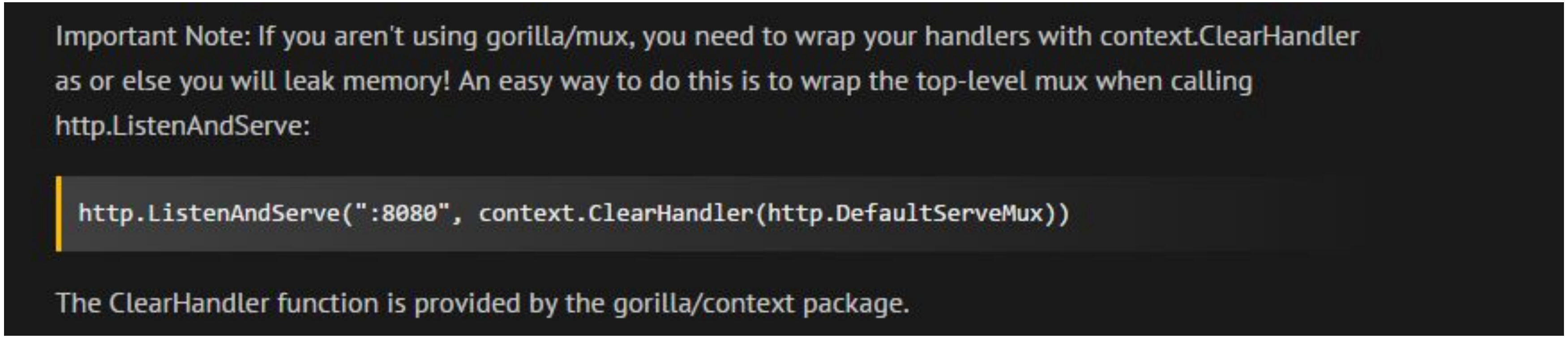
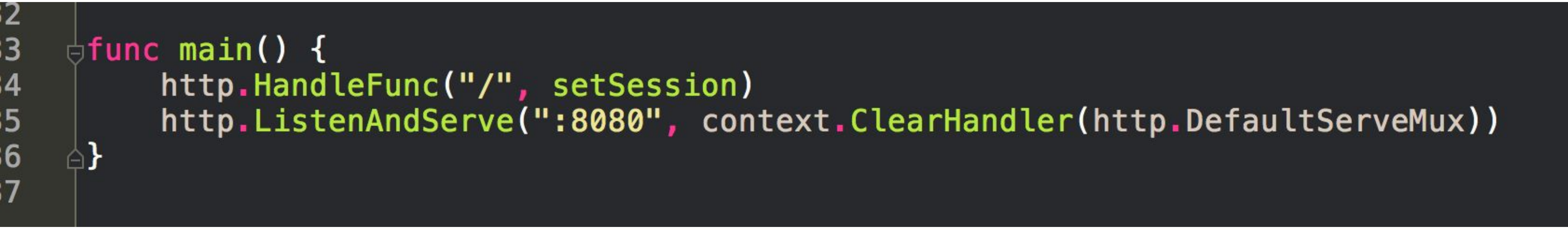
Gorilla Sessions

memory leak fix


Important Note: If you aren't using gorilla/mux, you need to wrap your handlers with context.ClearHandler as or else you will leak memory! An easy way to do this is to wrap the top-level mux when calling http.ListenAndServe:

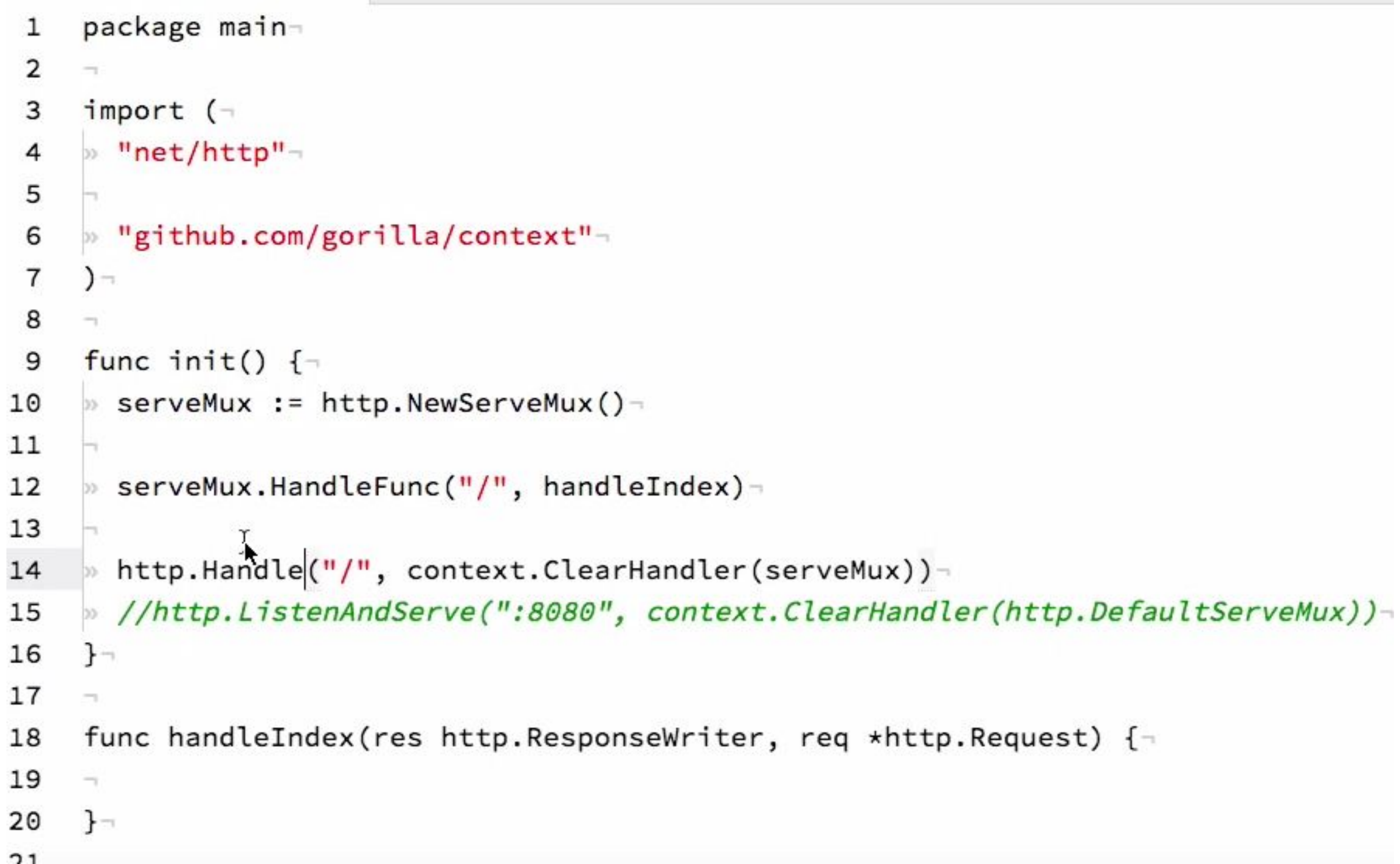
```
http.ListenAndServe(":8080", context.ClearHandler(http.DefaultServeMux))
```

