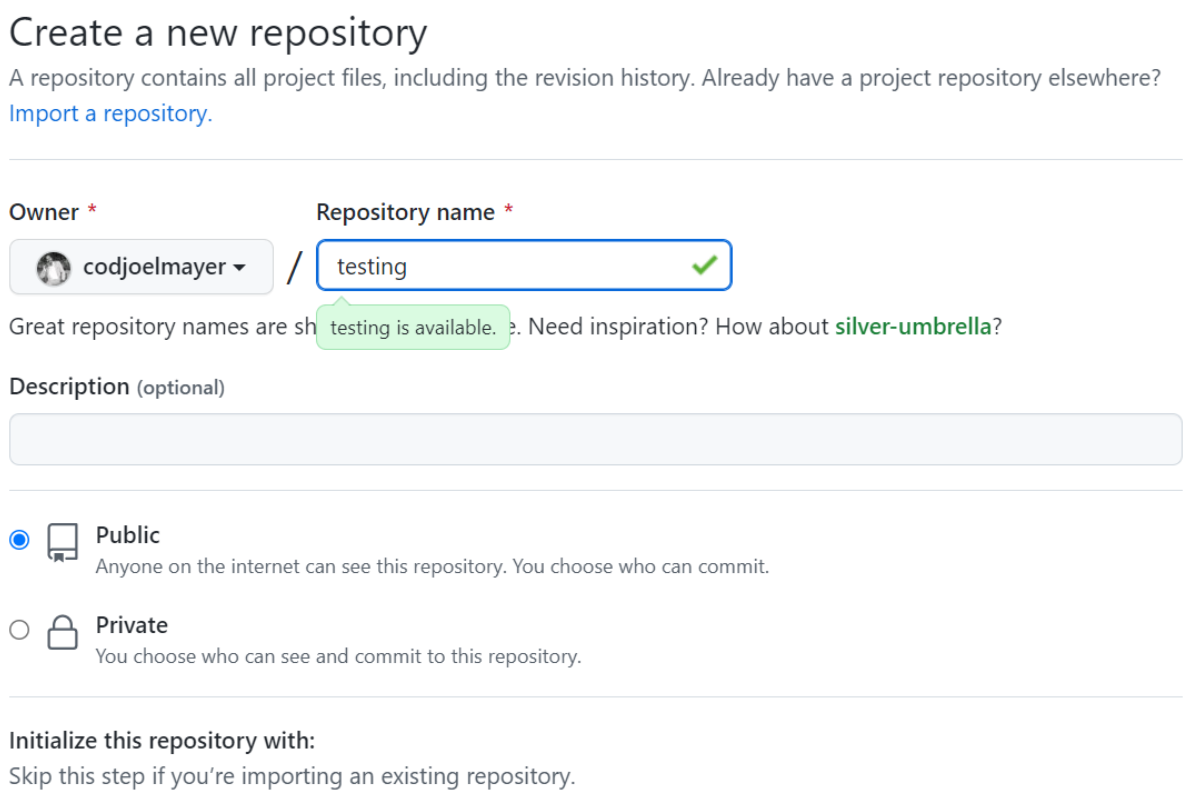
Guide on git and hosting VueJS on Firebase

NB: This document is a step-by-step guide from creating a GitHub repository to deployment on Firebase and render.com. We will deploy our backend to render and VueJS on Firebase.

1. Let's create a repo. “It will be the name of your project”

   I will name my repo “testing”. Go to https://github.com/
   Once again, don’t select readMe; we will add one later, because if you do, it will set up a main branch for you.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * codjoelmayer / Repository name * testing

Great repository names are sh testing is available. e. Need inspiration? How about silver-umbrella?

Description (optional)

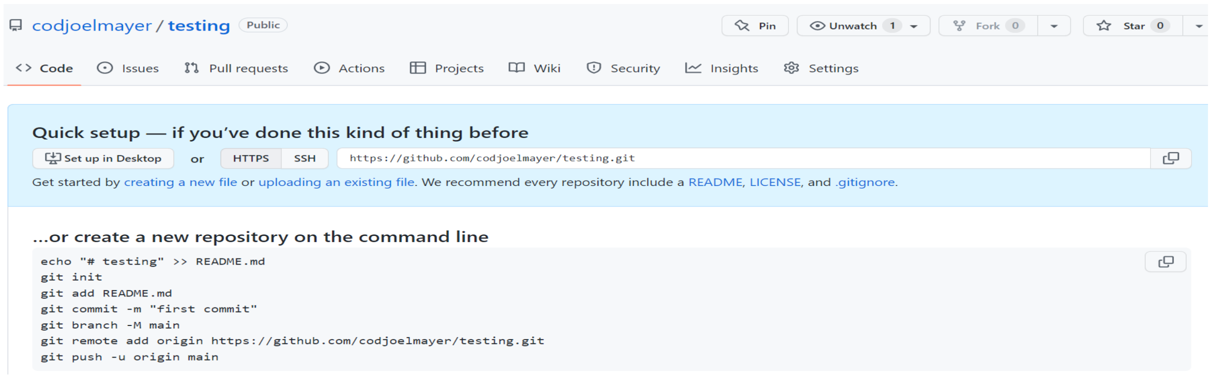
● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you’re importing an existing repository.

