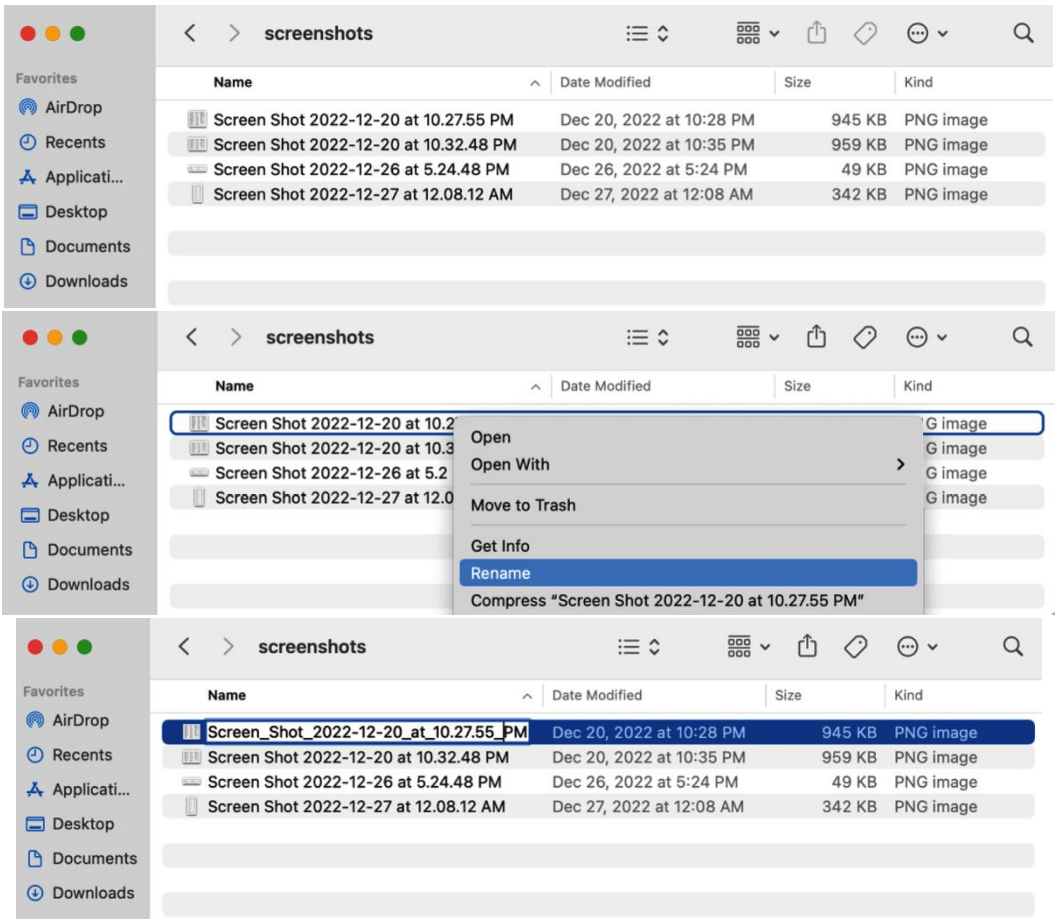
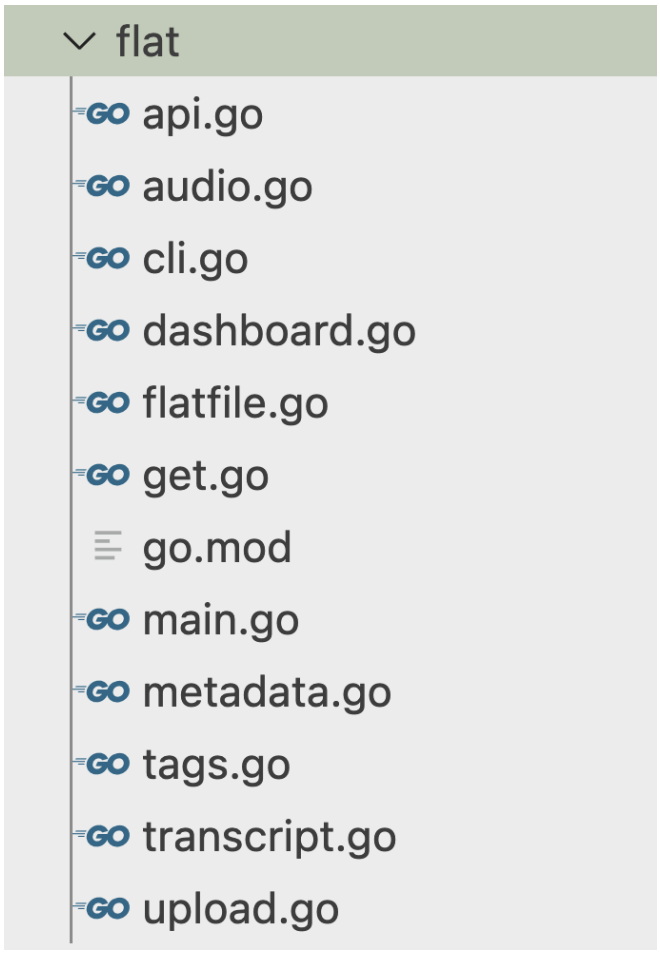
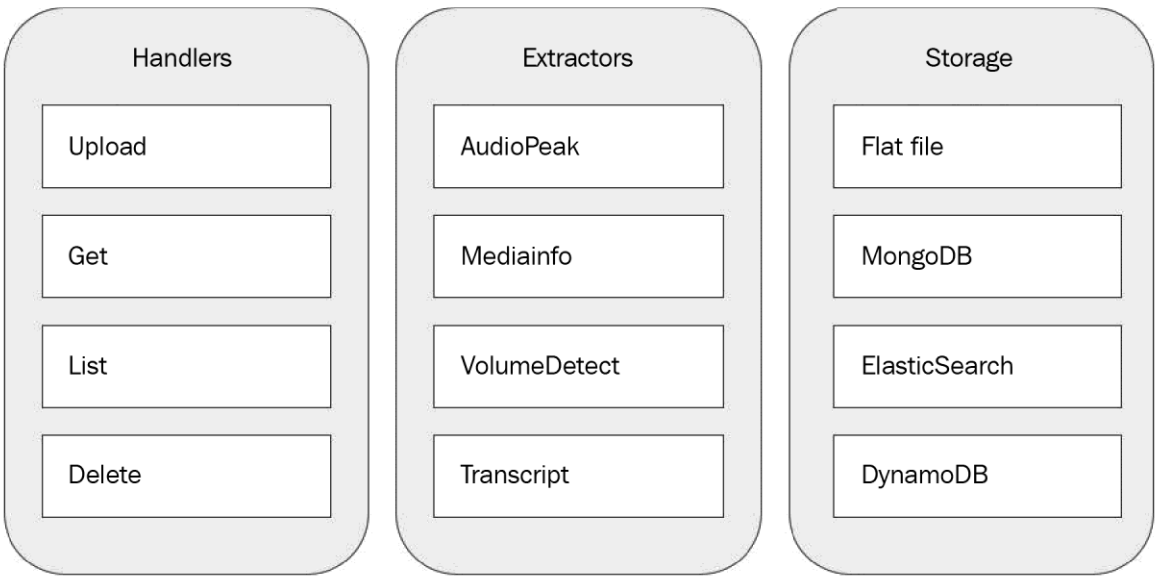
Chapter 1: Understanding CLI Standards

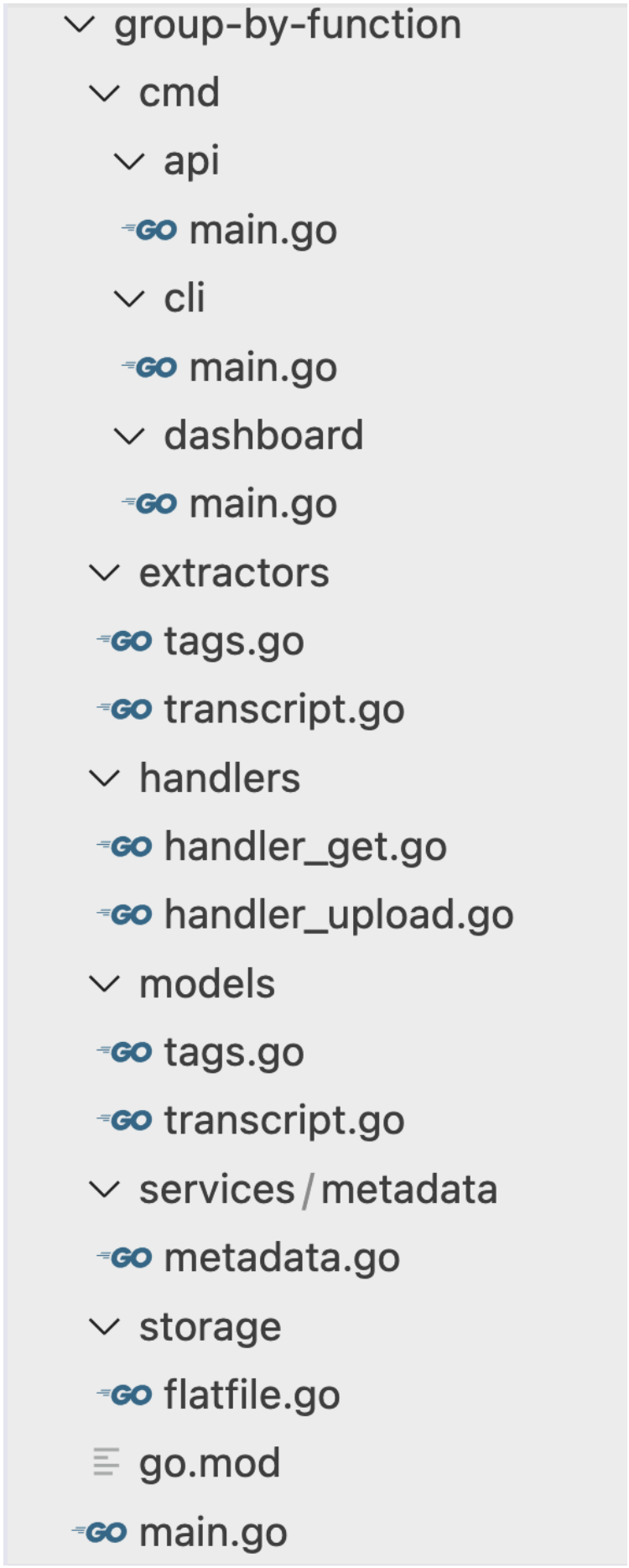


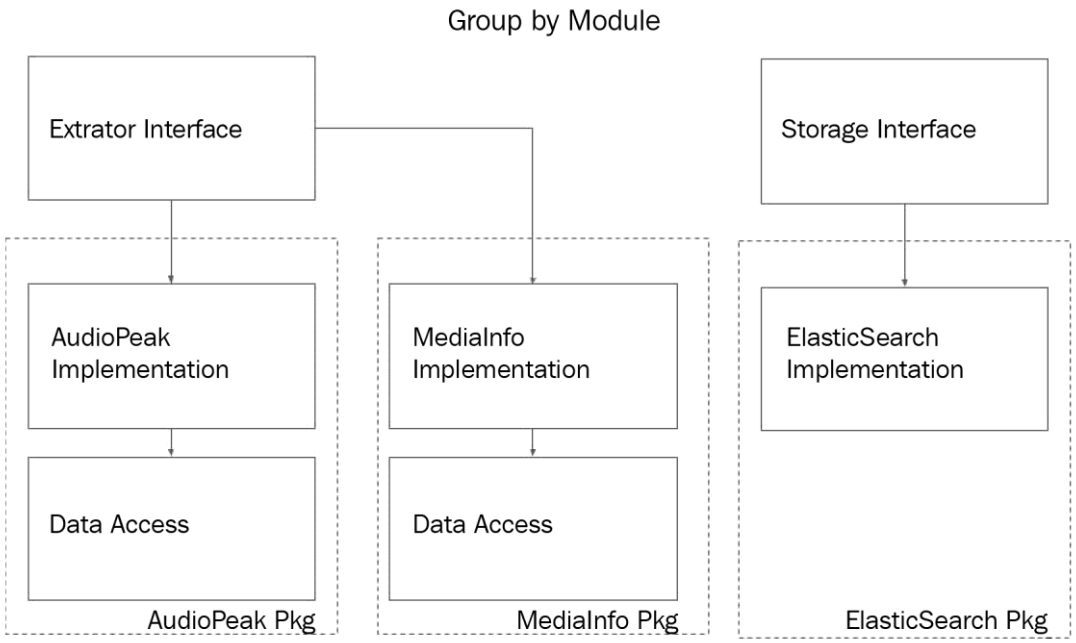
Chapter 2: Structuring Go Code for CLI Applications