The ClearHandler function is provided by the gorilla/context package.

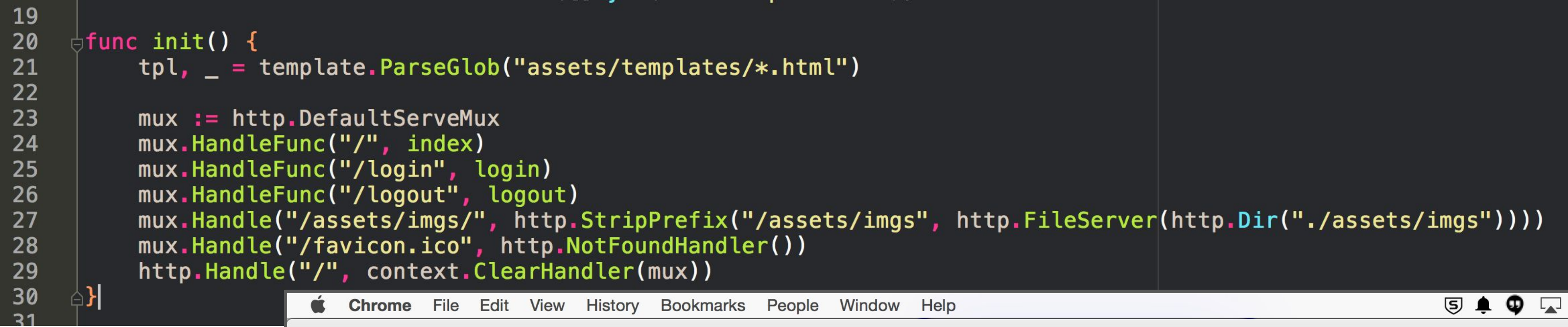


```go

func main() {
	http.HandleFunc("/", setSession)
	http.ListenAndServe(":8080", context.ClearHandler(http.DefaultServeMux))
}

