GCS

Google Cloud Storage

Which way?

- P12 / PEM
  - no
  - https://cloud.google.com/appengine/docs/go/googlecloudstorageclient/getstarted
- gcloud - aedeploy
  - yes - if managed VM
  - https://cloud.google.com/
  - https://cloud.google.com/docs/
  - https://cloud.google.com/go/
  - https://cloud.google.com/go/getting-started/using-cloud-storage
- JSON
  - yes - if app engine
  - https://godoc.org/google.golang.org/cloud/storage
  - https://godoc.org/golang.org/x/oauth2/google

GCS

Google Cloud Storage

https://cloud.google.com/appengine/docs/go/googlecloudstorageclient/

- Buckets, objects, and ACLs
- Modifying Cloud Storage objects
- Cloud Storage and subdirectories
- Alternative methods for accessing Cloud Storage

https://cloud.google.com/appengine/docs/go/googlecloudstorageclient/activate

---

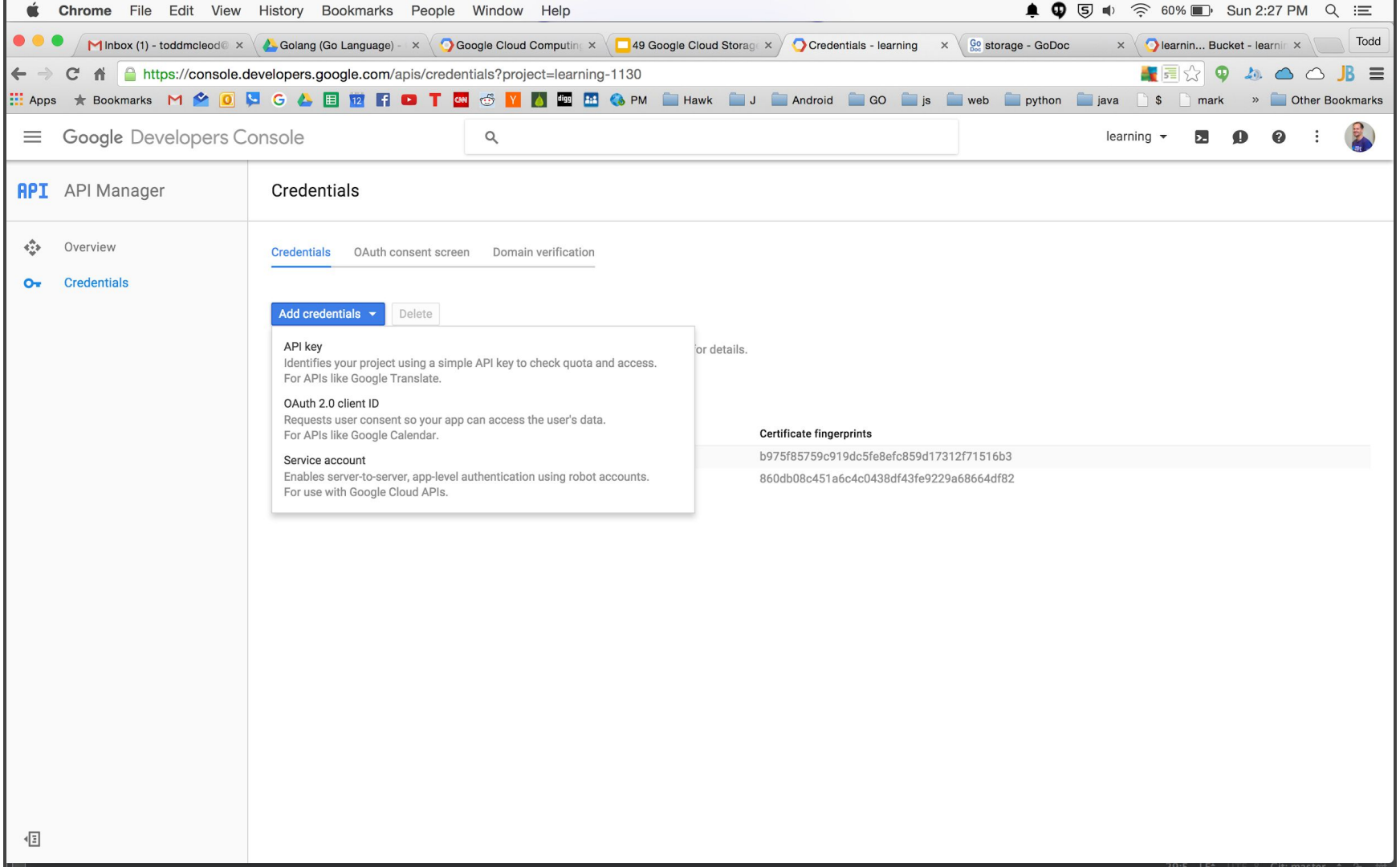
Create a bucket

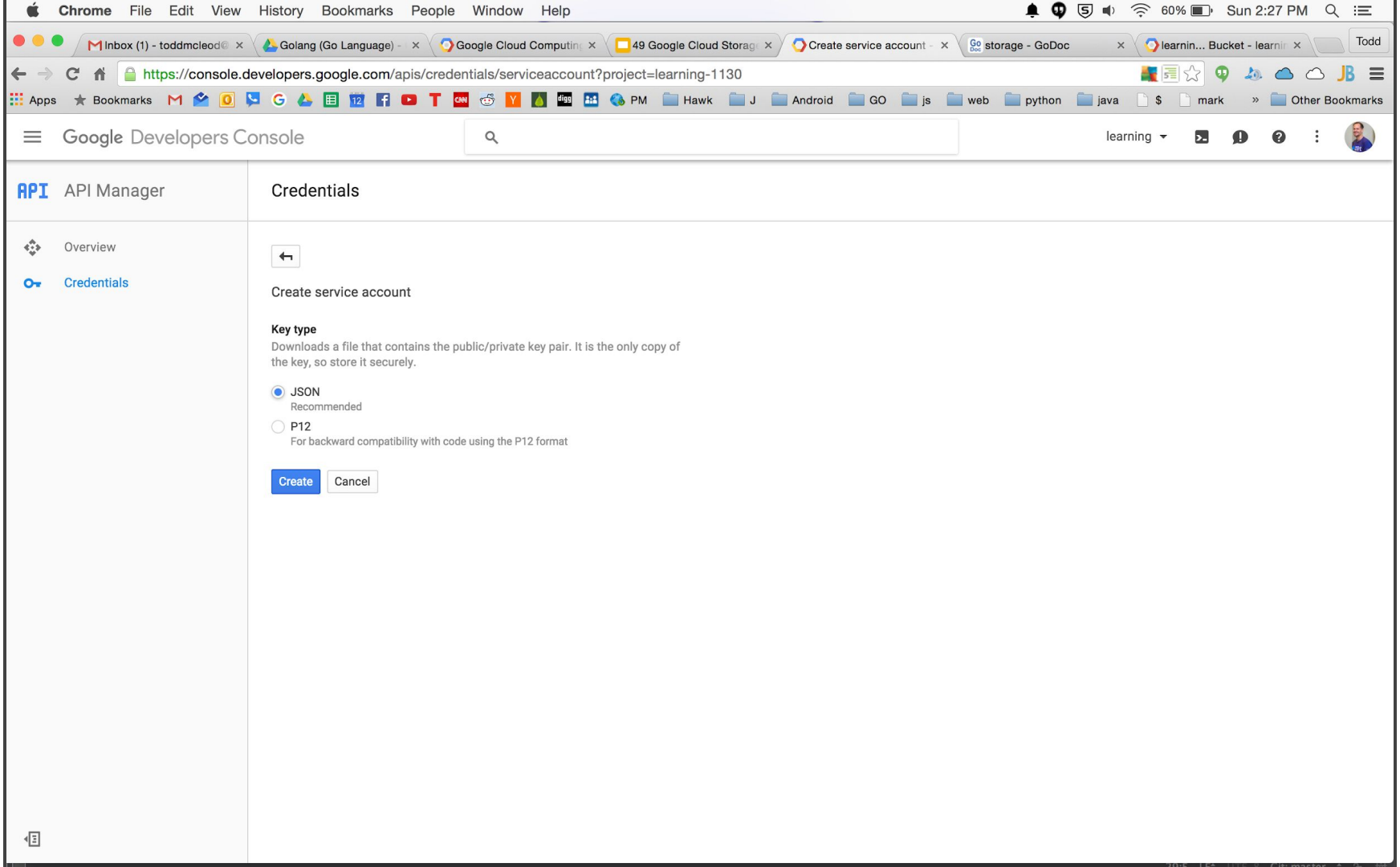
https://cloud.google.com/appengine/docs/go/googlecloudstorageclient/download