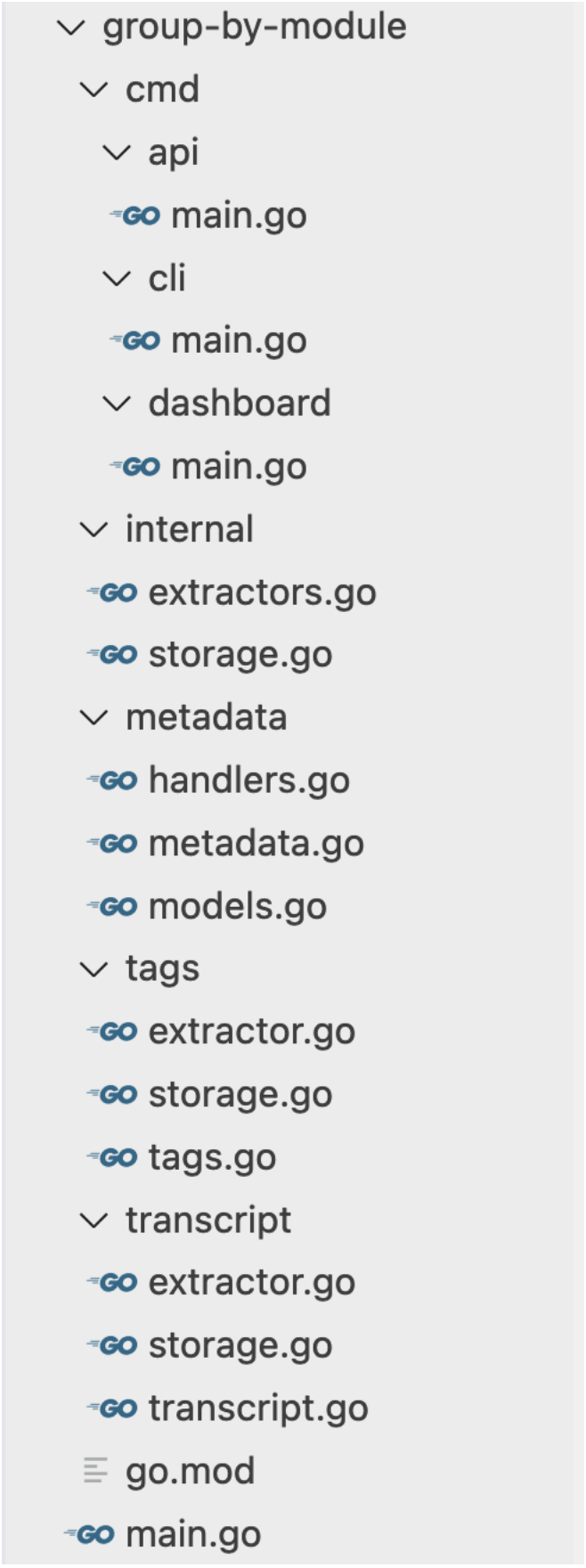


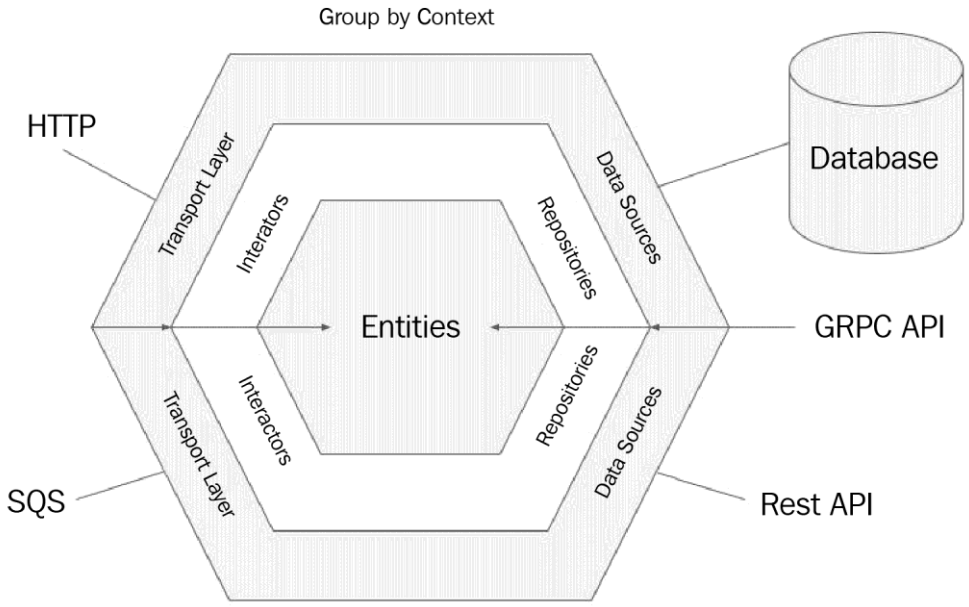
Group by Function

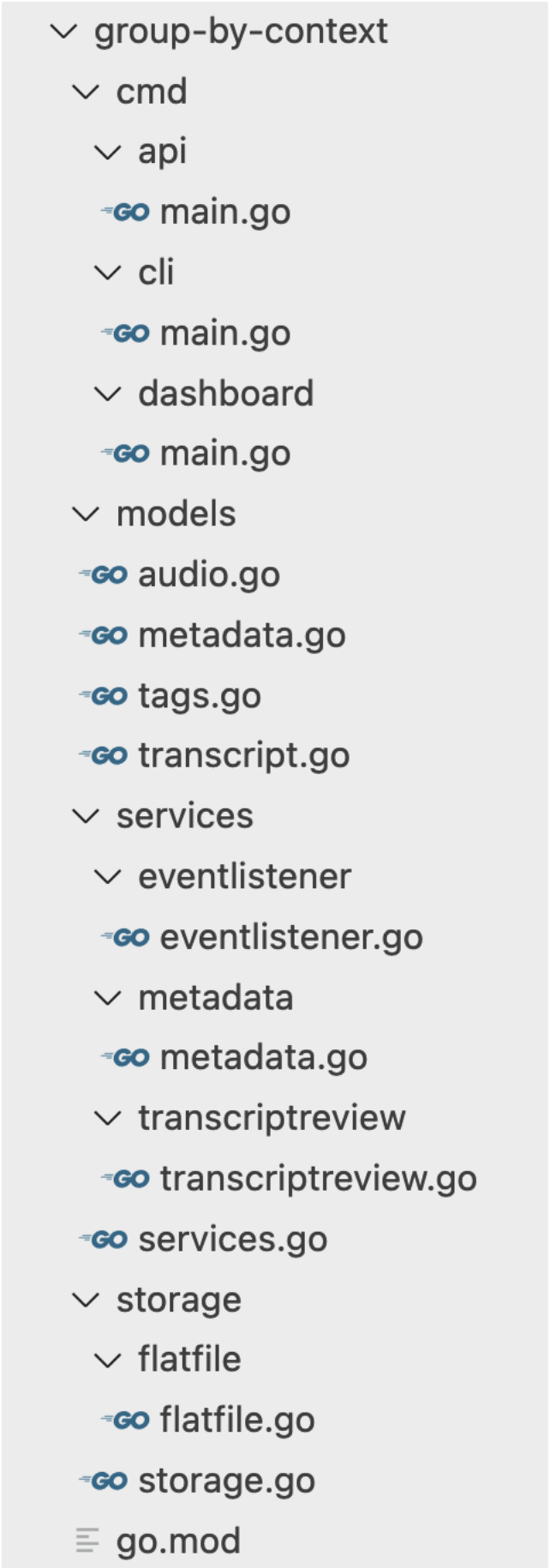


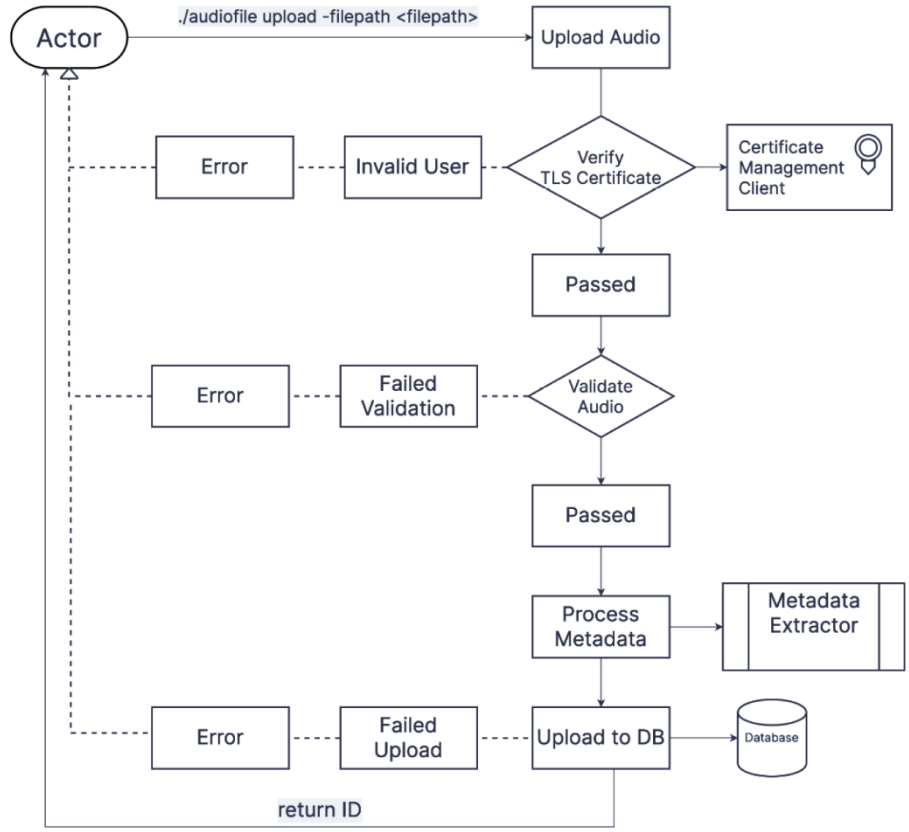

group-by-function
cmd
api
main.go
cli
main.go
dashboard
main.go
extractors
tags.go
transcript.go
handlers
handler_get.go
handler_upload.go
models
tags.go
transcript.go
services / metadata
metadata.go
storage
flatfile.go
go.mod
main.go

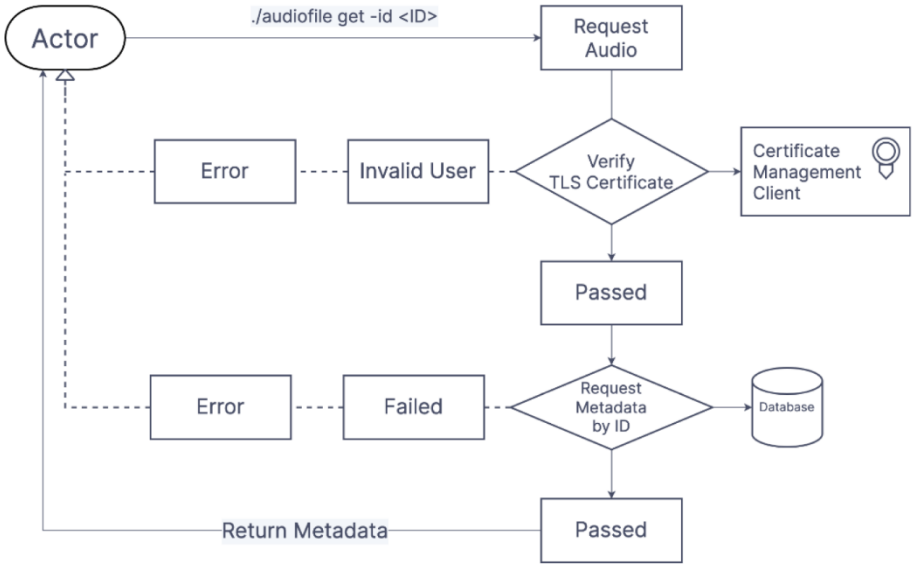

Group by Module
Extrator Interface
Storage Interface
AudioPeak
Implementation
MediaInfo
Implementation
ElasticSearch
Implementation
Data Access
Data Access
AudioPeak Pkg
MediaInfo Pkg
ElasticSearch Pkg