```

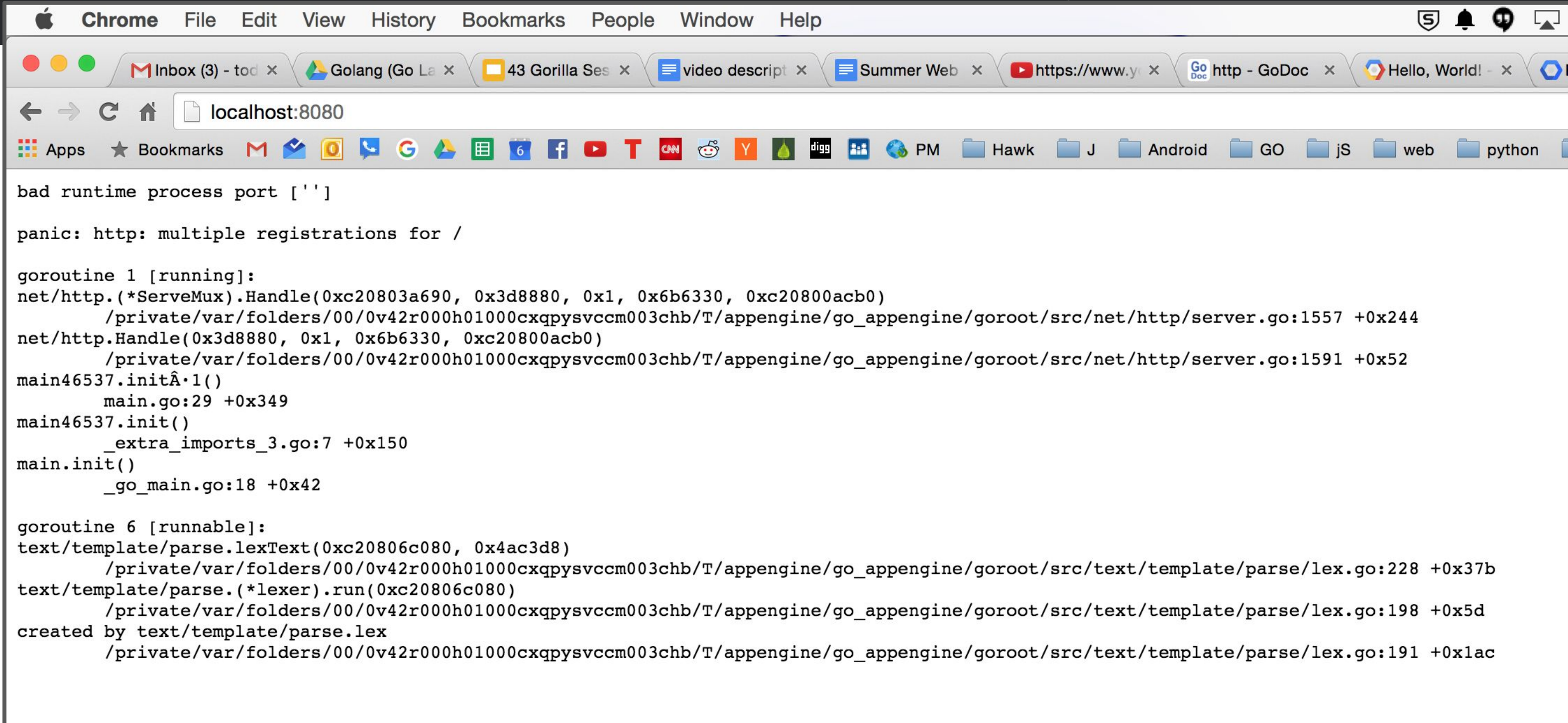
on App Engine



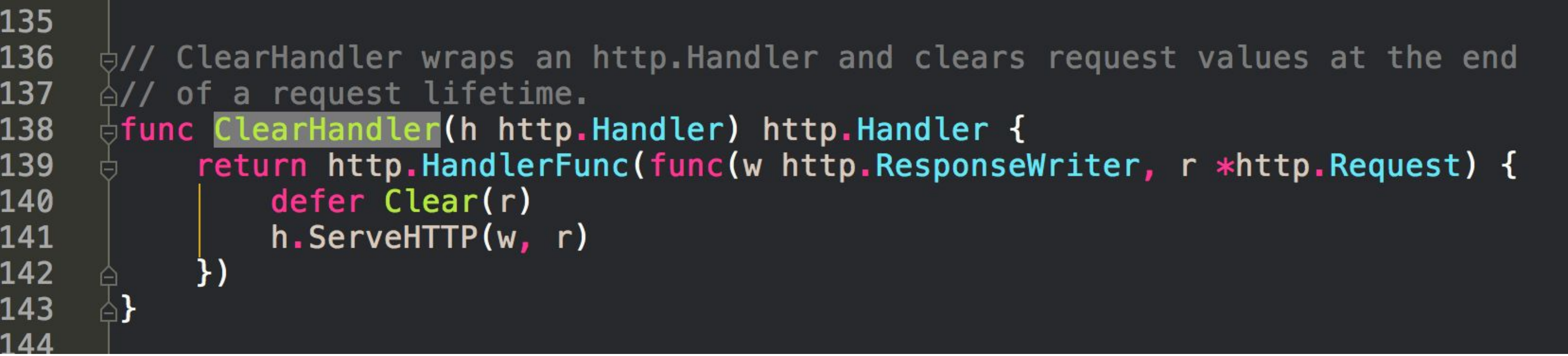
```go
package main

import (
	"net/http"

	"github.com/gorilla/context"
)

func init() {
	serveMux := http.NewServeMux()

	serveMux.HandleFunc("/", handleIndex)

	http.Handle("/", context.ClearHandler(serveMux))
	//http.ListenAndServe(":8080", context.ClearHandler(http.DefaultServeMux))
}

func handleIndex(res http.ResponseWriter, req *http.Request) {

}
```



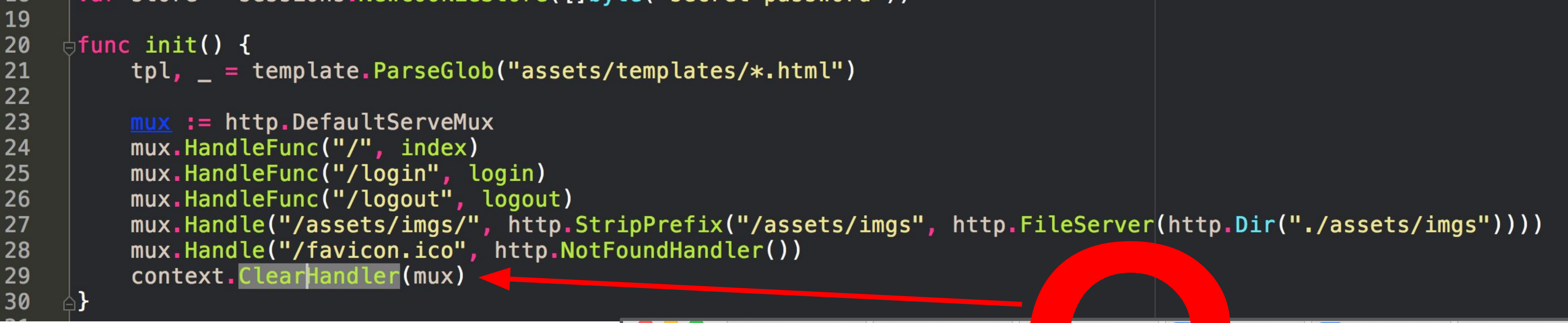
```go
func init() {
	tpl, _ = template.ParseGlob("assets/templates/*.html")

	mux := http.DefaultServeMux
	mux.HandleFunc("/", index)
	mux.HandleFunc("/login", login)
	mux.HandleFunc("/logout", logout)
	mux.Handle("/assets/imgs/", http.StripPrefix("/assets/imgs", http.FileServer(http.Dir("./assets/imgs"))))
	mux.Handle("/favicon.ico", http.NotFoundHandler())
	http.Handle("/", context.ClearHandler(mux))
}
```