---

```
go get -u google.golang.org/cloud/storage
```

setting up dev authorization


Chrome File Edit View History Bookmarks People Window Help
60% Sun 2:27 PM
Inbox (1) - toddmcleod@
Golang (Go Language) -
Google Cloud Computin
49 Google Cloud Storag
Credentials - learning
storage - GoDoc
learnin... Bucket - learnir
Todd
https://console.developers.google.com/apis/credentials?project=learning-1130
Apps Bookmarks PM Hawk J Android GO js web python java $ mark Other Bookmarks
Google Developers Console
learning
API API Manager
Overview
Credentials
Credentials
Credentials OAuth consent screen Domain verification
Add credentials Delete
API key
Identifies your project using a simple API key to check quota and access.
For APIs like Google Translate.
OAuth 2.0 client ID
Requests user consent so your app can access the user's data.
For APIs like Google Calendar.
Service account
Enables server-to-server, app-level authentication using robot accounts.
For use with Google Cloud APIs.
or details.
Certificate fingerprints
b975f85759c919dc5fe8efc859d17312f71516b3
860db08c451a6c4c0438df43fe9229a68664df82


Chrome File Edit View History Bookmarks People Window Help
60% Sun 2:27 PM
Inbox (1) - toddmcleod@
Golang (Go Language) -
Google Cloud Computin
49 Google Cloud Storag
Create service account -
storage - GoDoc
learnin... Bucket - learnir
Todd
https://console.developers.google.com/apis/credentials/serviceaccount?project=learning-1130
Apps Bookmarks PM Hawk J Android GO js web python java $ mark Other Bookmarks
Google Developers Console
learning
API API Manager
Overview
Credentials
Credentials
Create service account
Key type
Downloads a file that contains the public/private key pair. It is the only copy of the key, so store it securely.
JSON
Recommended
P12
For backward compatibility with code using the P12 format
Create
Cancel

P12 → PEM AUTH

(old way)

code

---

write code for your app to work with GCS

storage.go

```go
import (
    "io"
    "net/http"

    "golang.org/x/oauth2"
    "golang.org/x/oauth2/google"

    "google.golang.org/appengine"
    "google.golang.org/appengine/urlfetch"
    "google.golang.org/cloud"
    "google.golang.org/cloud/storage"
)

func init() {
    http.HandleFunc("/put", handlePut)

}

func handlePut(res http.ResponseWriter, req *http.Request) {
    ctx := appengine.NewContext(req)
    bucket := "learning-1130-bucket-01"

    hc := &http.Client{
        Transport: &oauth2.Transport{
            Source: google.AppEngineTokenSource(ctx, storage.ScopeFullControl),
            Base:   &urlfetch.Transport{Context: ctx},
        },
    }

    cctx := cloud.NewContext(appengine.AppID(ctx), hc)

    writer := storage.NewWriter(cctx, bucket, "example.txt")
    io.WriteString(writer, "Hello World!!!!!")
    err := writer.Close()
    if err != nil {
        http.Error(res, "ERROR WRITING TO BUCKET: "+err.Error(), 500)
        return
    }
}
```

https://cloud.google.com/appengine/docs/go/googlecloudstorageclient/getstarted

---

download p12 for service account

convert to pem

terminal this:

```
/path/to/AppEngSDK/dev_appserver.py . --appidentity_email_address
<your_app_email_address>@developer.gserviceaccount.com --appidentity_private_key_path pem_file.pem
```

JSON AUTH

(new way)

package storage

```
import "google.golang.org/cloud/storage"
```

Package storage contains a Google Cloud Storage client.