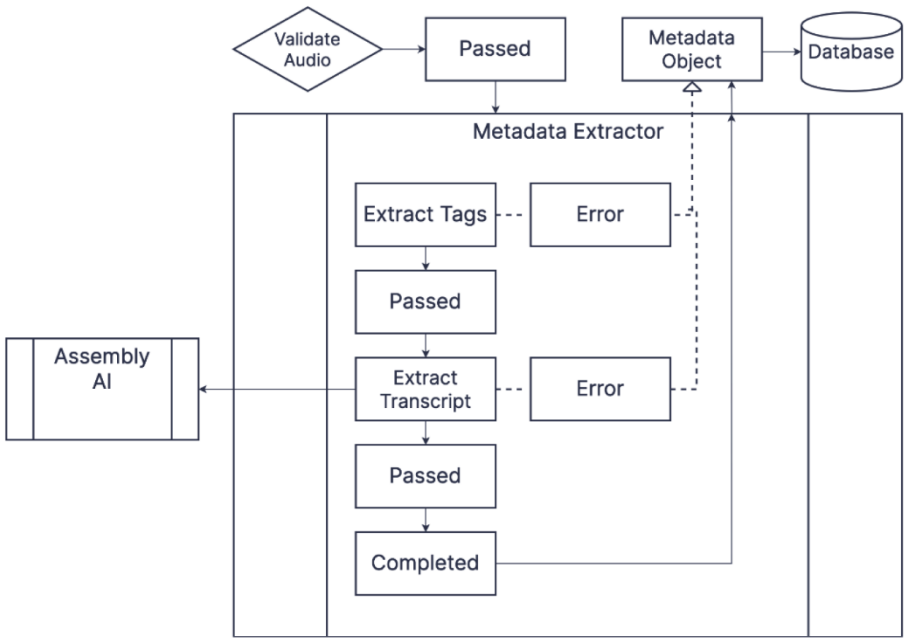

group-by-module
cmd
api
main.go
cli
main.go
dashboard
main.go
internal
extractors.go
storage.go
metadata
handlers.go
metadata.go
models.go
tags
extractor.go
storage.go
tags.go
transcript
extractor.go
storage.go
transcript.go
go.mod
main.go


Group by Context
HTTP
Transport Layer
Interators
Entities
Repositories
Data Sources
Database
GRPC API
SQS
Transport Layer
Interactors
Repositories
Data Sources
Rest API

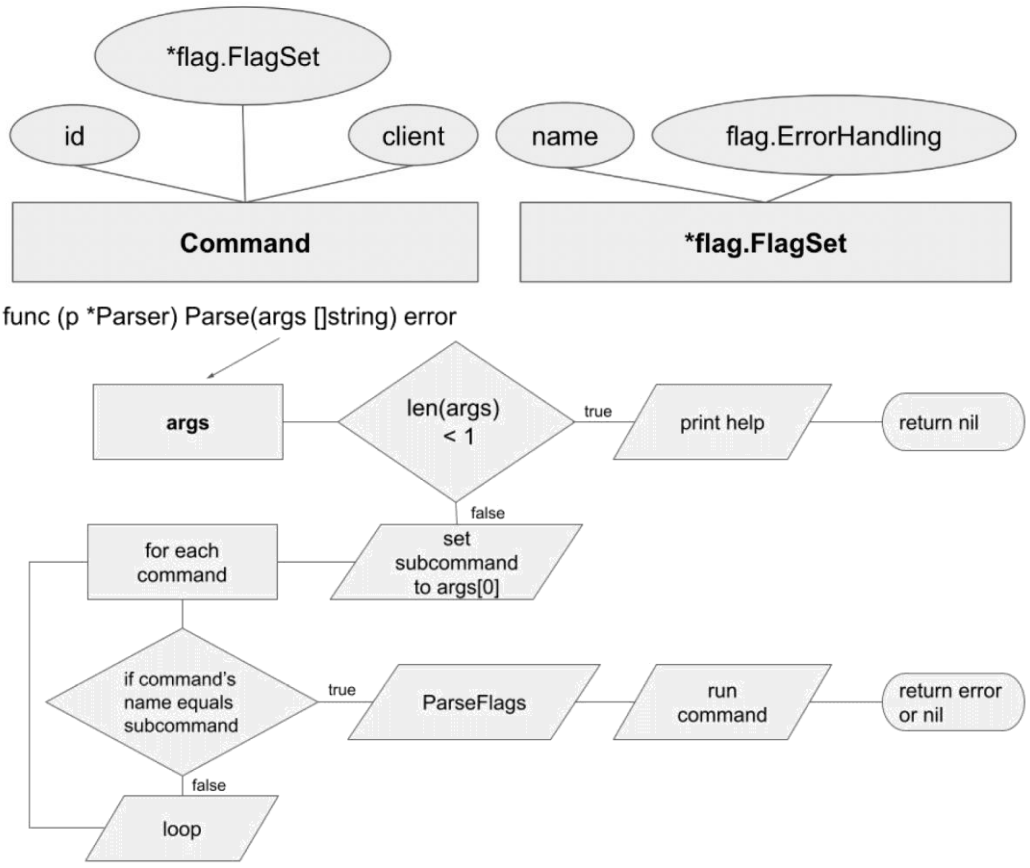

group-by-context
cmd
api
main.go
cli
main.go
dashboard
main.go
models
audio.go
metadata.go
tags.go
transcript.go
services
eventlistener
eventlistener.go
metadata
metadata.go
transcriptreview
transcriptreview.go
services.go
storage
flatfile
flatfile.go
storage.go
go.mod

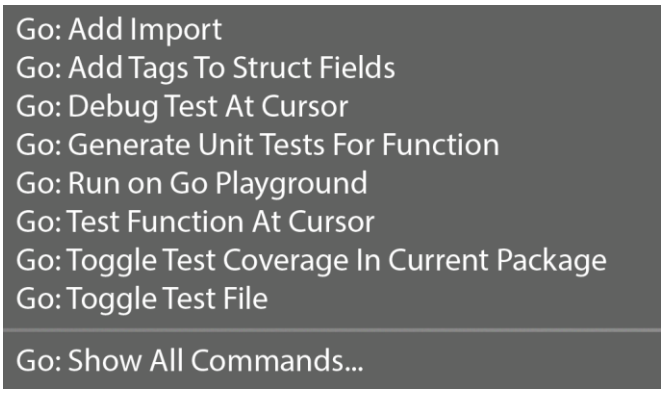

Actor
./audiofile upload -filepath <filepath>
Upload Audio
Verify TLS Certificate
Certificate Management Client
Invalid User
Error
Passed
Validate Audio
Failed Validation
Error
Passed
Process Metadata
Metadata Extractor
Upload to DB
Database
Failed Upload
Error
return ID


Actor
./audiofile get -id <ID>
Request Audio
Verify TLS Certificate
Certificate Management Client
Invalid User
Error
Passed
Request Metadata by ID
Database
Failed
Error
Passed
Return Metadata


Validate Audio
Passed
Metadata Object
Database
Metadata Extractor
Extract Tags
Error
Passed
Assembly AI
Extract Transcript
Error
Passed
Completed

Chapter 3: Design your tool and its purpose





Chapter 4: Popular frameworks in Go to speed up and enhance CLI development

No images…

Chapter 5: Processing Different Types of Input via Arguments and Flags

Signal (Key Combination)	Description
SIGINT (Ctrl-C)	Interrupt application
SIGTSTP (Ctrl-Z)	Suspend application
SIGQUIT (Ctrl-\)	Quit application
SIGHUP	Hang up on controlling terminal or death of controlling process
SIGFPE	Illegal mathematical operation attempted
SIGKILL	Quit immediately with no cleanup process
SIGALRM	Alarm clock signal (for timers)
SIGTERM	Software termination signal (kill by default)

Chapter 6: Calling External Processes and API commands and Handle Timeouts, and Error Handling

No images…

Chapter 7: Developing for Different Platforms

No images…

Chapter 8: Being Clear, Concise and Consistent with CLIs

Input		Output
7	6	&
1	1	1
1	1	1
1	0	0

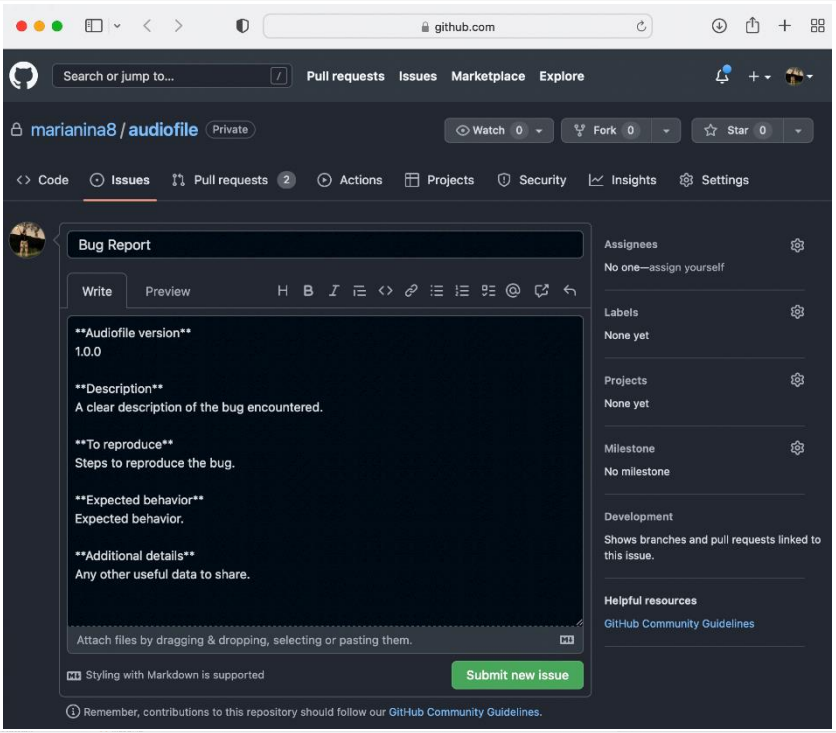
Is a TTY	
Code	**Value**
fileInfo.Mode()	Dcrw--w----
os.ModeCharDevice	c---------
fileInfo.Mode() & os.ModeCharDevice	c---------
(fileInfo.Mode() & os.ModeCharDevice) != 0	TRUE

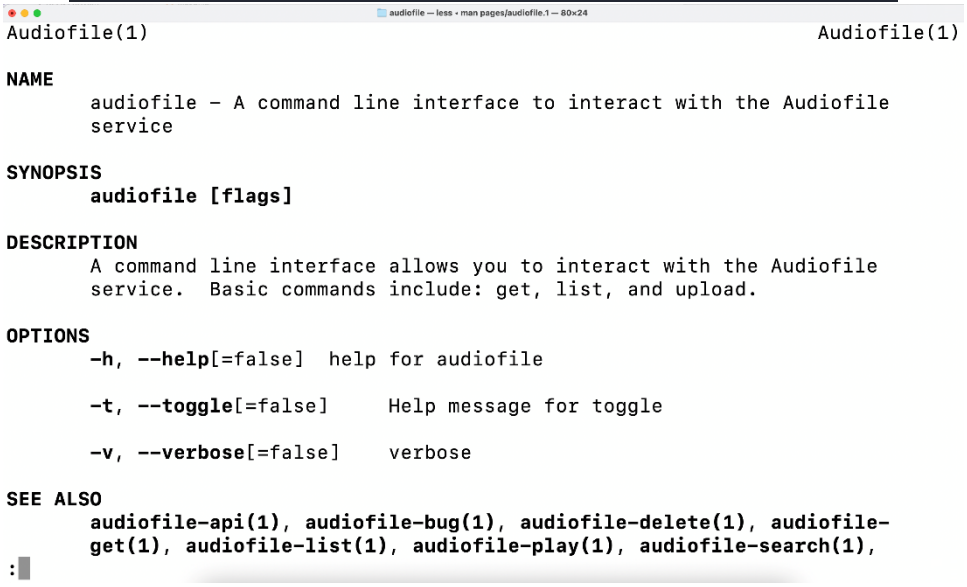
Is not a TTY	
Code	**Value**
fileInfo.Mode()	prw-rw----
os.ModeCharDevice	c---------
fileInfo.Mode() & os.ModeCharDevice	----------
(fileInfo.Mode() & os.ModeCharDevice) != 0	FALSE