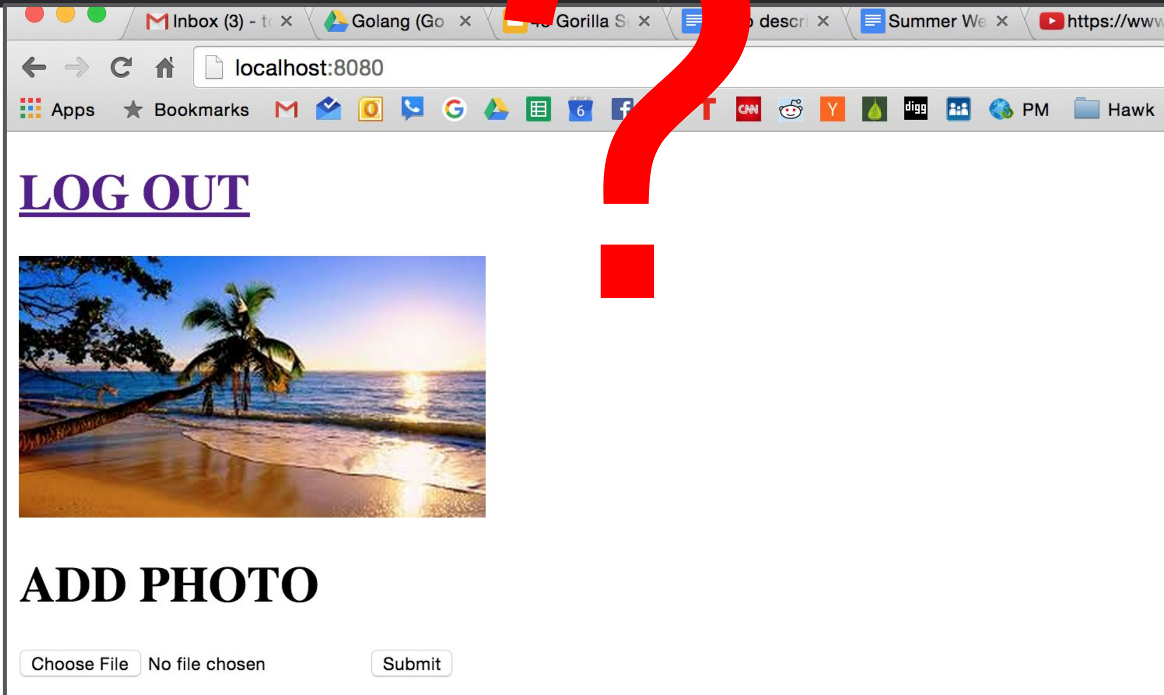
```
bad runtime process port ['']

panic: http: multiple registrations for /

goroutine 1 [running]:
net/http.(*ServeMux).Handle(0xc20803a690, 0x3d8880, 0x1, 0x6b6330, 0xc20800acb0)
	/private/var/folders/00/0v42r000h01000cxqpysvccm003chb/T/appengine/go_appengine/goroot/src/net/http/server.go:1557 +0x244
net/http.Handle(0x3d8880, 0x1, 0x6b6330, 0xc20800acb0)
	/private/var/folders/00/0v42r000h01000cxqpysvccm003chb/T/appengine/go_appengine/goroot/src/net/http/server.go:1591 +0x52
main46537.initÂ·1()
	main.go:29 +0x349
main46537.init()
	_extra_imports_3.go:7 +0x150
main.init()
	_go_main.go:18 +0x42

goroutine 6 [runnable]:
text/template/parse.lexText(0xc20806c080, 0x4ac3d8)
	/private/var/folders/00/0v42r000h01000cxqpysvccm003chb/T/appengine/go_appengine/goroot/src/text/template/parse/lex.go:228 +0x37b
text/template/parse.(*lexer).run(0xc20806c080)
	/private/var/folders/00/0v42r000h01000cxqpysvccm003chb/T/appengine/go_appengine/goroot/src/text/template/parse/lex.go:198 +0x5d
created by text/template/parse.lex
	/private/var/folders/00/0v42r000h01000cxqpysvccm003chb/T/appengine/go_appengine/goroot/src/text/template/parse/lex.go:191 +0x1ac
```



```go

// ClearHandler wraps an http.Handler and clears request values at the end
// of a request lifetime.
func ClearHandler(h http.Handler) http.Handler {
	return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
		defer Clear(r)
		h.ServeHTTP(w, r)
	})
}

```



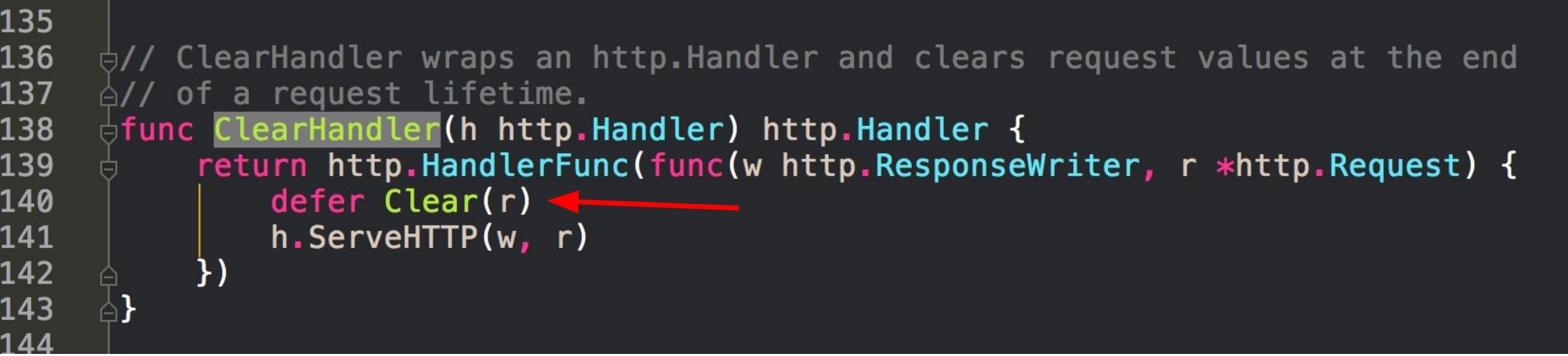
```go
func init() {
	tpl, _ = template.ParseGlob("assets/templates/*.html")

	mux := http.DefaultServeMux
	mux.HandleFunc("/", index)
	mux.HandleFunc("/login", login)
	mux.HandleFunc("/logout", logout)
	mux.Handle("/assets/imgs/", http.StripPrefix("/assets/imgs", http.FileServer(http.Dir("./assets/imgs"))))
	mux.Handle("/favicon.ico", http.NotFoundHandler())
	context.ClearHandler(mux)
}
```