This package is experimental and may make backwards-incompatible changes.

Example (Auth)

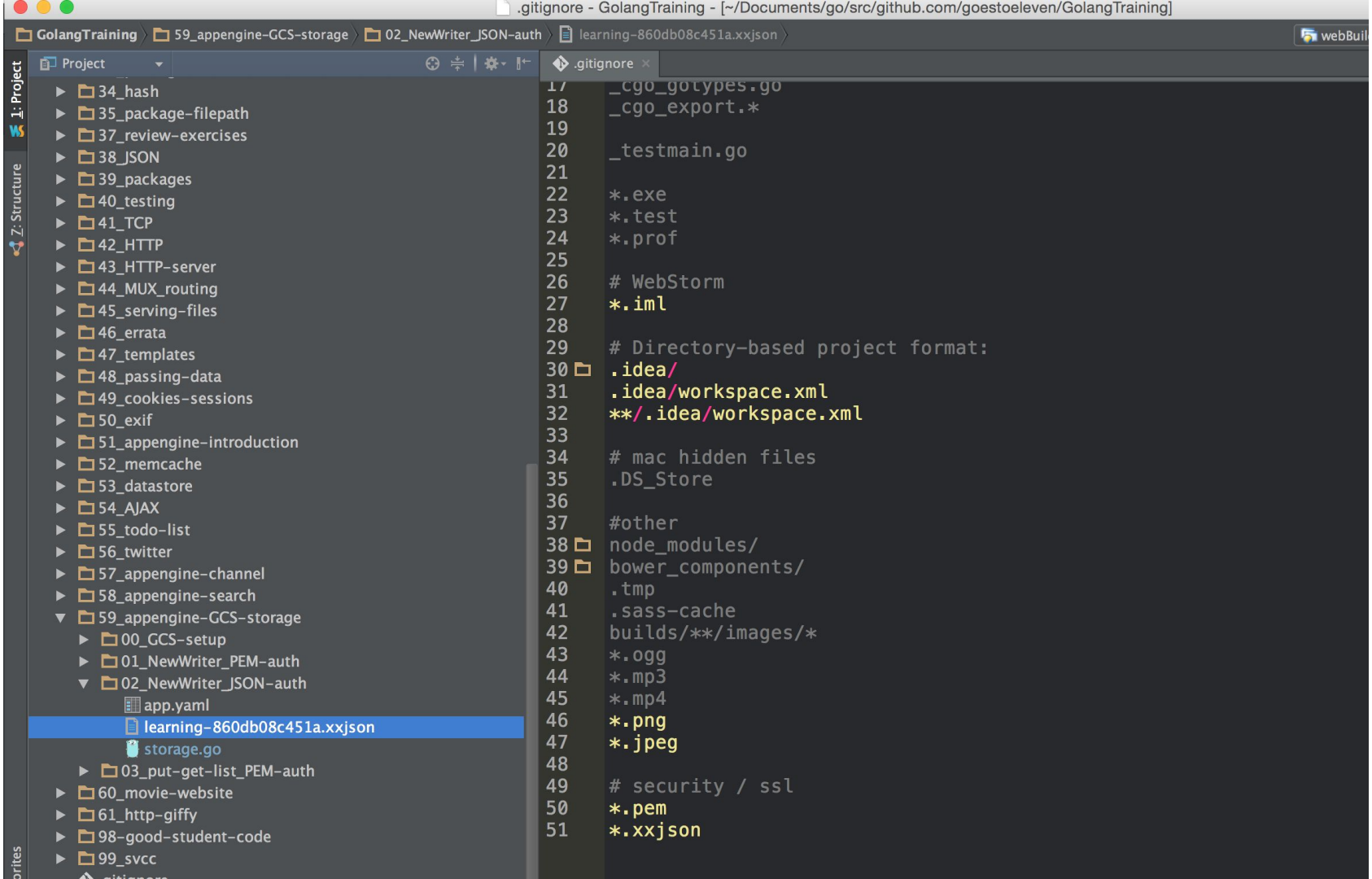
Code: play

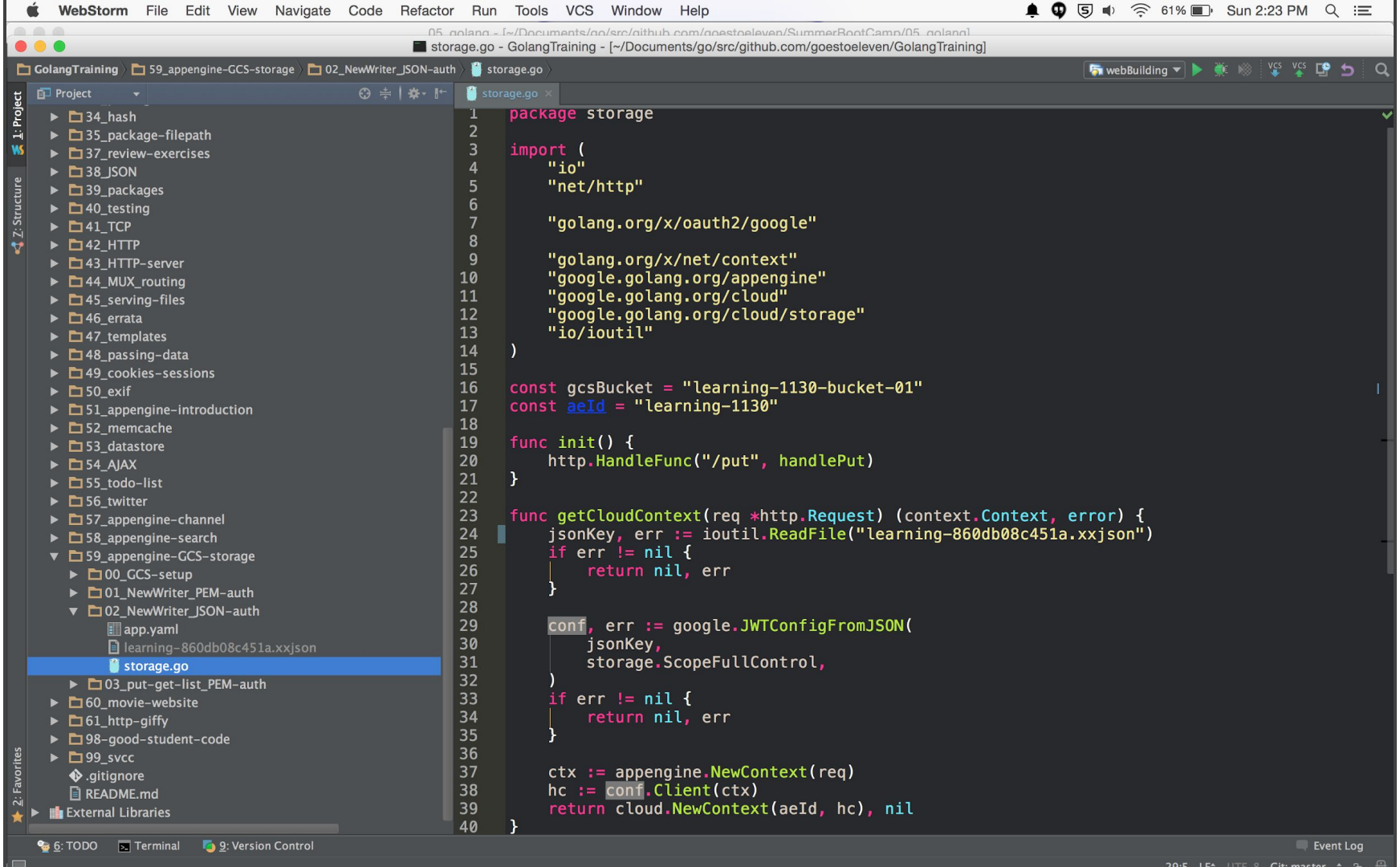
```
// Initialize an authorized context with Google Developers Console
// JSON key. Read the google package examples to learn more about
// different authorization flows you can use.
// http://godoc.org/golang.org/x/oauth2/google
jsonKey, err := ioutil.ReadFile("/path/to/json/keyfile.json")
if err != nil {
    log.Fatal(err)
}
conf, err := google.JWTConfigFromJSON(
    jsonKey,
    storage.ScopeFullControl,
)
if err != nil {
    log.Fatal(err)
}
ctx := cloud.NewContext("project-id", conf.Client(oauth2.NoContext))
// Use the context (see other examples)
return ctx
```