CLI Design	
Good	Bad
Checks if outputting to a human versus another machine process	Outputs the same to human versus another machine process
Allows user to change output format	No options to change output format
Allows user to quiet unnecessary output	No silence options
Uses conversational text for humans	Sends conversational text, emojis, and color output to another machine process
Outputs plain text for machine processes	Sends no output at all
Validates input early	No input validation
Transparent with users	Local file modification and network requests without notifying the user
Prioritizes response time over speed	Long response time
Prompts for missing input	No prompts for required input
Confirmation before irreversible deletions	No confirmation before major modifications
Notify user of network requests	Unclear what is happening
Empathic text	Unempathetic text
Use of intentional visual language	Many colors used and meaning is lost
Allows personalization	No personalization
Pagination for long text output	No pagination on long text output
Progress bars/spinners to show processing	No progress bar to indicate processing
Importing information at the bottom	Importing information at the top of output or missing

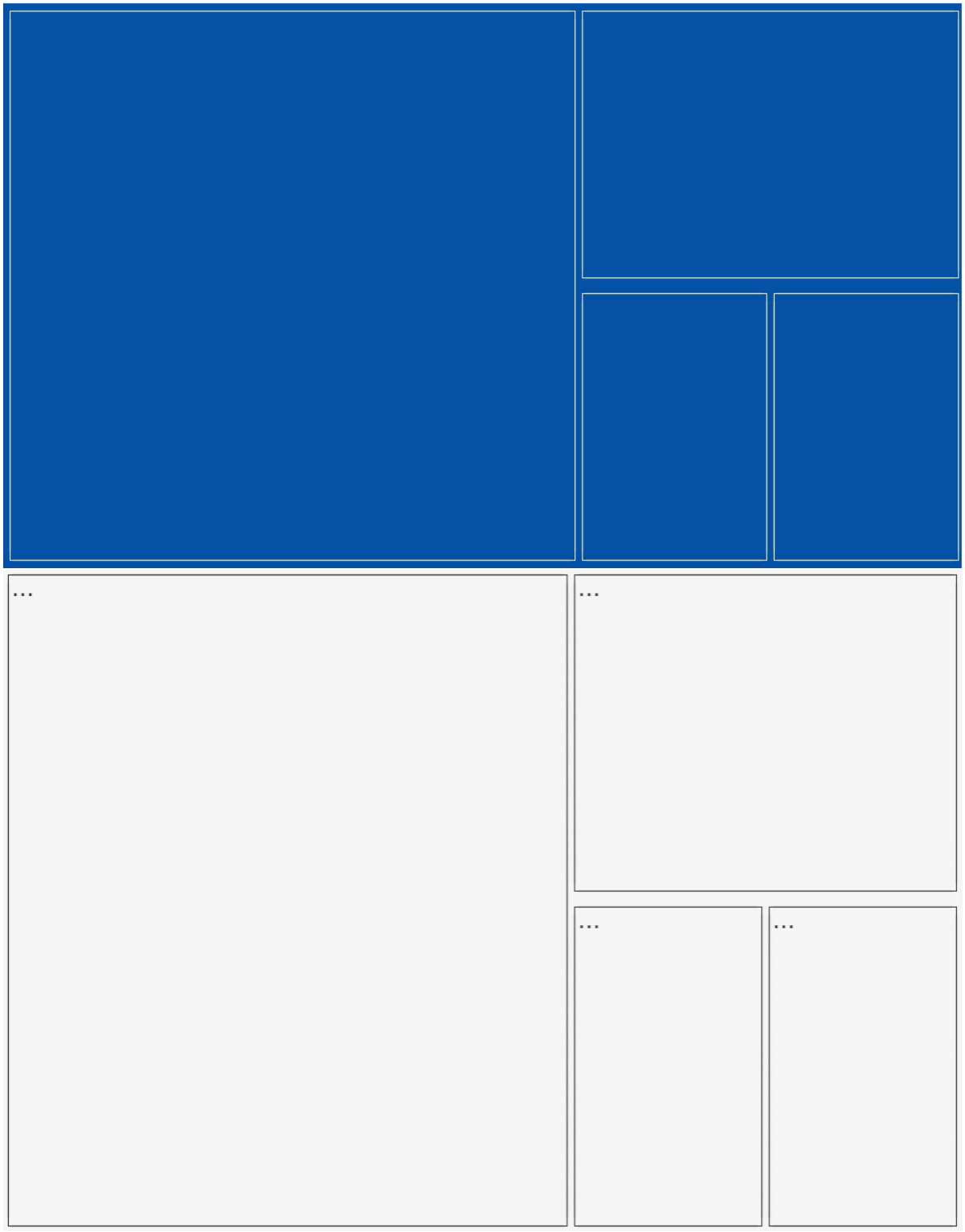
Chapter 9: Empathic Side of Development

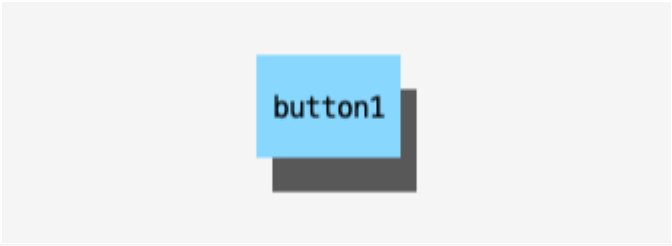
```
Attempting to vote at 12 years old...
error occurred:  eligibleToVote: ❌ too young to vote, at 12, wait 6 more years
```

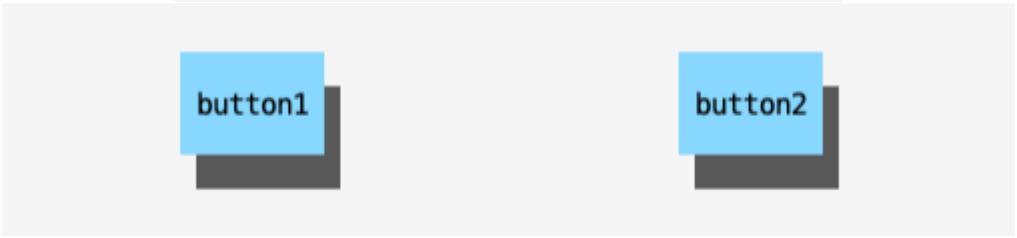


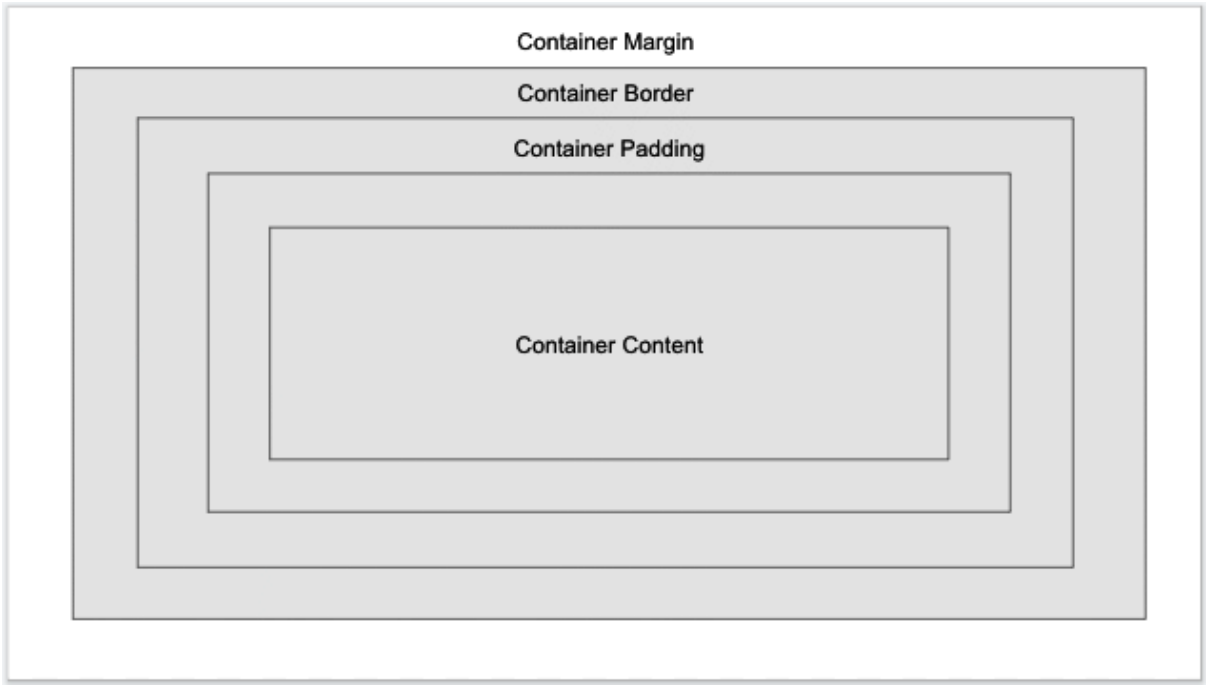


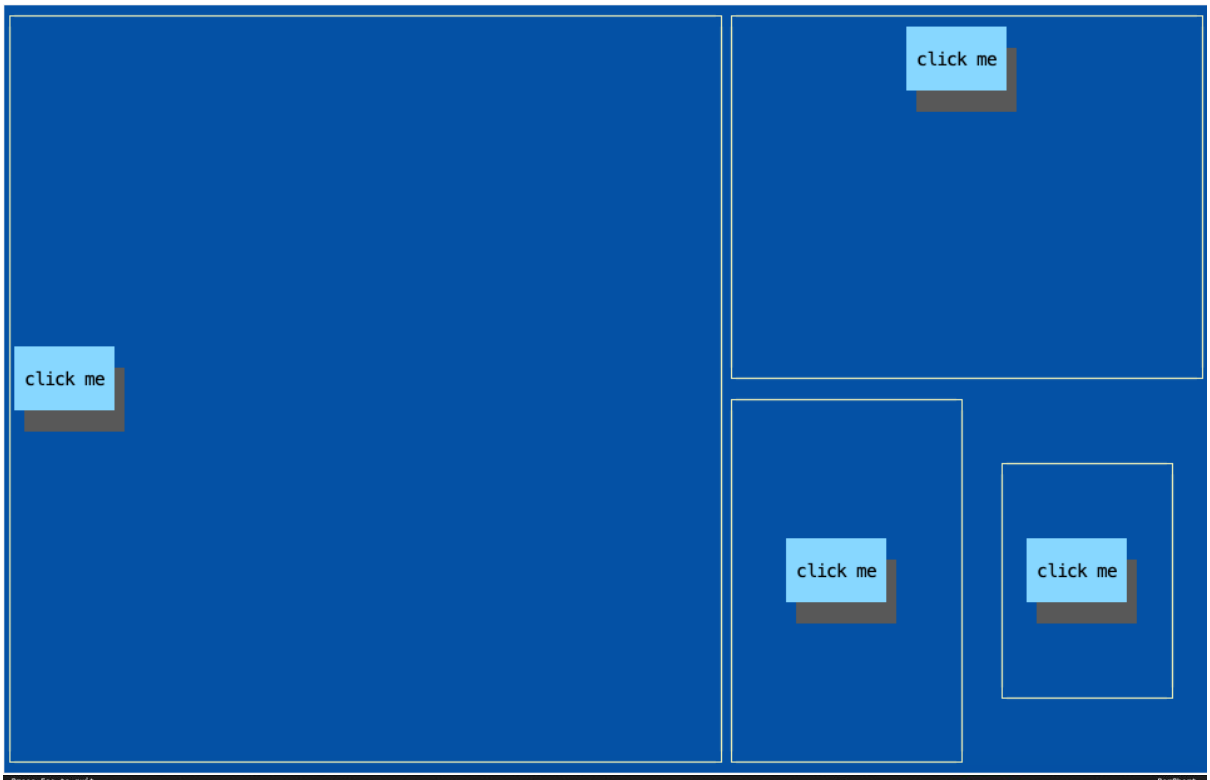
Chapter 10: Interactivity with Prompts and Terminal Dashboards

