Basic mac commands

- Screenshot + copy Command + Control + Shift + 4
- Screenshot only command shift 4
- For nano use crtl O and crtl X

Task - User API:

build an rest api using GO, with following apis should be provided:
- to add new user (name, age, id <- unique, just need to be a plain number)
- to get data of a user by providing user's id
- to update user data (name + age) by providing id
- to list out all users registered

-tech stack: Go, Gin (rest framework), for now can just save data on RAM no need database yet (meaning if you restart your api, data lost)

https://go.dev/doc/tutorial/web-service-gin

Instructions:

After installing GO on mac:
In the terminal:

```
pncy1926@V-SPDT-NATHANIELCHIN-MB ~ % nano ~/.zshrc
```

Then paste:

```
export PATH=$PATH:/usr/local/go/bin
export PATH=$PATH:$GOPATH/bin
```

Then source the path file

```
pncy1926@V-SPDT-NATHANIELCHIN-MB ~ % . ~/.zshrc
```

Create a service:

Mkdir goapi
Cd goapi
Go mod init project_name/web_service_name

Import Paths in Go Code:

- When other developers clone your repository, they will import your Go packages using the module path. For example, if your module path is "goapi/web-service," import statements in Go code within your project would look like:

```go
import "goapi/web-service/package-name"
```

- If the module path were simply "web-service," the import path would be:

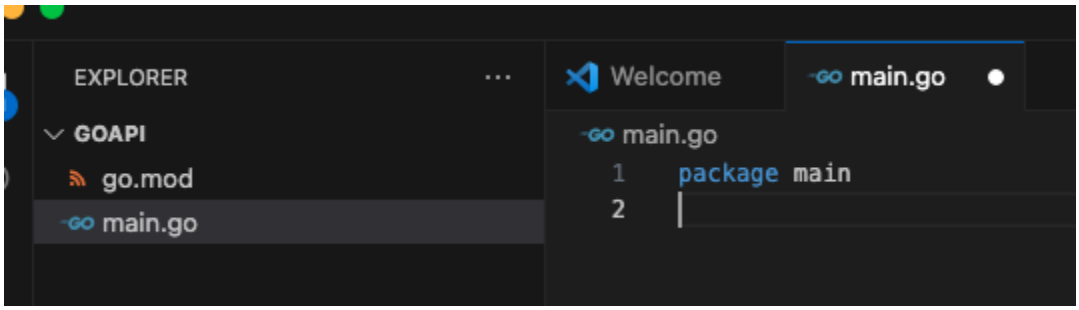
```go
import "web-service/package-name"
```

If you only have a single web service for gin, then most of the time you can initialise directly as a single service without the projectname e.g. above its “goapi”, usually a project name is only used if you have multiple webservices, e.g. web-service-gin-1, web-service-gin-2 etc. So actually can just do **go mod init web-service-gin**

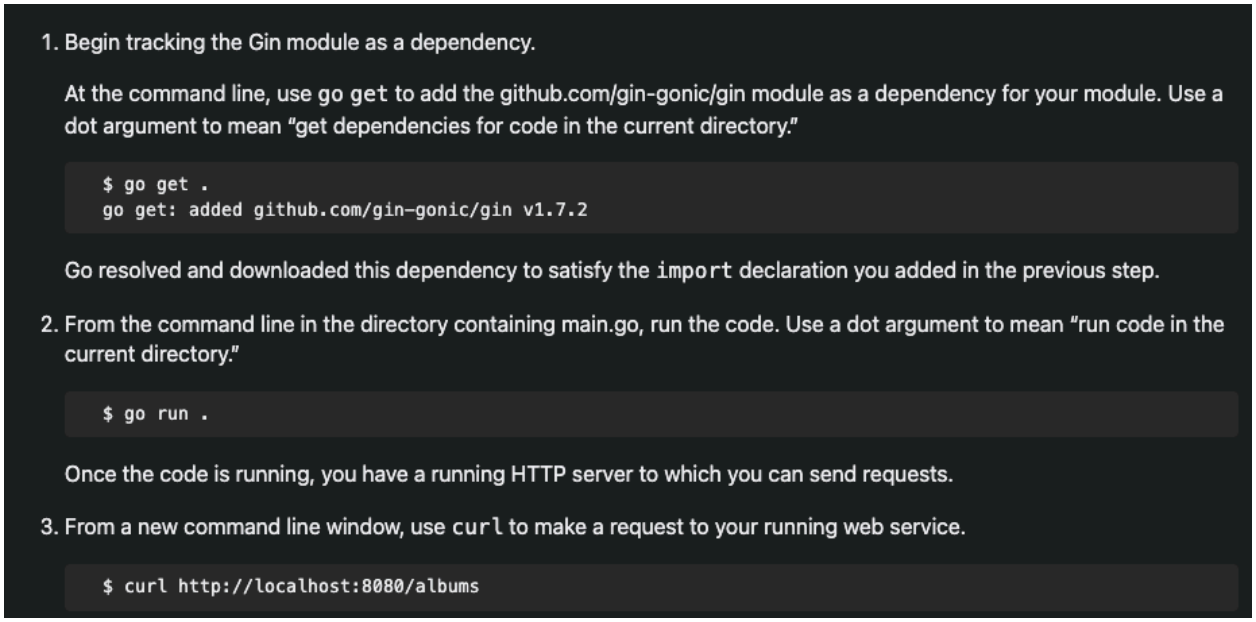
```
[pncy1926@V-SPDT-NATHANIELCHIN-MB goapi % go mod init example/web-service
go: creating new go.mod: module example/web-service
[pncy1926@V-SPDT-NATHANIELCHIN-MB goapi % ls
go.mod
```

This creates the go.mod file that contains a list of all the directories for tracking

Then next in our webservice folder we create a package



Once you finish the code for the api, we need to use go get to install dependencies for our current directory




```
curl http://localhost:8080/users \
    --include \
    --header "Content-Type: application/json" \
    --request "POST" \
    --data '{"id": "4","name": "testname","age": 55}'
```

Task 2: url shortening service

Once postgresql is installed you can launch the ui interface pgadmin

- My superuser password during installation was set to “123”
- I then create a new db called urlDB and set host name to “localhost” and put a password for the DB to 123