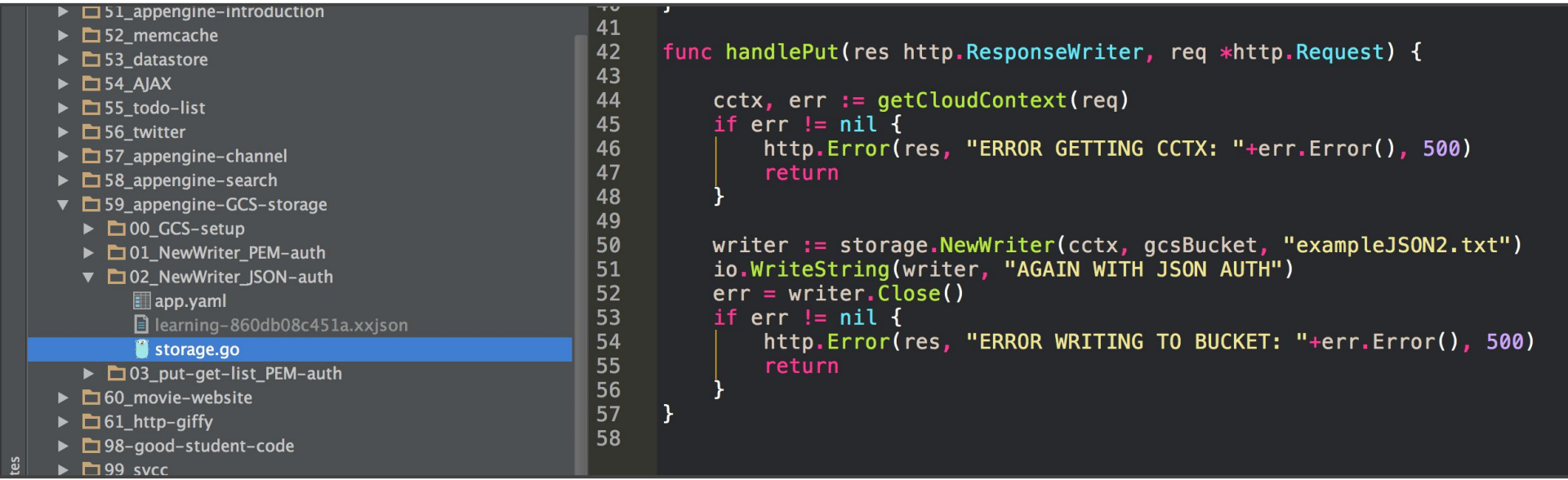
code

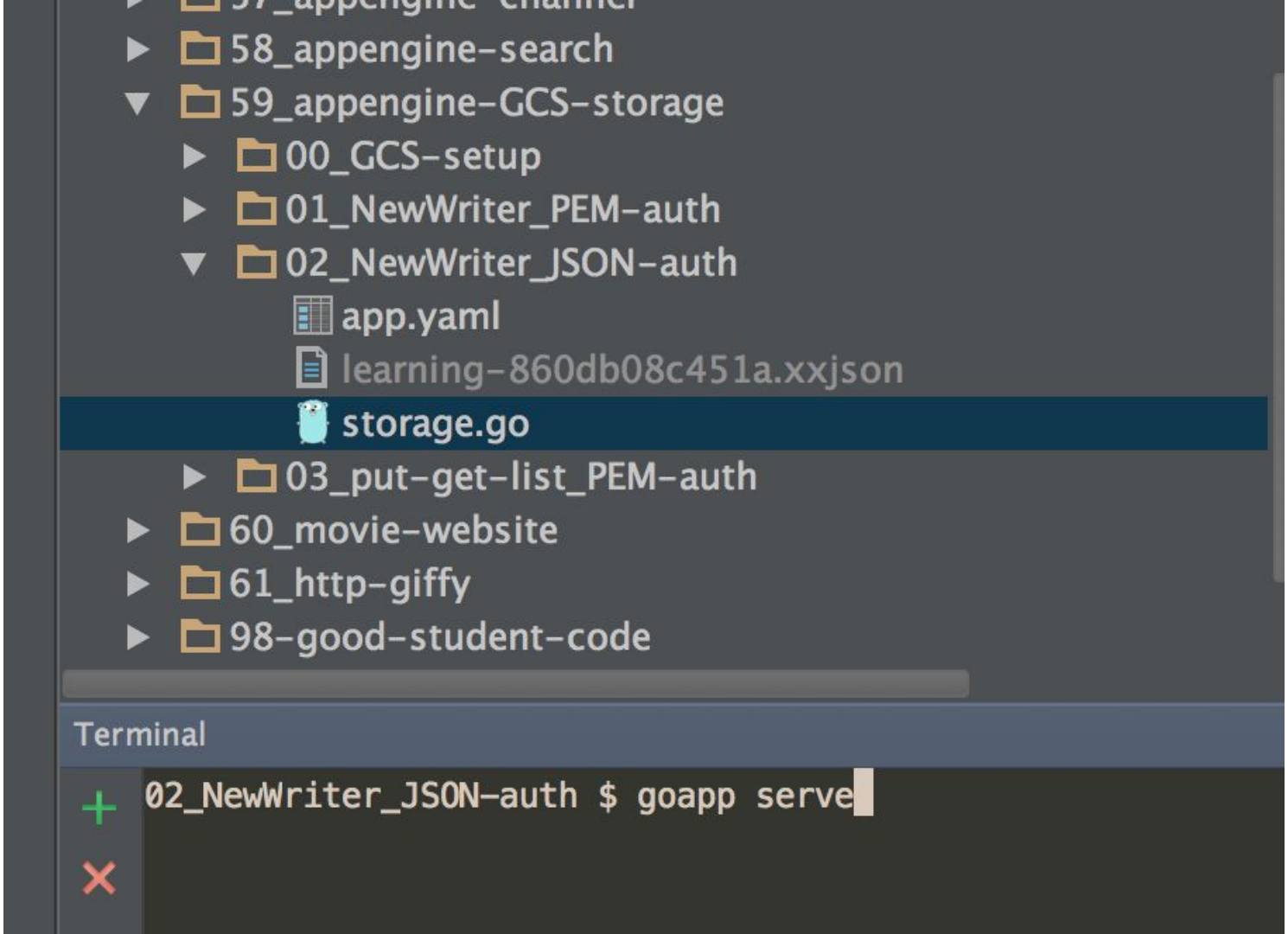
---

write code for your app to work with GCS


.gitignore - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining
59_appengine-GCS-storage
02_NewWriter_JSON-auth
learning-860db08c451a.xxjson
webBuil
1: Project
Z: Structure
Project
34_hash
35_package-filepath
37_review-exercises
38_JSON
39_packages
40_testing
41_TCP
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
53_datastore
54_AJAX
55_todo-list
56_twitter
57_appengine-channel
58_appengine-search
59_appengine-GCS-storage
00_GCS-setup
01_NewWriter_PEM-auth
02_NewWriter_JSON-auth
app.yaml
learning-860db08c451a.xxjson
storage.go
03_put-get-list_PEM-auth
60_movie-website
61_http-giffy
98-good-student-code
99_svcc
.gitignore
_cgo_gotypes.go
_cgo_export.*
_testmain.go
*.exe
*.test
*.prof
WebStorm
*.iml
Directory-based project format:
.idea/
.idea/workspace.xml
**/.idea/workspace.xml
mac hidden files
.DS_Store
#other
node_modules/
bower_components/
.tmp
.sass-cache
builds/**/images/*
*.ogg
*.mp3
*.mp4
*.png
*.jpeg
security / ssl
*.pem
*.xxjson


WebStorm File Edit View Navigate Code Refactor Run Tools VCS Window Help
61% Sun 2:23 PM