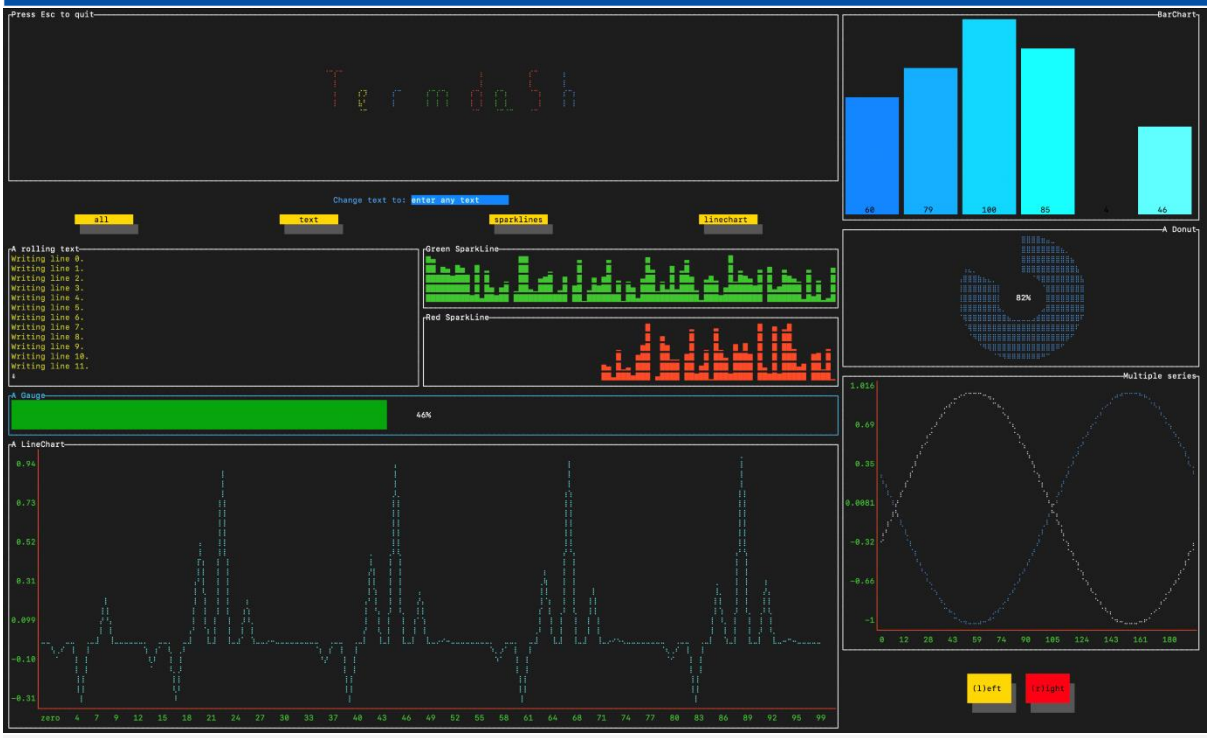


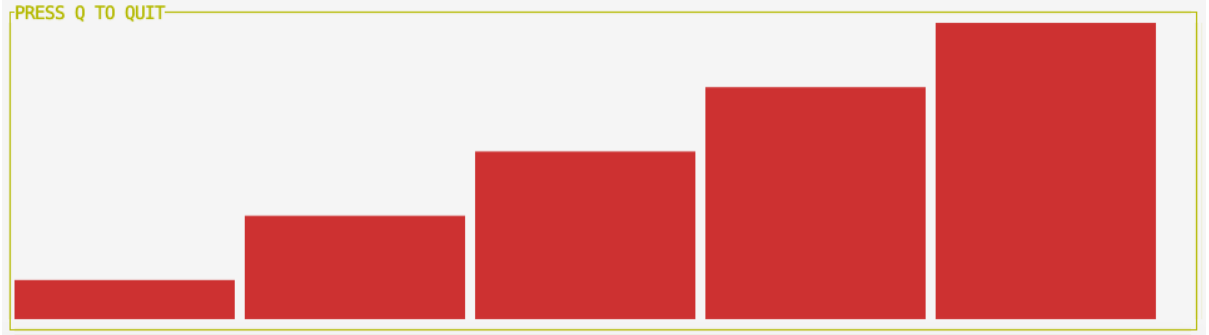

button1

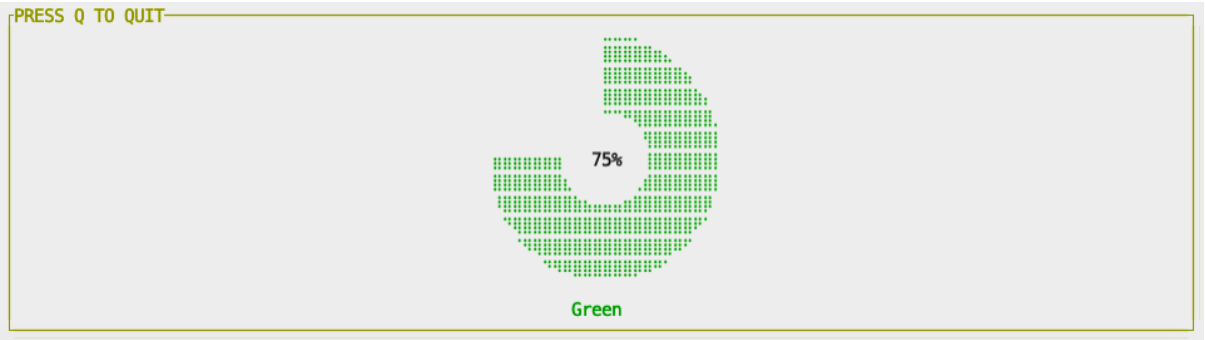

button1
button2

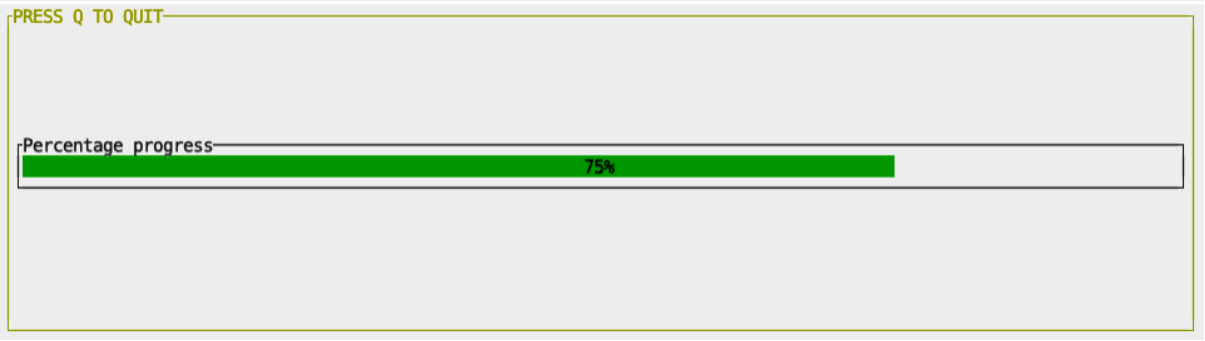

Container Margin
Container Border
Container Padding
Container Content


click me
click me
click me
click me


Press Esc to quit
Terminal
Change text to: enter any text
all
text
sparklines
linechart
A rolling text
Writing line 0.
Writing line 1.
Writing line 2.
Writing line 3.
Writing line 4.
Writing line 5.
Writing line 6.
Writing line 7.
Writing line 8.
Writing line 9.
Writing line 10.
Writing line 11.
Green SparkLine
Red SparkLine
A Gauge
46%
A LineChart
BarChart
60
79
100
85
4
46
A Donut
82%
Multiple series
(l)eft
(r)ight