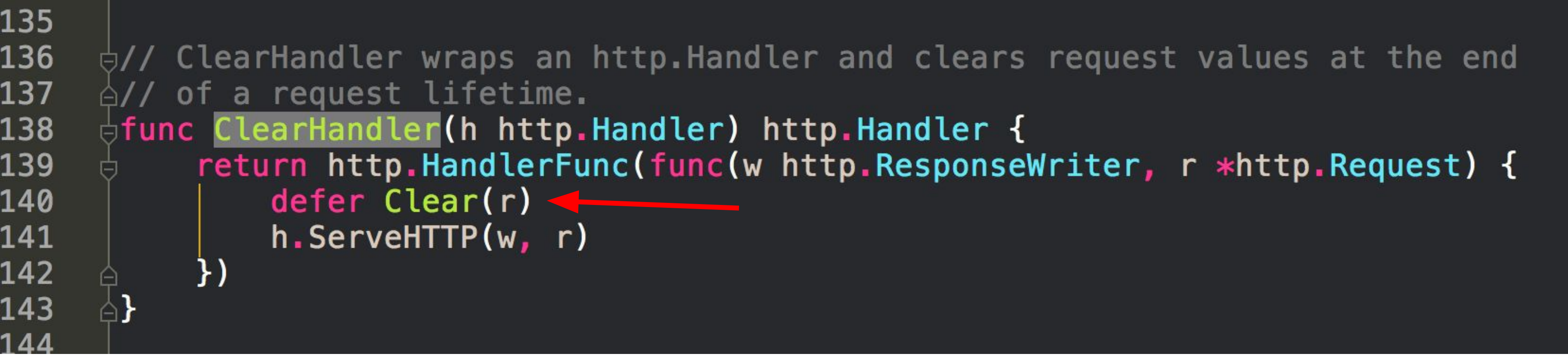
LOG OUT

ADD PHOTO

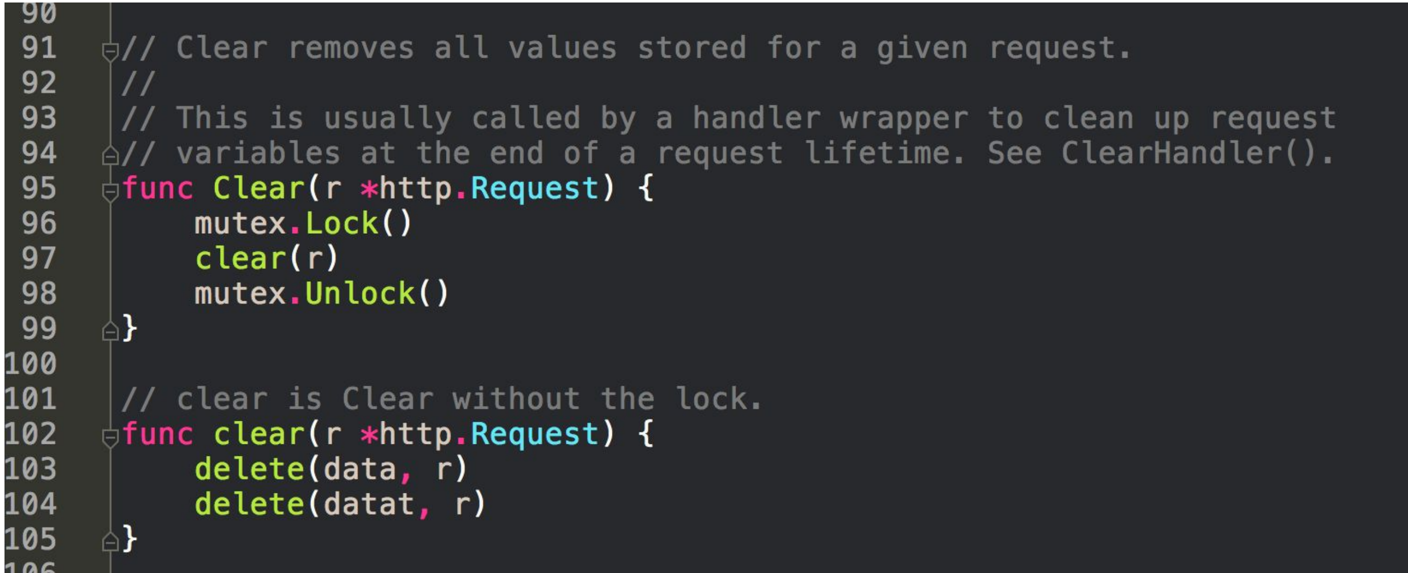
Choose File No file chosen Submit



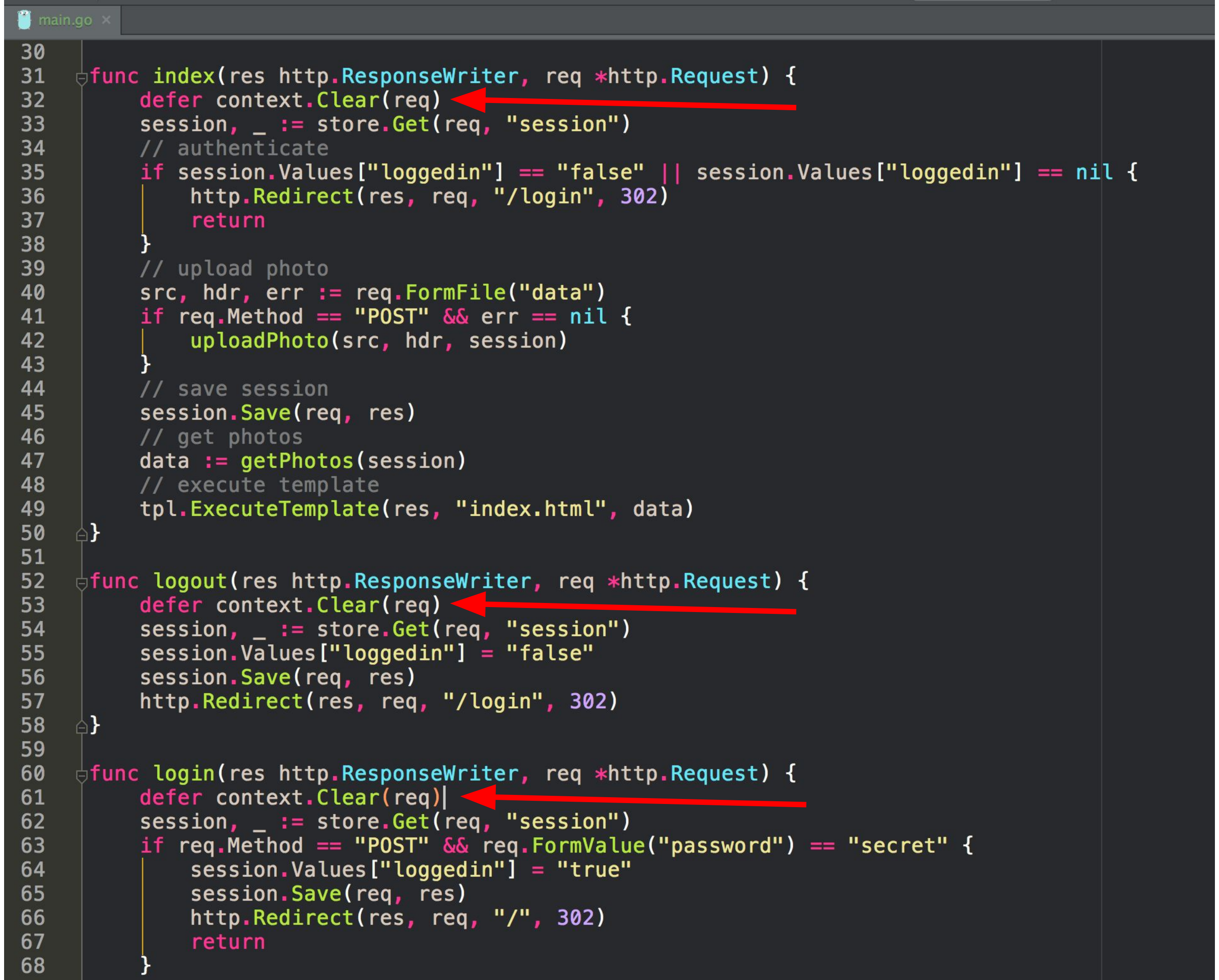
```go

// ClearHandler wraps an http.Handler and clears request values at the end
// of a request lifetime.
func ClearHandler(h http.Handler) http.Handler {
	return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
		defer Clear(r)
		h.ServeHTTP(w, r)
	})
}