And press on “**Create repository**”

You should be able to see the screen below.



Quick setup — if you’ve done this kind of thing before

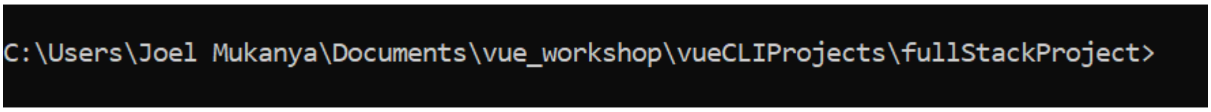
https://github.com/codjoelmayer/testing.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

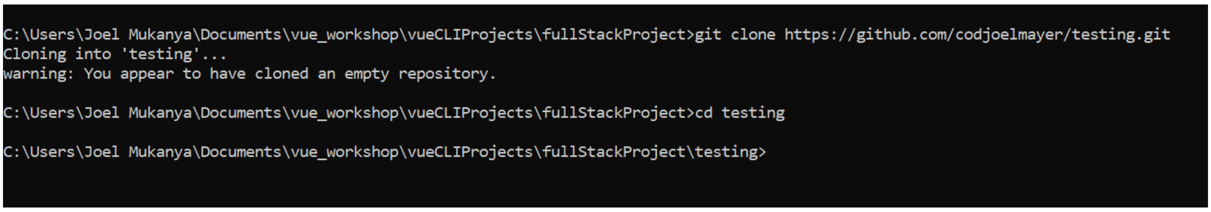
```
echo "# testing" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/codjoelmayer/testing.git
git push -u origin main
```

Now we have to clone this repository.
Open your command prompt. And locate the main directory, to which you want to clone your repository.



```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject>
```

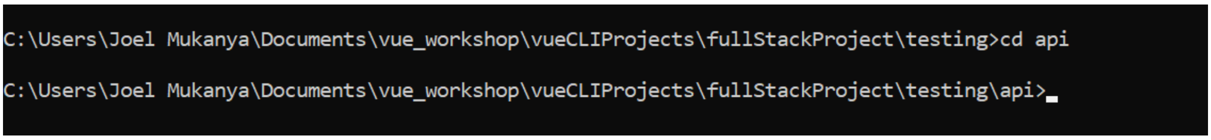
Now you have to clone your repository. and change the directory. Make sure you are on the cloned repository.



```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject>git clone https://github.com/codjoelmayer/testing.git
Cloning into 'testing'...
warning: You appear to have cloned an empty repository.

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject>cd testing

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>
```

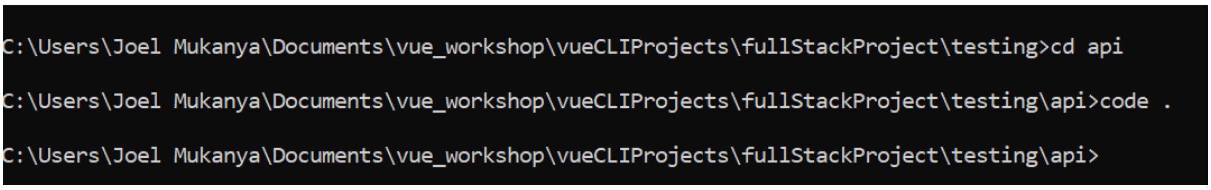
2. Please create an api folder under our cloned repository.

3. Our next step is to focus on the api so we have to change our directory to api.



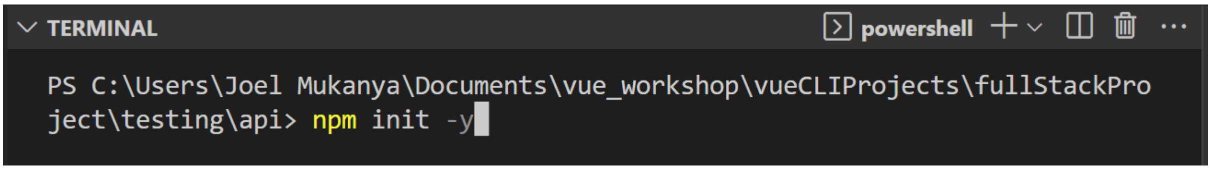
```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>cd api

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\api>
```

Open VS Code from here.



```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>cd api

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\api>code .

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\api>
```