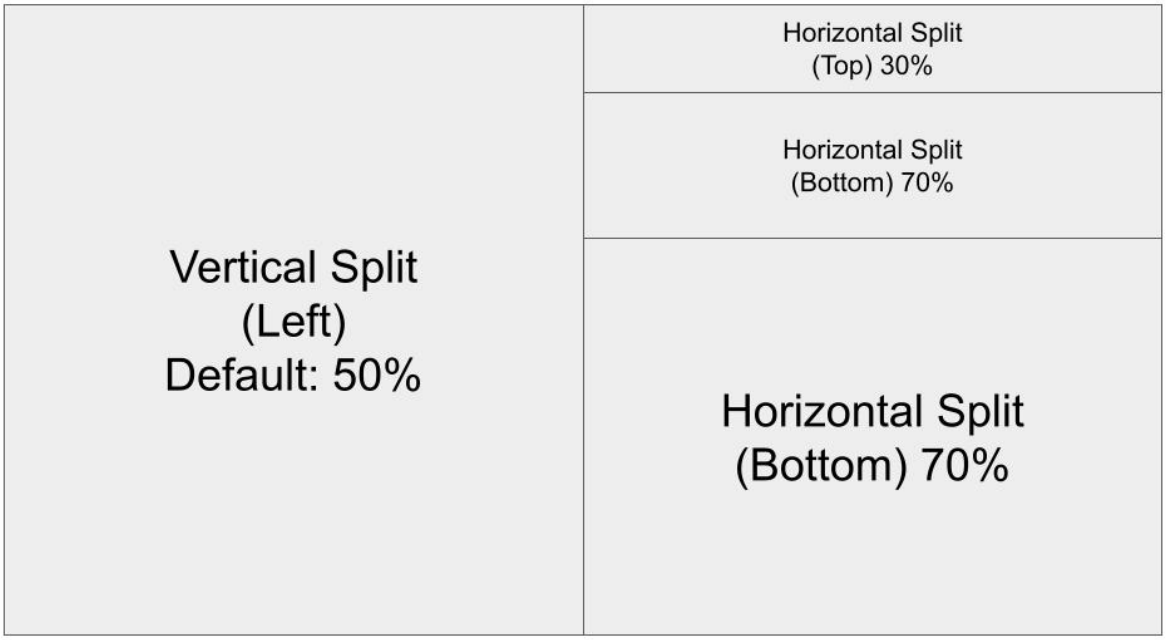

PRESS Q TO QUIT

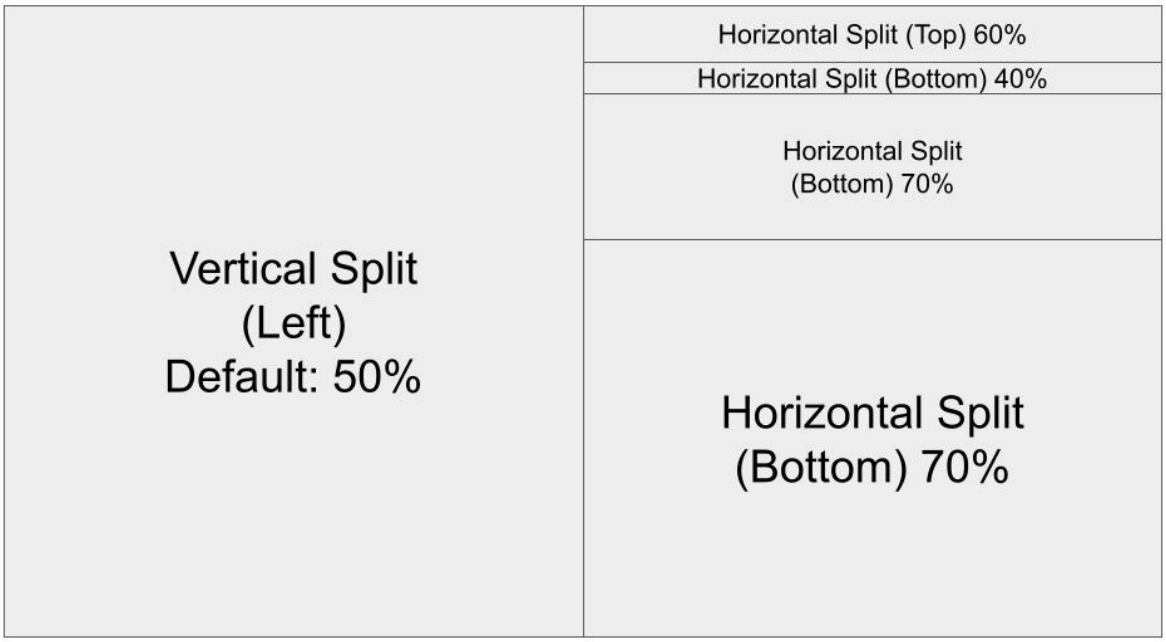

PRESS Q TO QUIT
75%
Green

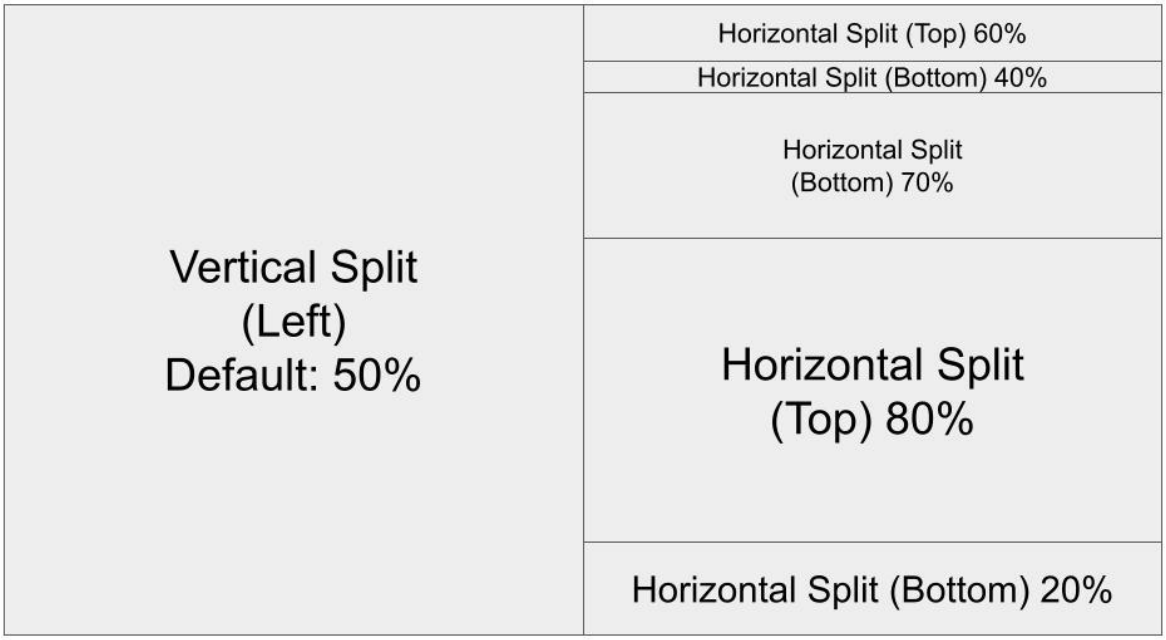

PRESS Q TO QUIT
Percentage progress
75%

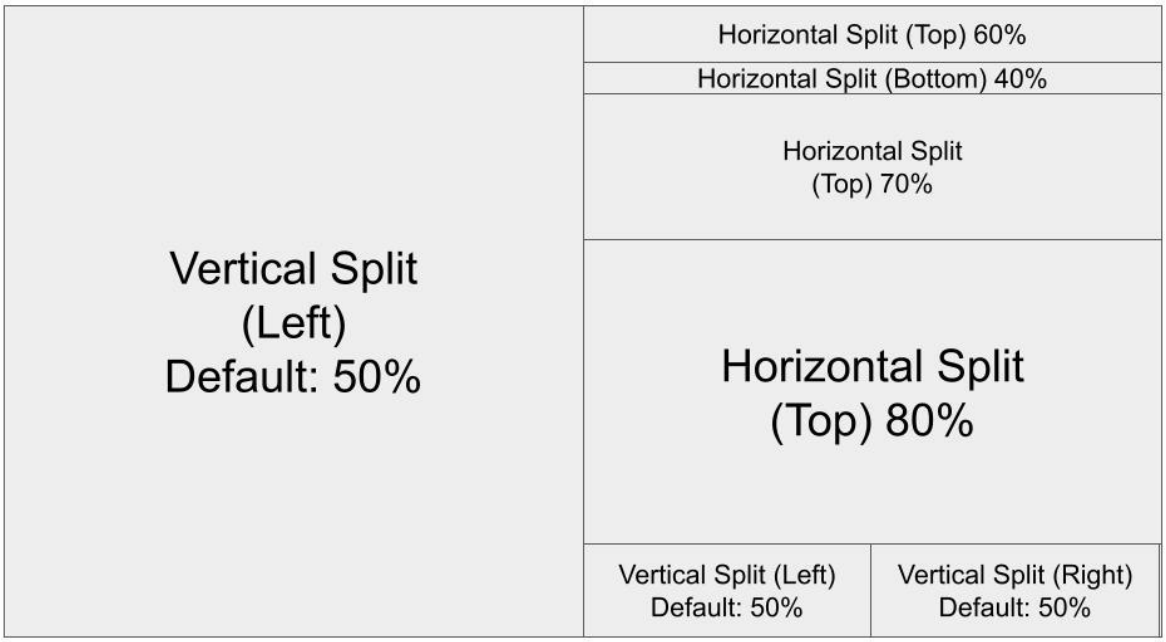

Music Library List
Input
Error messages
Selected Song:
Title by Artist
Metadata
Play Button
Stop Button

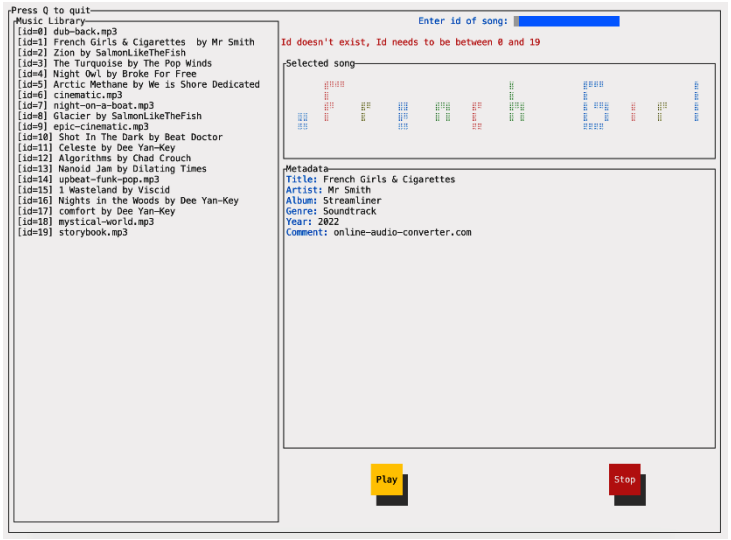

Vertical Split
(Left)
Default: 50%
Vertical Split
(Right)
Default: 50%
Vertical Split
(Left)
Default: 50%
Horizontal Split
(Top) 30%
Horizontal Split
(Bottom) 70%


Vertical Split
(Left)
Default: 50%
Horizontal Split
(Top) 30%
Horizontal Split
(Bottom) 70%
Horizontal Split
(Bottom) 70%


Vertical Split
(Left)
Default: 50%
Horizontal Split (Top) 60%
Horizontal Split (Bottom) 40%
Horizontal Split
(Bottom) 70%
Horizontal Split
(Bottom) 70%


Vertical Split
(Left)
Default: 50%
Horizontal Split (Top) 60%
Horizontal Split (Bottom) 40%
Horizontal Split
(Bottom) 70%
Horizontal Split
(Top) 80%
Horizontal Split (Bottom) 20%


Vertical Split
(Left)
Default: 50%
Horizontal Split (Top) 60%
Horizontal Split (Bottom) 40%
Horizontal Split
(Top) 70%
Horizontal Split
(Top) 80%
Vertical Split (Left)
Default: 50%
Vertical Split (Right)
Default: 50%

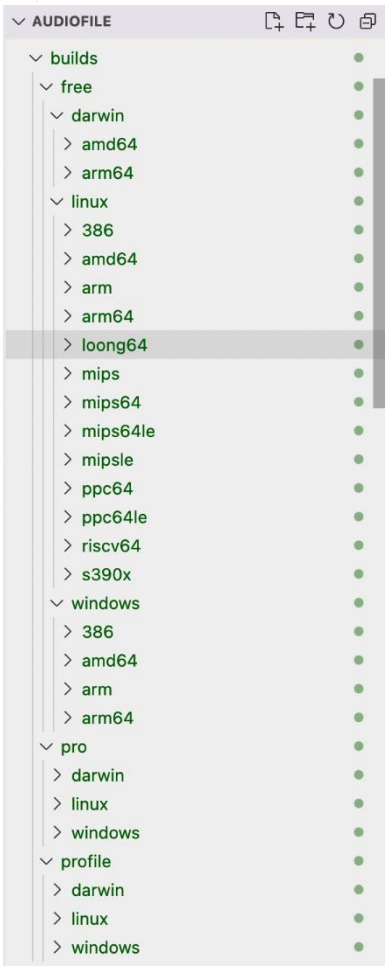

Press Q to quit
Music Library
[id=0] dub-back.mp3
[id=1] French Girls & Cigarettes by Mr Smith
[id=2] Zion by SalmonLikeTheFish
[id=3] The Turquoise by The Pop Winds
[id=4] Night Owl by Broke For Free
[id=5] Arctic Methane by We is Shore Dedicated
[id=6] cinematic.mp3
[id=7] night-on-a-boat.mp3
[id=8] Glacier by SalmonLikeTheFish
[id=9] epic-cinematic.mp3
[id=10] Shot In The Dark by Beat Doctor
[id=11] Celeste by Dee Yan-Key
[id=12] Algorithms by Chad Crouch
[id=13] Nanoid Jam by Dilating Times
[id=14] upbeat-funk-pop.mp3
[id=15] 1 Wasteland by Viscid
[id=16] Nights in the Woods by Dee Yan-Key
[id=17] comfort by Dee Yan-Key
[id=18] mystical-world.mp3
[id=19] storybook.mp3
Enter id of song:
Id doesn't exist, Id needs to be between 0 and 19
Selected song
Metadata
Title: French Girls & Cigarettes
Artist: Mr Smith
Album: Streamliner
Genre: Soundtrack
Year: 2022
Comment: online-audio-converter.com
Play
Stop