storage.go - GolangTraining - [~/Documents/go/src/github.com/goestoeleven/GolangTraining]
GolangTraining 59_appengine-GCS-storage 02_NewWriter_JSON-auth storage.go
webBuilding
Project
34_hash
35_package-filepath
37_review-exercises
38_JSON
39_packages
40_testing
41_TCP
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
53_datastore
54_AJAX
55_todo-list
56_twitter
57_appengine-channel
58_appengine-search
59_appengine-GCS-storage
00_GCS-setup
01_NewWriter_PEM-auth
02_NewWriter_JSON-auth
app.yaml
learning-860db08c451a.xxjson
storage.go
03_put-get-list_PEM-auth
60_movie-website
61_http-giffy
98-good-student-code
99_svcc
.gitignore
README.md
External Libraries
storage.go
package storage
import (
"io"
"net/http"
"golang.org/x/oauth2/google"
"golang.org/x/net/context"
"google.golang.org/appengine"
"google.golang.org/cloud"
"google.golang.org/cloud/storage"
"io/ioutil"
)
const gcsBucket = "learning-1130-bucket-01"
const aeId = "learning-1130"
func init() {
http.HandleFunc("/put", handlePut)
}
func getCloudContext(req *http.Request) (context.Context, error) {
jsonKey, err := ioutil.ReadFile("learning-860db08c451a.xxjson")
if err != nil {
return nil, err
}
conf, err := google.JWTConfigFromJSON(
jsonKey,
storage.ScopeFullControl,