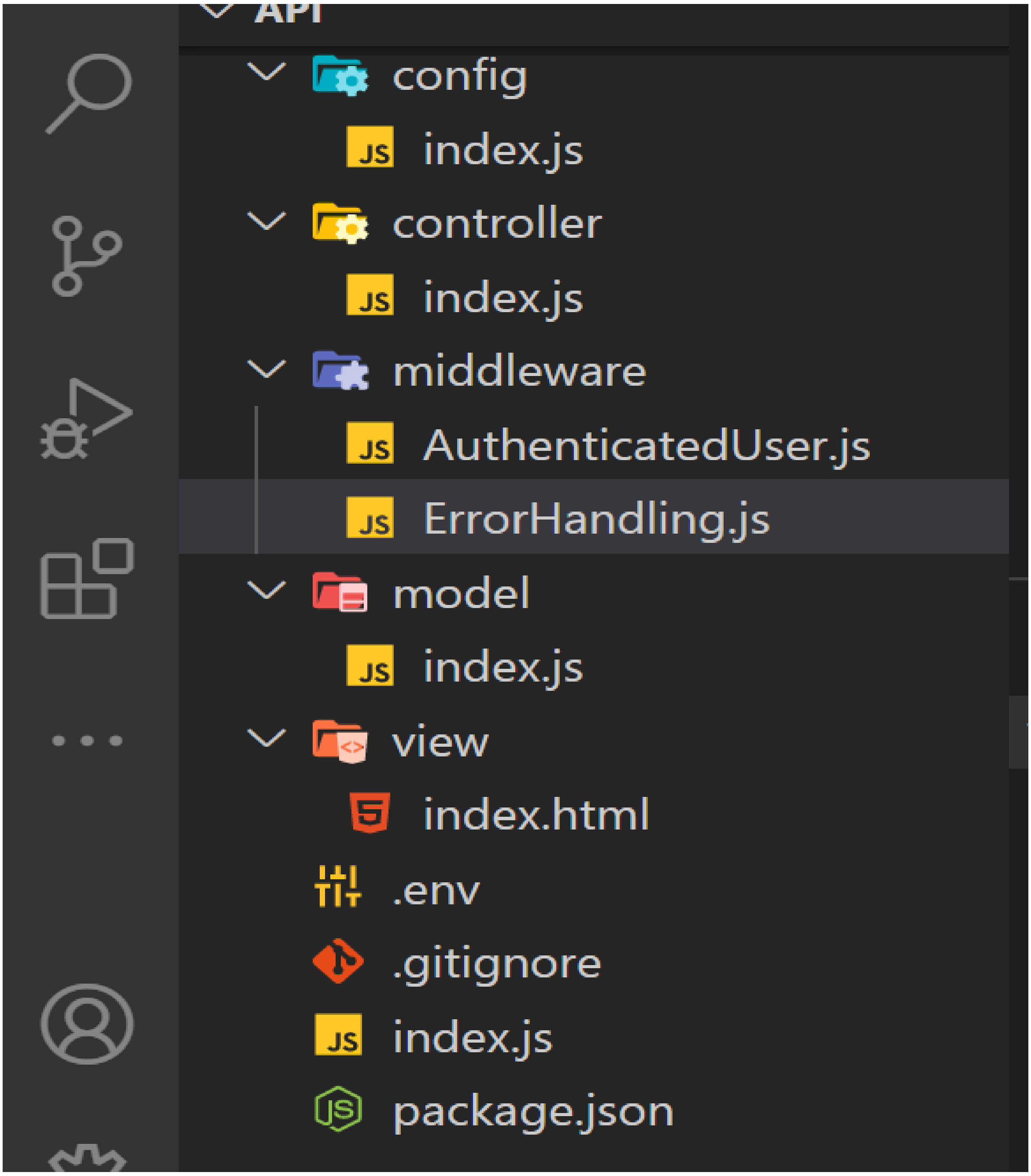
Run the below command to initialise our node application



```
PS C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackPro
ject\testing\api> npm init -y
```

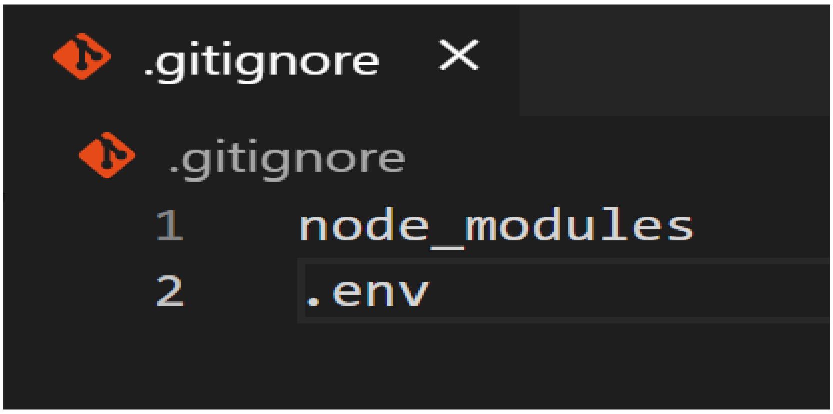
After you initialise your node application, you will be able to see the package.json.

4. Let’s create all necessary folders and files



```
API
  config
    index.js
  controller
    index.js
  middleware
    AuthenticatedUser.js
    ErrorHandling.js
  model
    index.js
  view
    index.html
  .env
  .gitignore
  index.js
  package.json
```

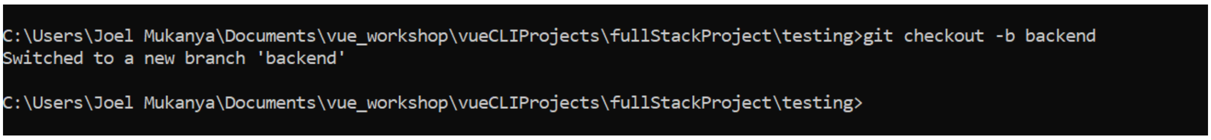
5. Specify all files and folders that it shouldn't be push to GitHub