Chapter 11: Customizing Go Binaries with Build Tags

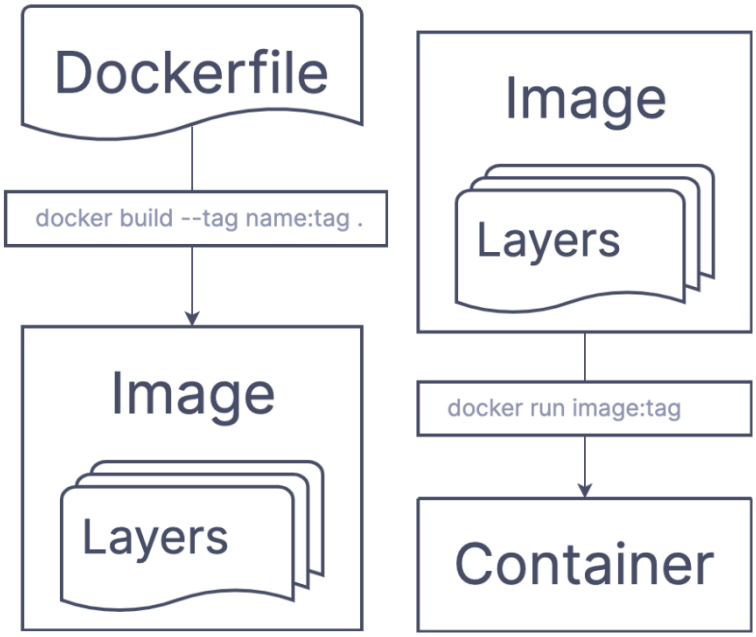
No images…

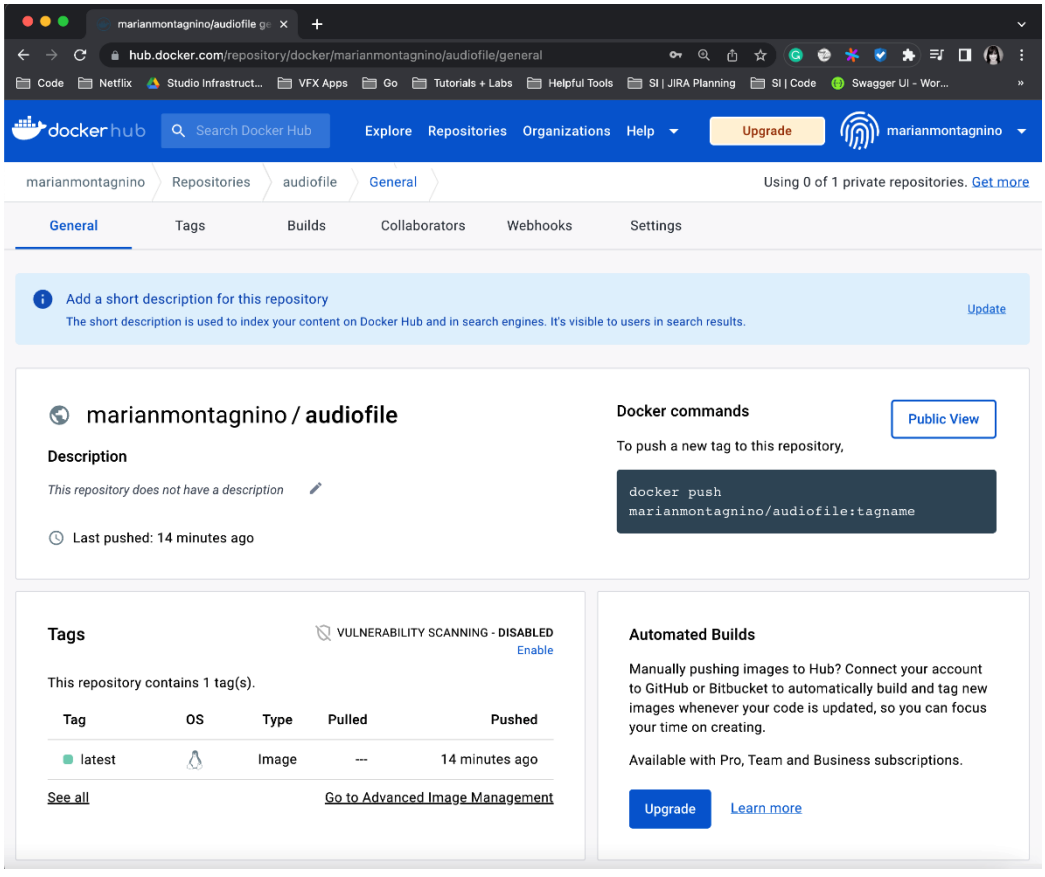
Chapter 12: Cross Compilation Across Different Platforms

```
aix/ppc64          linux/386          netbsd/arm64
android/386        linux/amd64        openbsd/386
android/amd64      linux/arm          openbsd/amd64
android/arm        linux/arm64        openbsd/arm
android/arm64      linux/loong64      openbsd/arm64
darwin/amd64       linux/mips         openbsd/mips64
darwin/arm64       linux/mips64       plan9/386
dragonfly/amd64    linux/mips64le     plan9/amd64
freebsd/386        linux/mipsle       plan9/arm
freebsd/amd64      linux/ppc64        solaris/amd64
freebsd/arm        linux/ppc64le      windows/386
freebsd/arm64      linux/riscv64      windows/amd64
illumos/amd64      linux/s390x        windows/arm
ios/amd64          netbsd/386         windows/arm64
ios/arm64          netbsd/amd64
js/wasm            netbsd/arm
```



Chapter 13: Using Containers for Distribution





README

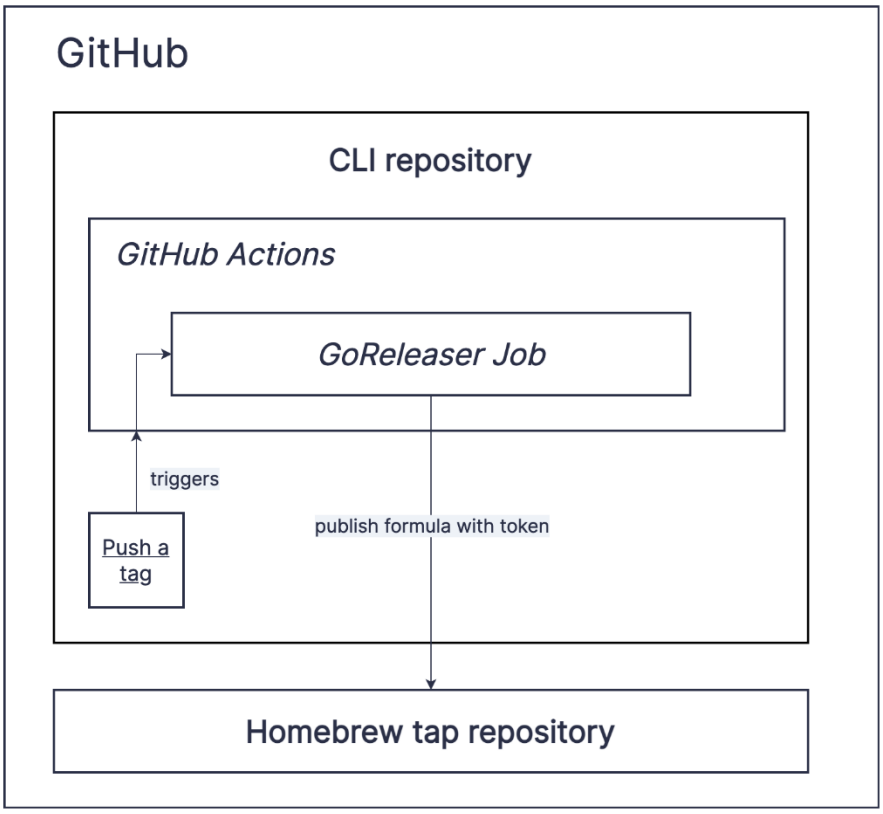
Repository description is empty. Click here to edit.

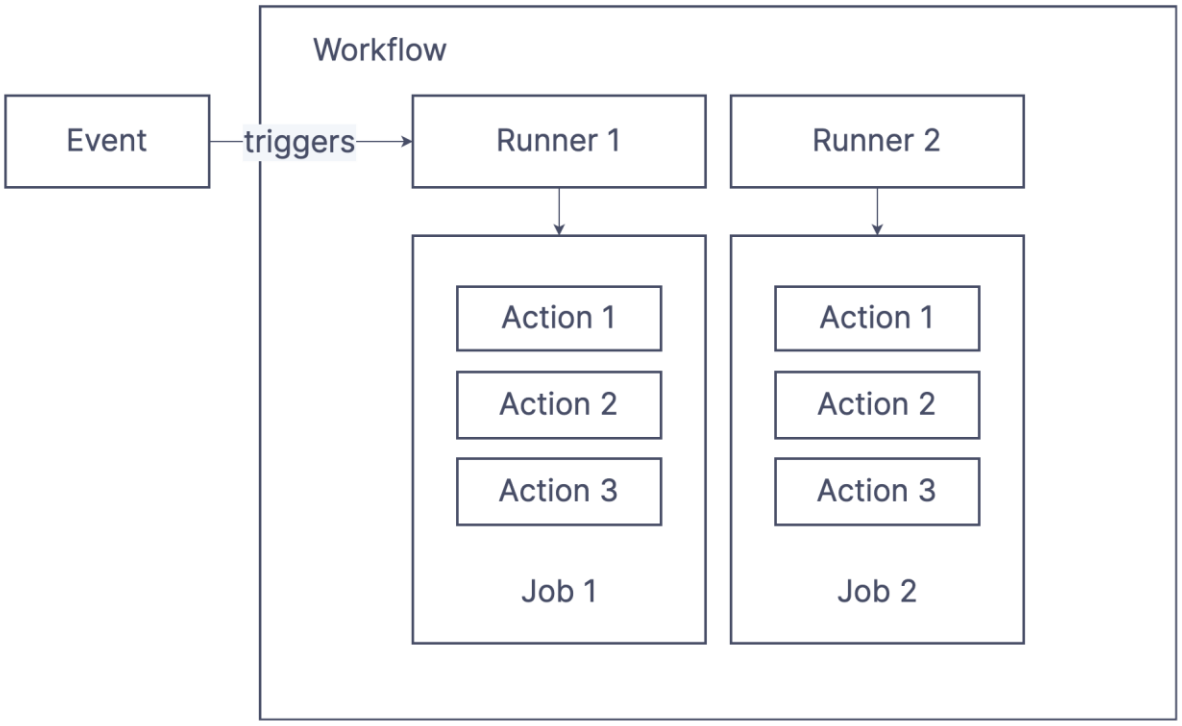
README

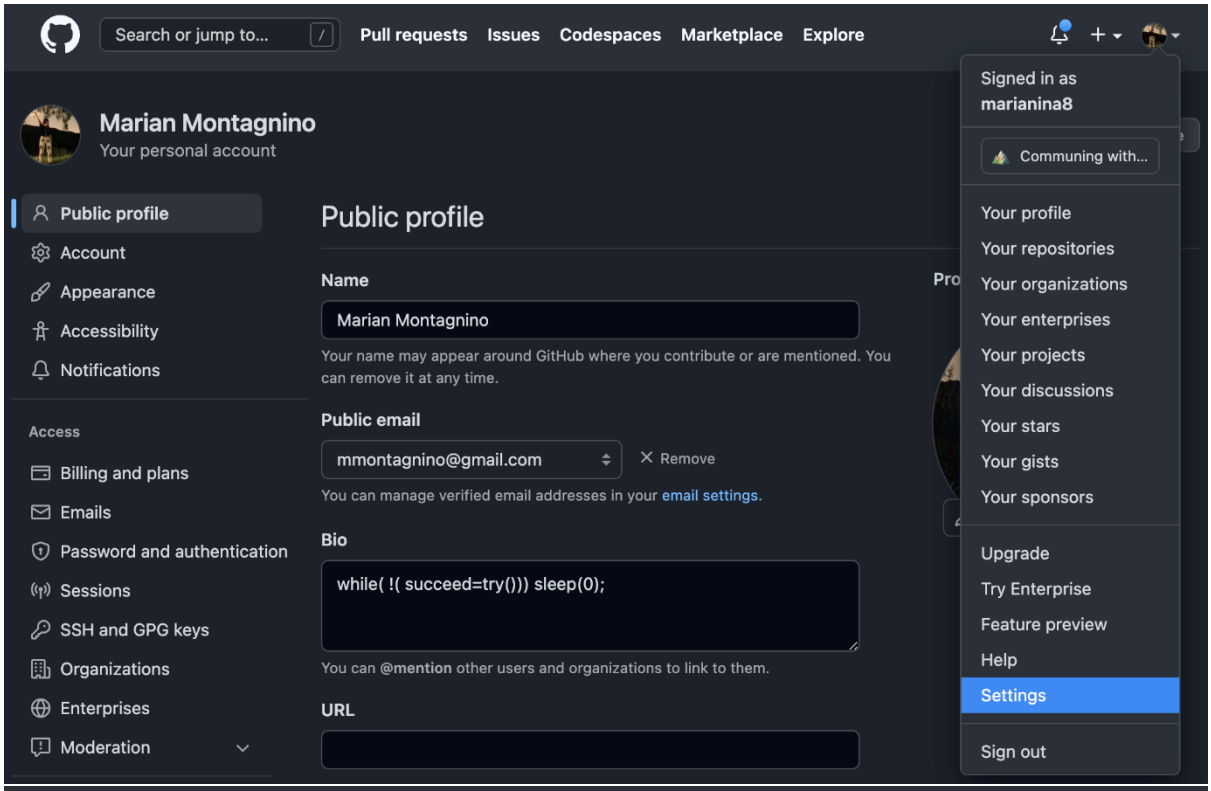
To run the audiofile CLI container, ensure that the audiofile API container is running first. Next, run the docker command:

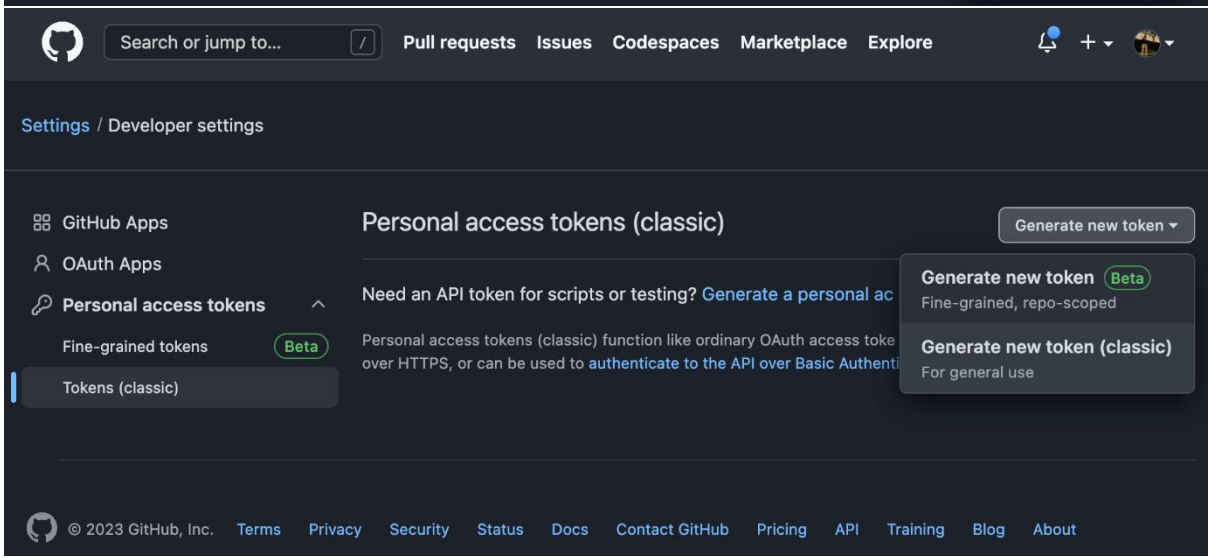
```
docker run --rm --network host -ti marianmontagnino/audiofile:latest help
```