```



```go

// ClearHandler wraps an http.Handler and clears request values at the end
// of a request lifetime.
func ClearHandler(h http.Handler) http.Handler {
	return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
		defer Clear(r)
		h.ServeHTTP(w, r)
	})
}

```

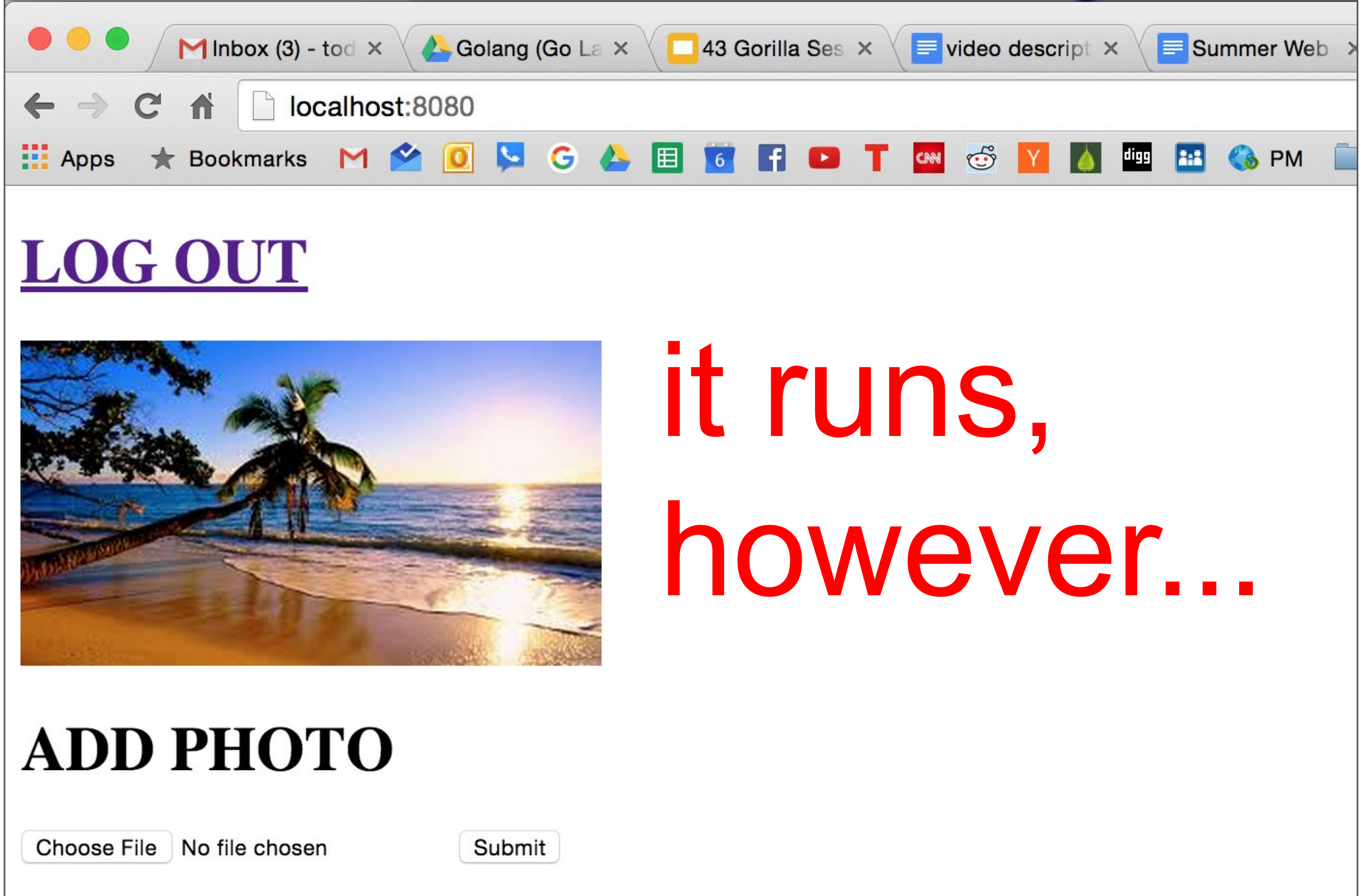


```go

// Clear removes all values stored for a given request.
//
// This is usually called by a handler wrapper to clean up request
// variables at the end of a request lifetime. See ClearHandler().
func Clear(r *http.Request) {
	mutex.Lock()
	clear(r)
	mutex.Unlock()
}

// clear is Clear without the lock.
func clear(r *http.Request) {
	delete(data, r)
	delete(datat, r)
}
```