.gitignore

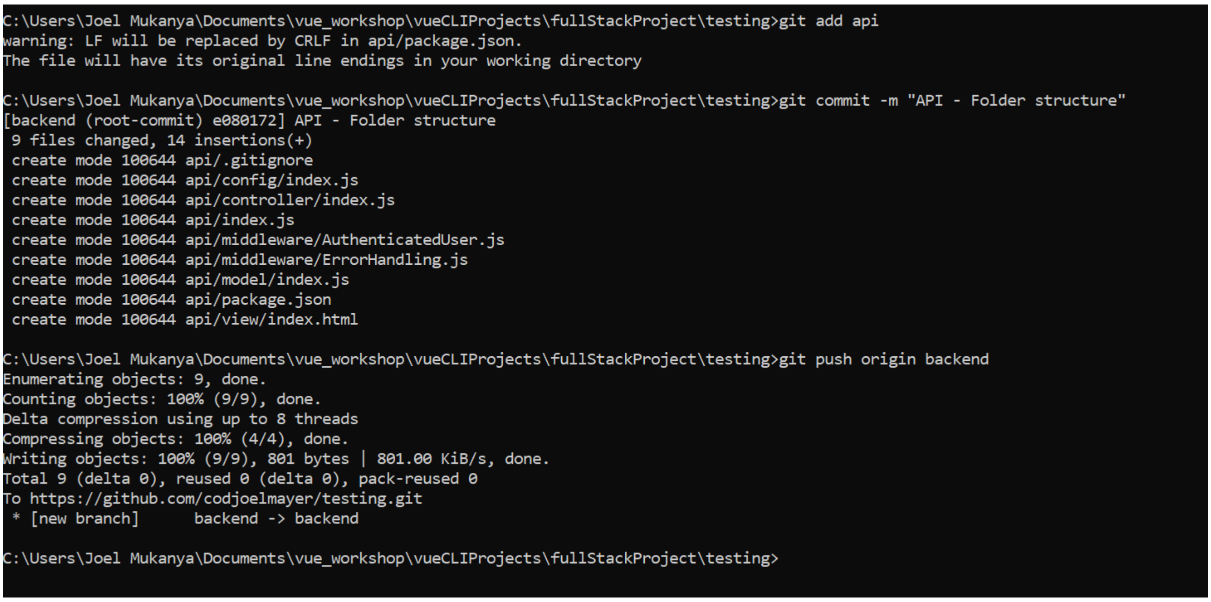
```
node_modules
.env
```

6. Let’s create a branch called “backend and push our folder structure.”
   NB: Make sure you are on your cloned repository

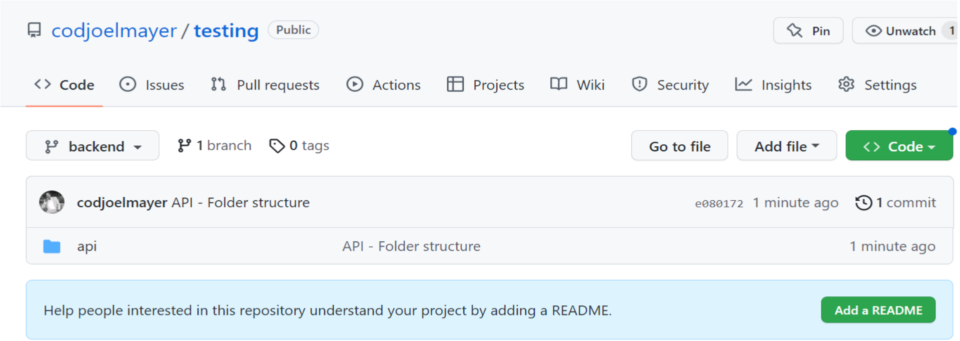


```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git checkout -b backend
Switched to a new branch 'backend'

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>
```