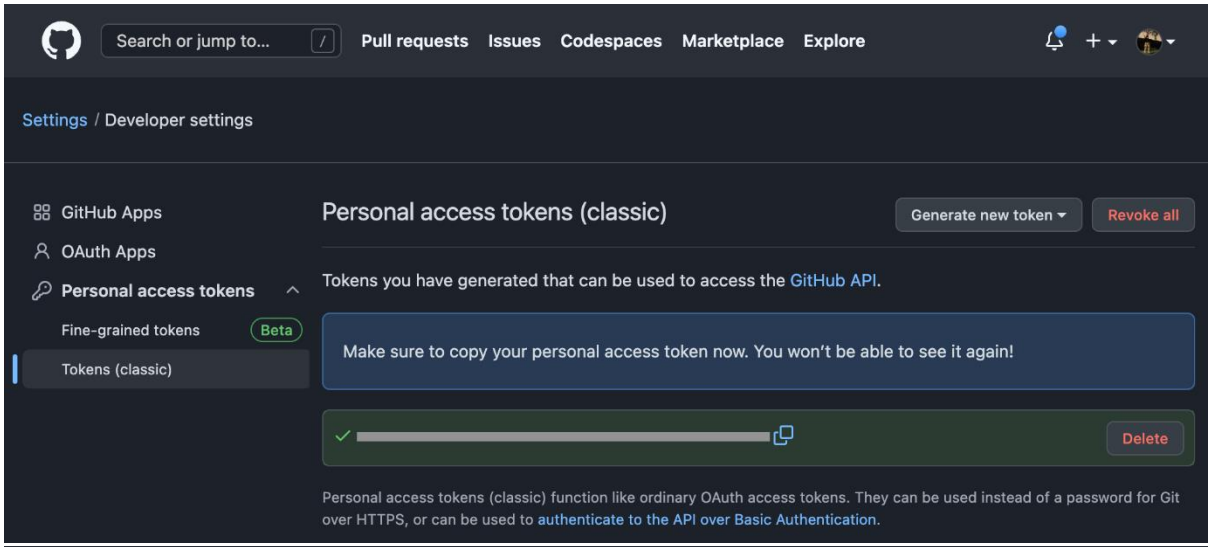
Chapter 14: Publish your Go binary as a Homebrew Formula with GoReleaser

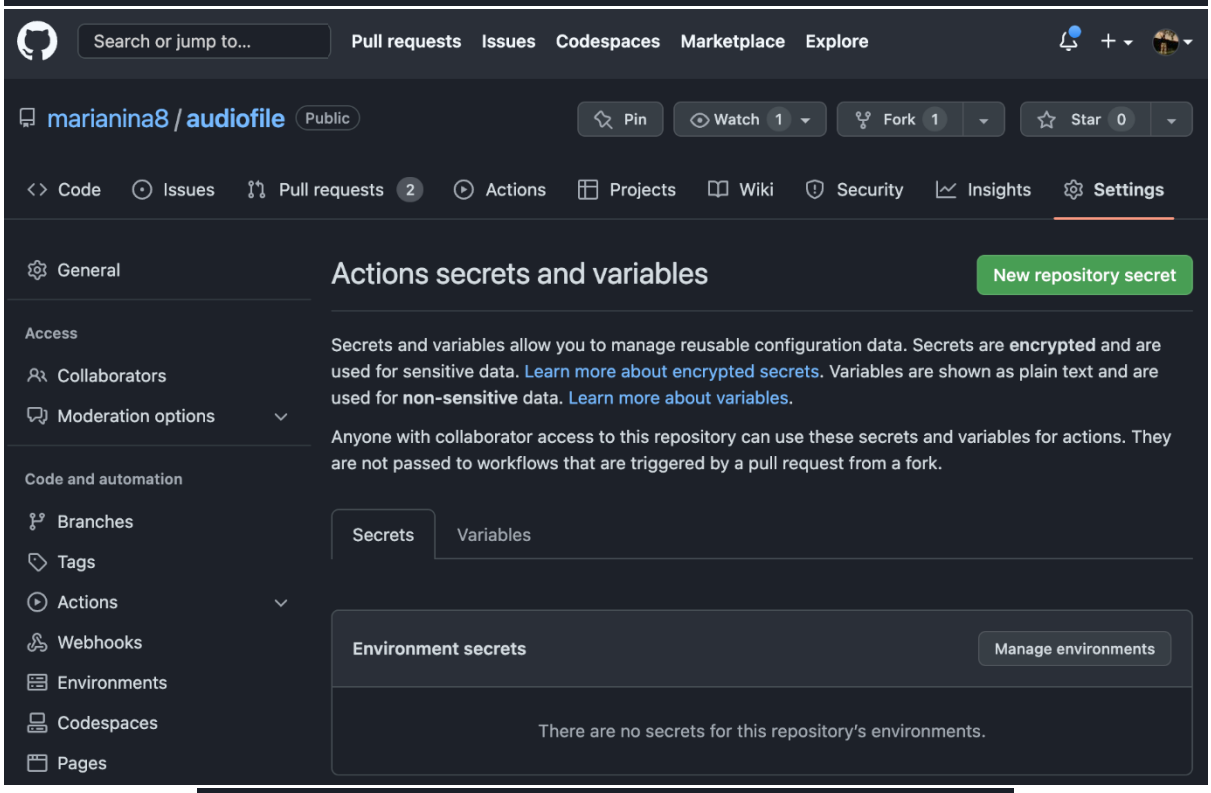


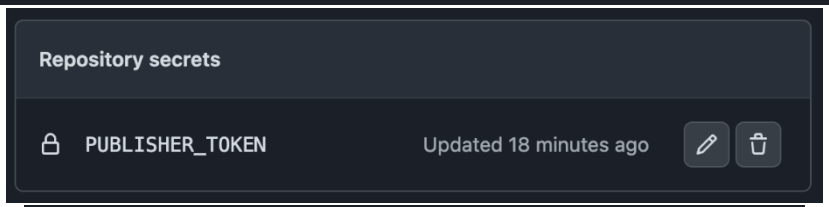


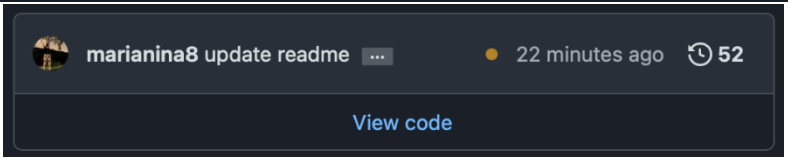

Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
Signed in as marianina8
Communing with...
Your profile
Your repositories
Your organizations
Your enterprises
Your projects
Your discussions
Your stars
Your gists
Your sponsors
Upgrade
Try Enterprise
Feature preview
Help
Settings
Sign out
Marian Montagnino
Your personal account
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation
Public profile
Name
Marian Montagnino
Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.
Public email
mmontagnino@gmail.com
Remove
You can manage verified email addresses in your email settings.
Bio
while(!(succeed=try())) sleep(0);
You can @mention other users and organizations to link to them.
URL

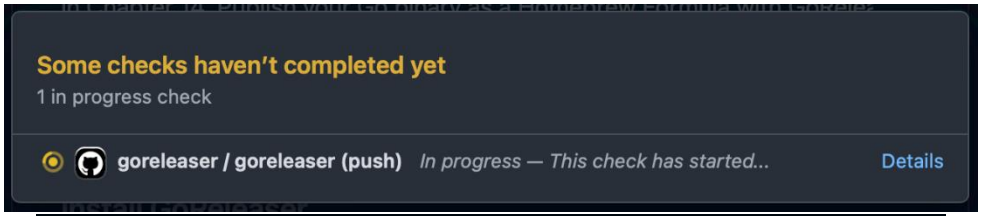

Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
Settings / Developer settings
GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens
Beta
Tokens (classic)
Personal access tokens (classic)
Generate new token
Need an API token for scripts or testing? Generate a personal ac
Personal access tokens (classic) function like ordinary OAuth access toke
over HTTPS, or can be used to authenticate to the API over Basic Authenti
Generate new token Beta
Fine-grained, repo-scoped
Generate new token (classic)
For general use
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

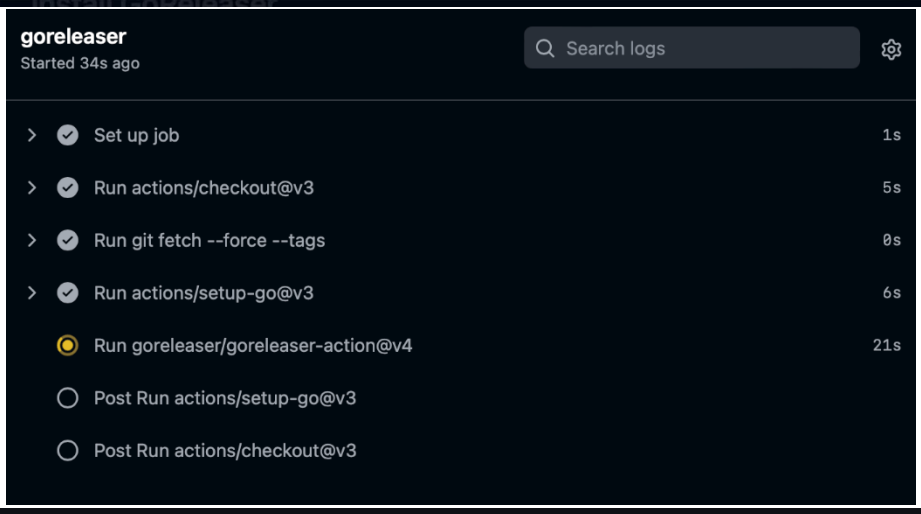

Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
Settings / Developer settings
GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens
Beta
Tokens (classic)
Personal access tokens (classic)
Generate new token
Revoke all
Tokens you have generated that can be used to access the GitHub API.
Make sure to copy your personal access token now. You won't be able to see it again!
Delete
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.


Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
marianina8 / audiofile
Public
Pin
Watch 1
Fork 1
Star 0
Code
Issues
Pull requests 2
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Actions
Webhooks
Environments
Codespaces
Pages
Actions secrets and variables
New repository secret
Secrets and variables allow you to manage reusable configuration data. Secrets are encrypted and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for non-sensitive data. Learn more about variables.
Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.
Secrets
Variables
Environment secrets
Manage environments
There are no secrets for this repository's environments.


Repository secrets
PUBLISHER_TOKEN
Updated 18 minutes ago


marianina8 update readme
22 minutes ago
52
View code


Some checks haven't completed yet
1 in progress check
goreleaser / goreleaser (push) In progress — This check has started...
Details


goreleaser
Started 34s ago
Search logs
Set up job 1s
Run actions/checkout@v3 5s
Run git fetch --force --tags 0s
Run actions/setup-go@v3 6s
Run goreleaser/goreleaser-action@v4 21s
Post Run actions/setup-go@v3
Post Run actions/checkout@v3


Assets 8
audiofile_Darwin_all.tar.gz 7.45 MB 22 minutes ago
audiofile_Linux_arm64.tar.gz 3.35 MB 22 minutes ago
audiofile_Linux_x86_64.tar.gz 3.68 MB 22 minutes ago
audiofile_Windows_arm64.zip 3.41 MB 22 minutes ago
audiofile_Windows_x86_64.zip 3.74 MB 22 minutes ago
checksums.txt 474 Bytes 22 minutes ago
Source code (zip) 25 minutes ago
Source code (tar.gz) 25 minutes ago

```
mmontagnino@Marians-MacBook-Pro audiofile % audiofile
A command line interface allows you to interact with the Audiofile service.
Basic commands include: get, list, and upload.

Usage:
  audiofile [command]

Available Commands:
  api         Start or stop the API required by the CLI
  bug         Submit a bug
  completion  Generate the autocompletion script for the specified shell
  delete      Delete audiofile by id
  get         Get audio metadata
  help        Help about any command
  list        List all audio files
  play        Play audio file by id
  player      Launch player dashboard
  search      Command to search for audiofiles by string
  upload      Upload an audio file
  version     Version of audiofile CLI

Flags:
  -h, --help      help for audiofile
  -t, --toggle    Help message for toggle
  -v, --verbose   verbose

Use "audiofile [command] --help" for more information about a command.
```