main.go

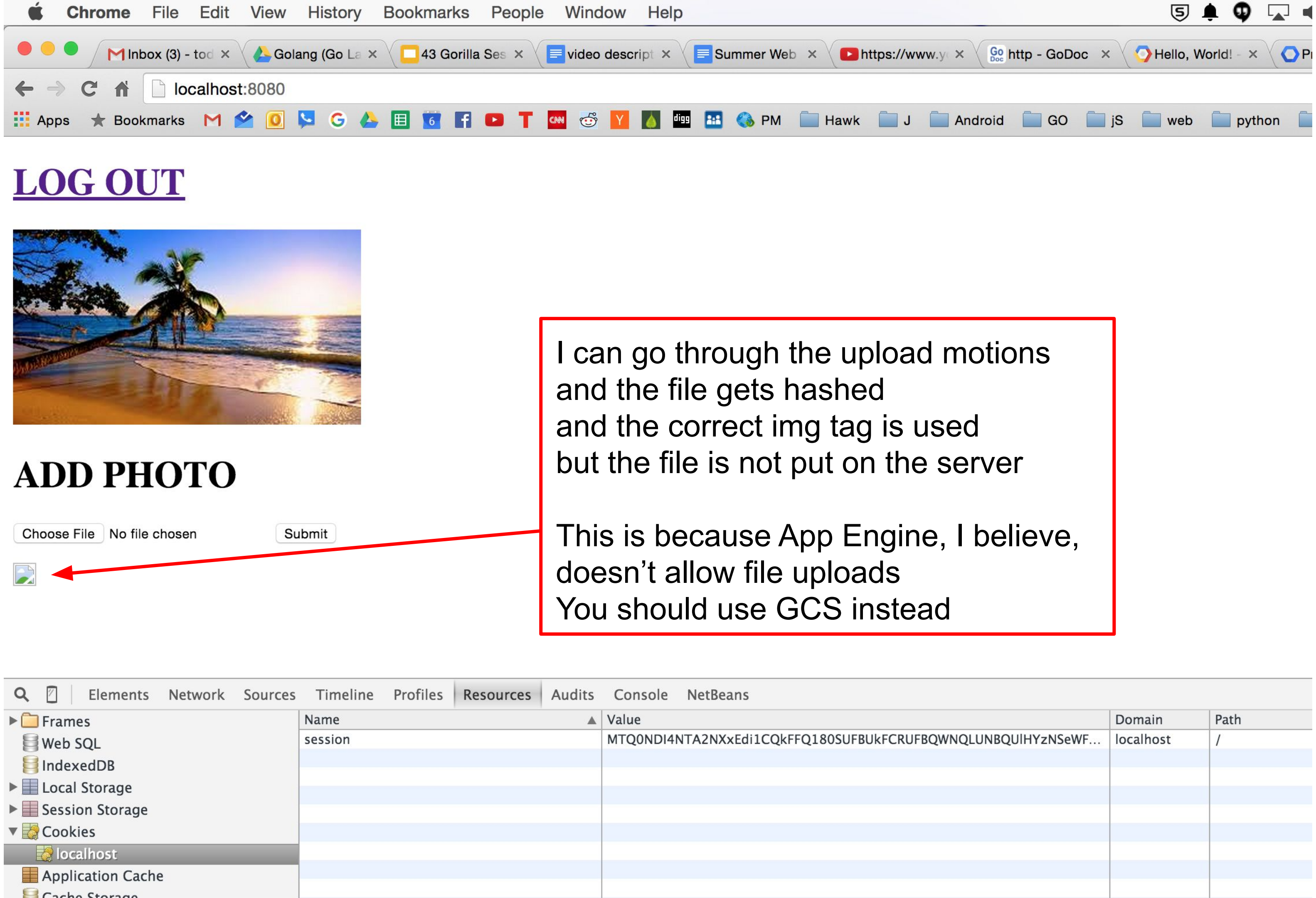
```go

func index(res http.ResponseWriter, req *http.Request) {
	defer context.Clear(req)
	session, _ := store.Get(req, "session")
	// authenticate
	if session.Values["loggedin"] == "false" || session.Values["loggedin"] == nil {
		http.Redirect(res, req, "/login", 302)
		return
	}
	// upload photo
	src, hdr, err := req.FormFile("data")
	if req.Method == "POST" && err == nil {
		uploadPhoto(src, hdr, session)
	}
	// save session
	session.Save(req, res)
	// get photos
	data := getPhotos(session)
	// execute template
	tpl.ExecuteTemplate(res, "index.html", data)
}

func logout(res http.ResponseWriter, req *http.Request) {
	defer context.Clear(req)
	session, _ := store.Get(req, "session")
	session.Values["loggedin"] = "false"
	session.Save(req, res)
	http.Redirect(res, req, "/login", 302)
}

func login(res http.ResponseWriter, req *http.Request) {
	defer context.Clear(req)
	session, _ := store.Get(req, "session")
	if req.Method == "POST" && req.FormValue("password") == "secret" {
		session.Values["loggedin"] = "true"
		session.Save(req, res)
		http.Redirect(res, req, "/", 302)
		return
	}
```



LOG OUT

ADD PHOTO

it runs, however...