   Let’s add our api folder to the staging area then commit

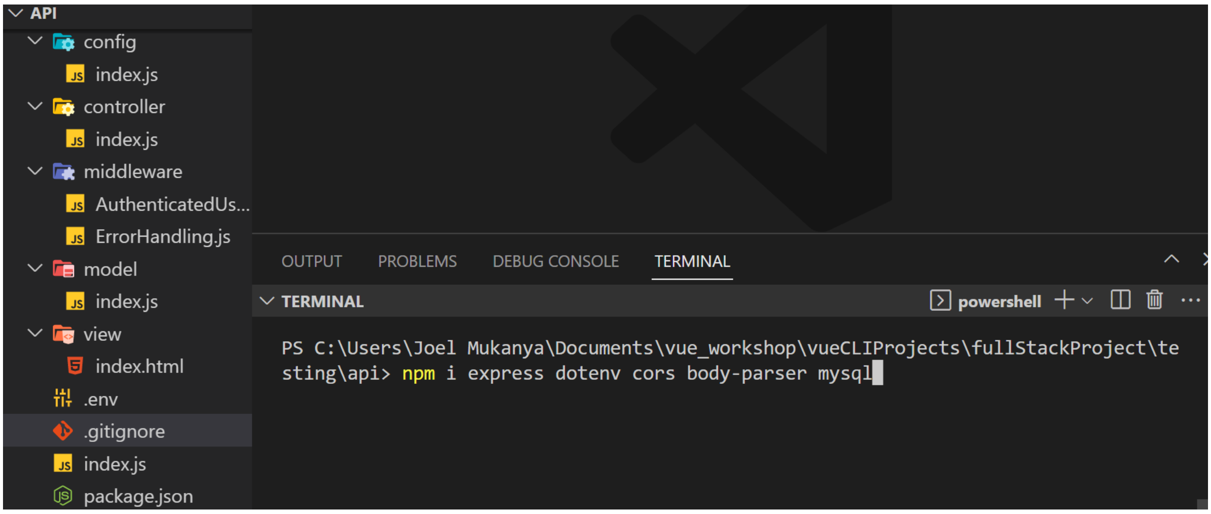


```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git add api
warning: LF will be replaced by CRLF in api/package.json.
The file will have its original line endings in your working directory

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git commit -m "API - Folder structure"
[backend (root-commit) e080172] API - Folder structure
 9 files changed, 14 insertions(+)
 create mode 100644 api/.gitignore
 create mode 100644 api/config/index.js
 create mode 100644 api/controller/index.js
 create mode 100644 api/index.js
 create mode 100644 api/middleware/AuthenticatedUser.js
 create mode 100644 api/middleware/ErrorHandling.js
 create mode 100644 api/model/index.js
 create mode 100644 api/package.json
 create mode 100644 api/view/index.html

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git push origin backend
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 801 bytes | 801.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/codjoelmayer/testing.git
 * [new branch]      backend -> backend

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>
```



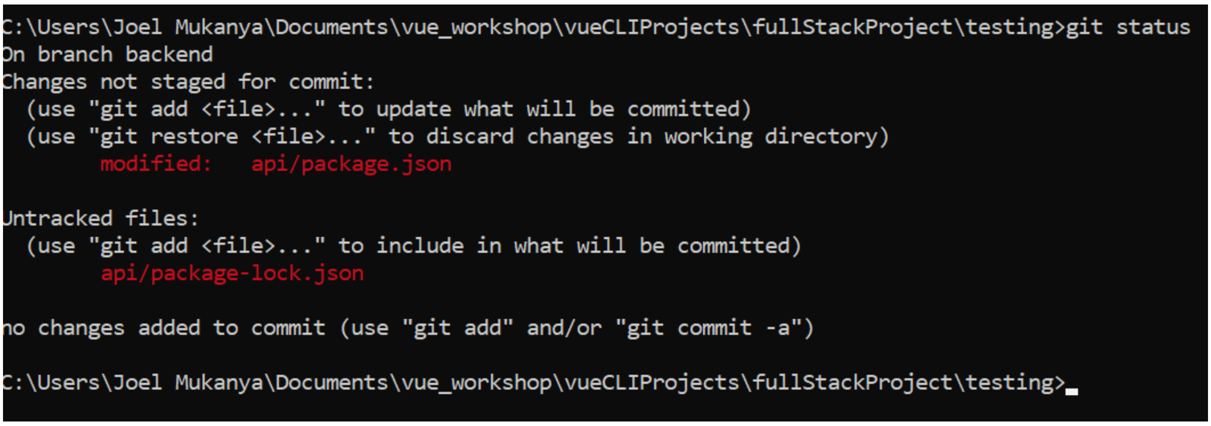
Help people interested in this repository understand your project by adding a README.

7. Let’s install all packages that we going to use
   Install the following packages locally: npm i express dotenv cors body-parser mysql
   Install the nodemon package globally: npm i -g nodemon --save-dev
   npm i bcrypt jsonwebtoken cookie-parser

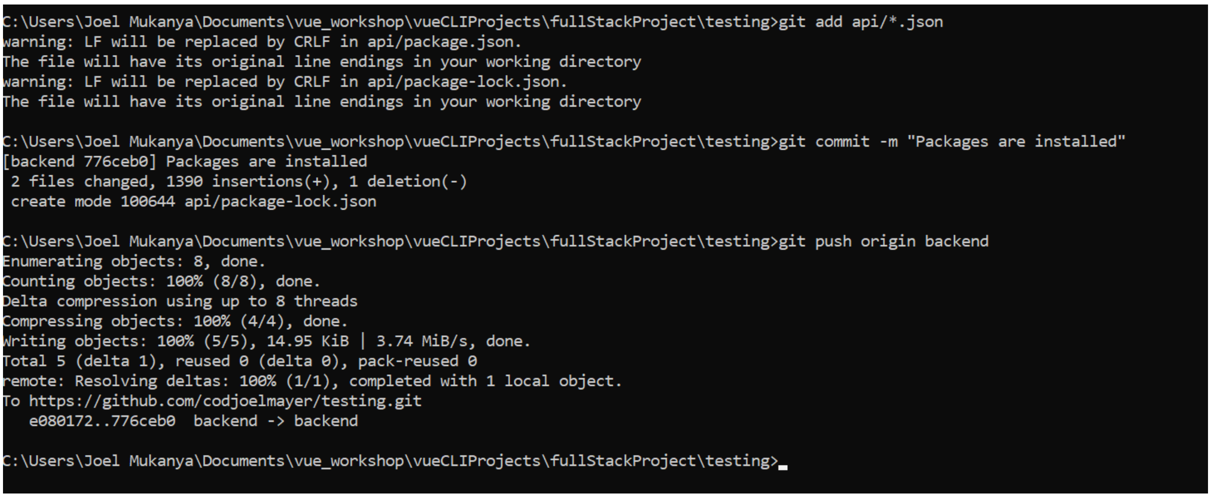


```
PS C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\te
sting\api> npm i express dotenv cors body-parser mysql
```

Let’s push our packages files to our GitHub repository.