)
if err != nil {
return nil, err
}
ctx := appengine.NewContext(req)
hc := conf.Client(ctx)
return cloud.NewContext(aeId, hc), nil
}
6: TODO
Terminal
9: Version Control
Event Log


51_appengine-introduction
52_memcache
53_datastore
54_AJAX
55_todo-list
56_twitter
57_appengine-channel
58_appengine-search
59_appengine-GCS-storage
00_GCS-setup
01_NewWriter_PEM-auth
02_NewWriter_JSON-auth
app.yaml
learning-860db08c451a.xxjson
storage.go
03_put-get-list_PEM-auth
60_movie-website
61_http-giffy
98-good-student-code
99_sycc
}
func handlePut(res http.ResponseWriter, req *http.Request) {
cctx, err := getCloudContext(req)
if err != nil {
http.Error(res, "ERROR GETTING CCTX: "+err.Error(), 500)
return
}
writer := storage.NewWriter(cctx, gcsBucket, "exampleJSON2.txt")
io.WriteString(writer, "AGAIN WITH JSON AUTH")
err = writer.Close()
if err != nil {
http.Error(res, "ERROR WRITING TO BUCKET: "+err.Error(), 500)
return
}
}


58_appengine-search
59_appengine-GCS-storage
00_GCS-setup
01_NewWriter_PEM-auth
02_NewWriter_JSON-auth
app.yaml
learning-860db08c451a.xxjson
storage.go
03_put-get-list_PEM-auth
60_movie-website
61_http-giffy
98-good-student-code
Terminal
02_NewWriter_JSON-auth $ goapp serve