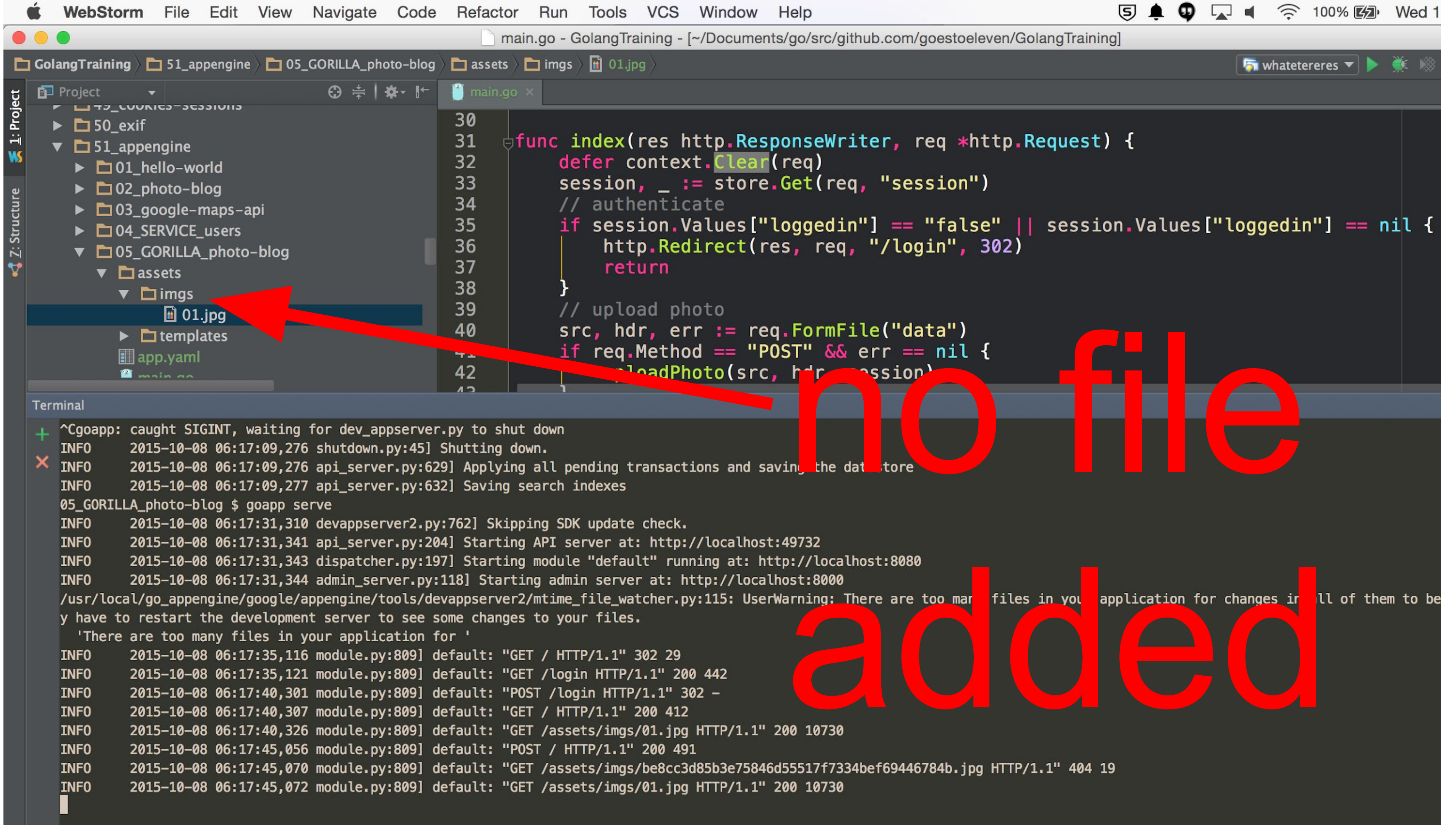
LOG OUT

ADD PHOTO

Name	Value	Domain	Path
session	MTQONDI4NTA2NXxEdi1CQkFFQ180SUFBUkFCRUFBQWNQLUNBQUlHYzNSeWF...	localhost	/

I can go through the upload motions
and the file gets hashed
and the correct img tag is used
but the file is not put on the server

This is because App Engine, I believe,
doesn't allow file uploads
You should use GCS instead


main.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining › 51_appengine › 05_GORILLA_photo-blog › assets › imgs › 01.jpg
50_exif
51_appengine
01_hello-world
02_photo-blog
03_google-maps-api
04_SERVICE_users
05_GORILLA_photo-blog
assets
imgs
01.jpg
templates
app.yaml
main.go
Terminal
no file added

```go
func index(res http.ResponseWriter, req *http.Request) {
	defer context.Clear(req)
	session, _ := store.Get(req, "session")
	// authenticate
	if session.Values["loggedin"] == "false" || session.Values["loggedin"] == nil {
		http.Redirect(res, req, "/login", 302)
		return
	}
	// upload photo
	src, hdr, err := req.FormFile("data")
	if req.Method == "POST" && err == nil {
		uploadPhoto(src, hdr, session)
```

```
^Cgoapp: caught SIGINT, waiting for dev_appserver.py to shut down
INFO     2015-10-08 06:17:09,276 shutdown.py:45] Shutting down.
INFO     2015-10-08 06:17:09,276 api_server.py:629] Applying all pending transactions and saving the datastore
INFO     2015-10-08 06:17:09,277 api_server.py:632] Saving search indexes
05_GORILLA_photo-blog $ goapp serve
INFO     2015-10-08 06:17:31,310 devappserver2.py:762] Skipping SDK update check.
INFO     2015-10-08 06:17:31,341 api_server.py:204] Starting API server at: http://localhost:49732
INFO     2015-10-08 06:17:31,343 dispatcher.py:197] Starting module "default" running at: http://localhost:8080
INFO     2015-10-08 06:17:31,344 admin_server.py:118] Starting admin server at: http://localhost:8000
/usr/local/go_appengine/google/appengine/tools/devappserver2/mtime_file_watcher.py:115: UserWarning: There are too many files in your application for changes in all of them to be
y have to restart the development server to see some changes to your files.
  'There are too many files in your application for '
INFO     2015-10-08 06:17:35,116 module.py:809] default: "GET / HTTP/1.1" 302 29
INFO     2015-10-08 06:17:35,121 module.py:809] default: "GET /login HTTP/1.1" 200 442
INFO     2015-10-08 06:17:40,301 module.py:809] default: "POST /login HTTP/1.1" 302 -
INFO     2015-10-08 06:17:40,307 module.py:809] default: "GET / HTTP/1.1" 200 412
INFO     2015-10-08 06:17:40,326 module.py:809] default: "GET /assets/imgs/01.jpg HTTP/1.1" 200 10730
INFO     2015-10-08 06:17:45,056 module.py:809] default: "POST / HTTP/1.1" 200 491
INFO     2015-10-08 06:17:45,070 module.py:809] default: "GET /assets/imgs/be8cc3d85b3e75846d55517f7334bef69446784b.jpg HTTP/1.1" 404 19
INFO     2015-10-08 06:17:45,072 module.py:809] default: "GET /assets/imgs/01.jpg HTTP/1.1" 200 10730
```