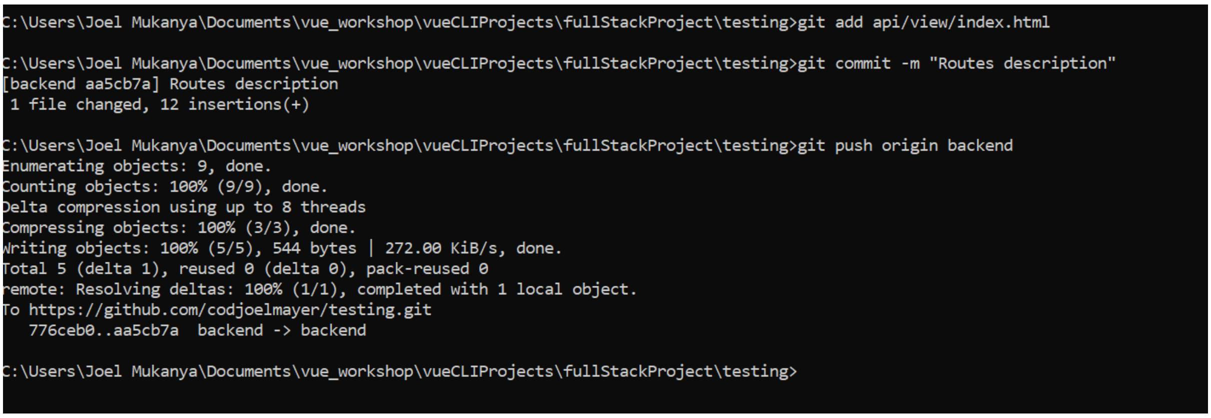


```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git status
On branch backend
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   api/package.json

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        api/package-lock.json

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>
```



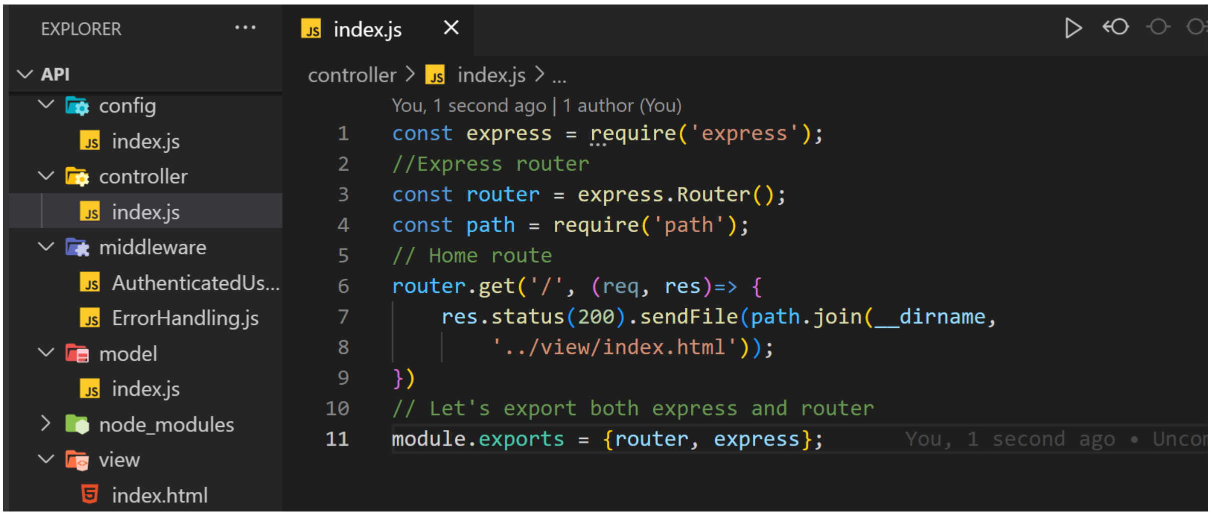
```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git add api/*.json
warning: LF will be replaced by CRLF in api/package.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in api/package-lock.json.
The file will have its original line endings in your working directory

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git commit -m "Packages are installed"
[backend 776ceb0] Packages are installed
 2 files changed, 1390 insertions(+), 1 deletion(-)
 create mode 100644 api/package-lock.json

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git push origin backend
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 14.95 KiB | 3.74 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/codjoelmayer/testing.git
   e080172..776ceb0  backend -> backend

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>
```

8. Let’s set routes description in a tabular format
   Let’s push the changes from view/index.html to GitHub



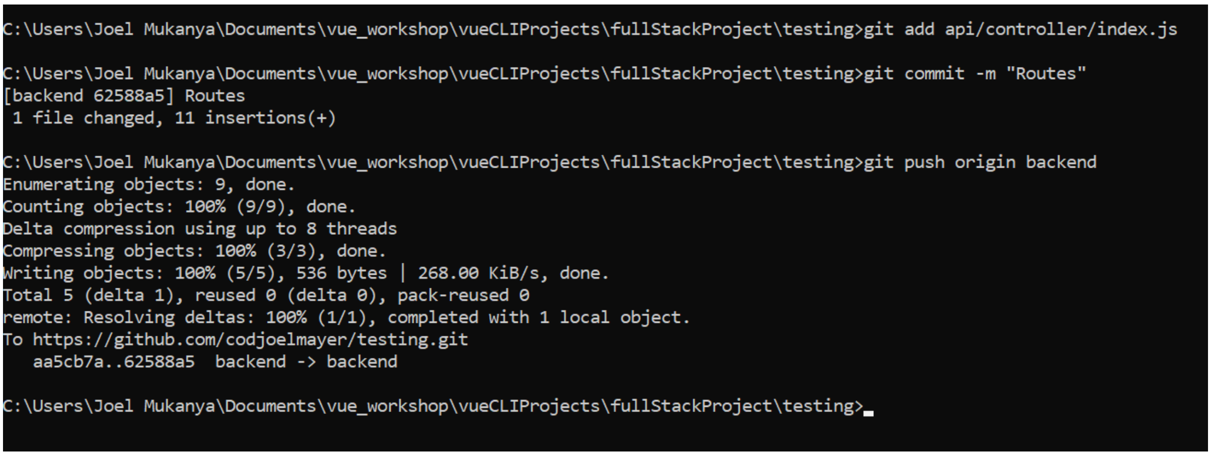
```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git add api/view/index.html

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git commit -m "Routes description"
[backend aa5cb7a] Routes description
 1 file changed, 12 insertions(+)

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git push origin backend
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 544 bytes | 272.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/codjoelmayer/testing.git
   776ceb0..aa5cb7a  backend -> backend

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>
```

9. Let’s set up the routes on our controller/index.js



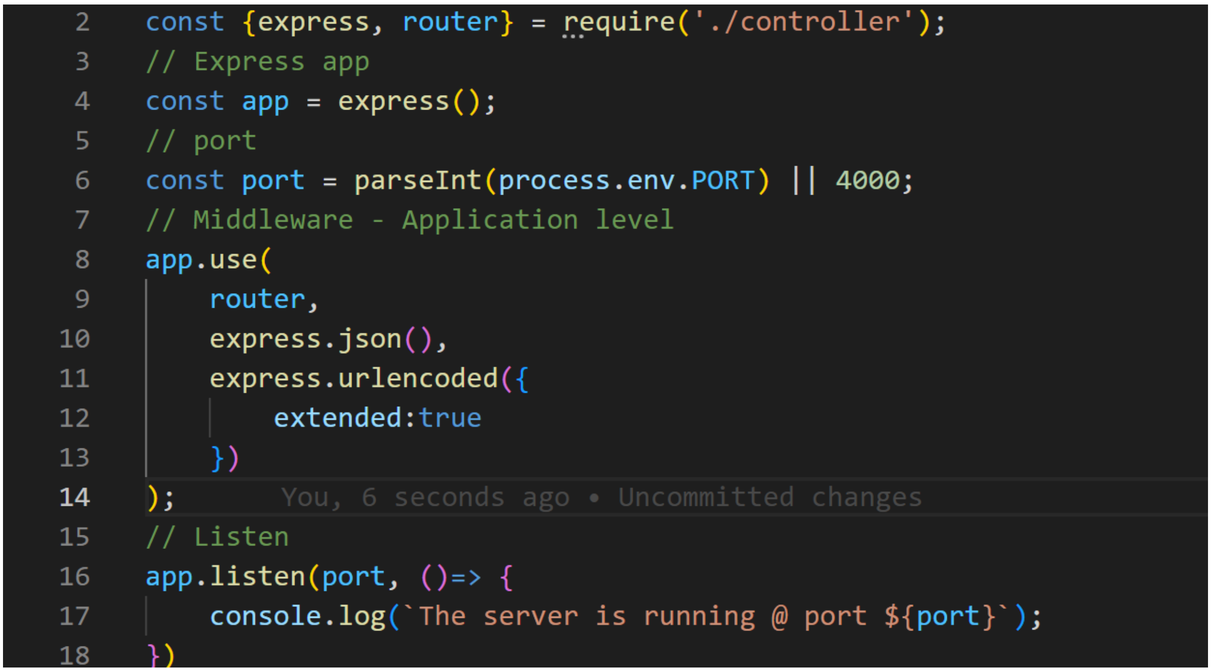
```js
const express = require('express');
//Express router
const router = express.Router();
const path = require('path');
// Home route
router.get('/', (req, res)=> {
    res.status(200).sendFile(path.join(__dirname,
        '../view/index.html'));
})
// Let's export both express and router
module.exports = {router, express};
```

Let’s push this change to GitHub



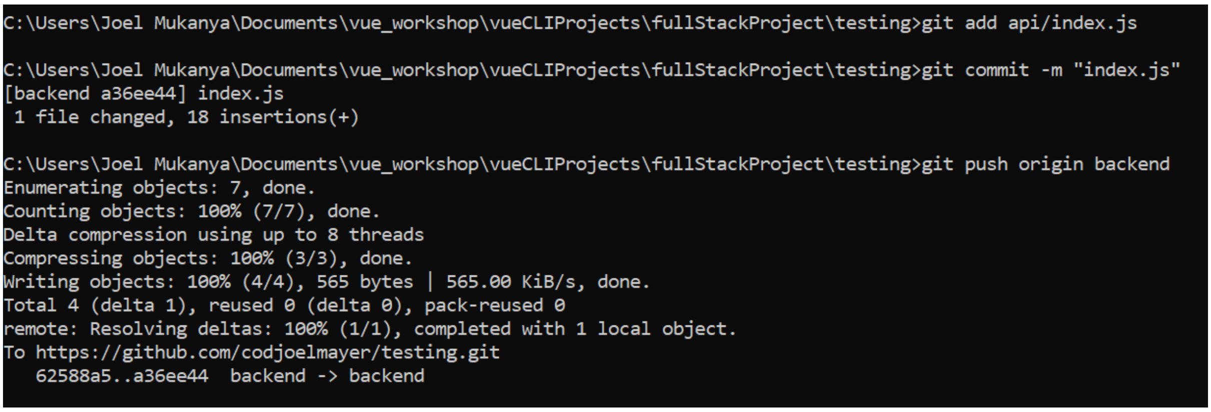
```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git add api/controller/index.js

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git commit -m "Routes"
[backend 62588a5] Routes
 1 file changed, 11 insertions(+)

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git push origin backend
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 536 bytes | 268.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/codjoelmayer/testing.git
   aa5cb7a..62588a5  backend -> backend

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>_
```

10. Let’s work on the index.js



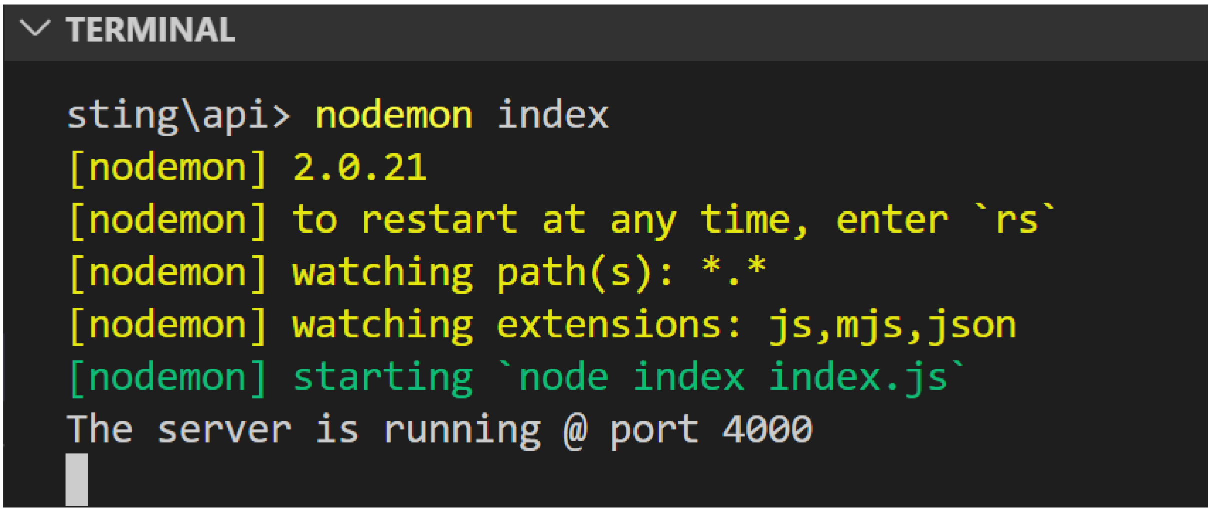
```
const {express, router} = require('./controller');
// Express app
const app = express();
// port
const port = parseInt(process.env.PORT) || 4000;
// Middleware - Application level
app.use(
    router,
    express.json(),
    express.urlencoded({
        extended:true
    })
);
// Listen
app.listen(port, ()=> {
    console.log(`The server is running @ port ${port}`);
})
```

Let’s push the change to GitHub

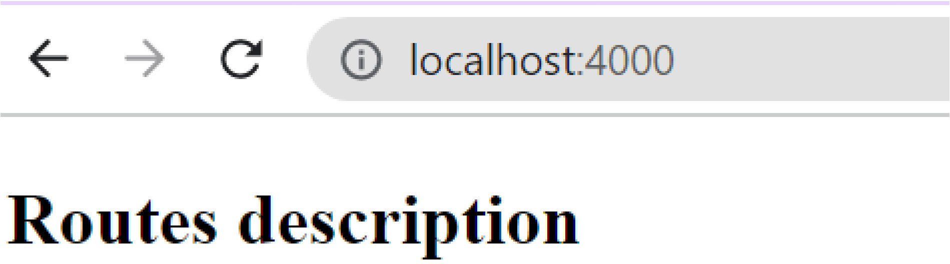


```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git add api/index.js

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git commit -m "index.js"
[backend a36ee44] index.js
 1 file changed, 18 insertions(+)

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git push origin backend
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 565 bytes | 565.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/codjoelmayer/testing.git
   62588a5..a36ee44  backend -> backend
```

Let’s test our program

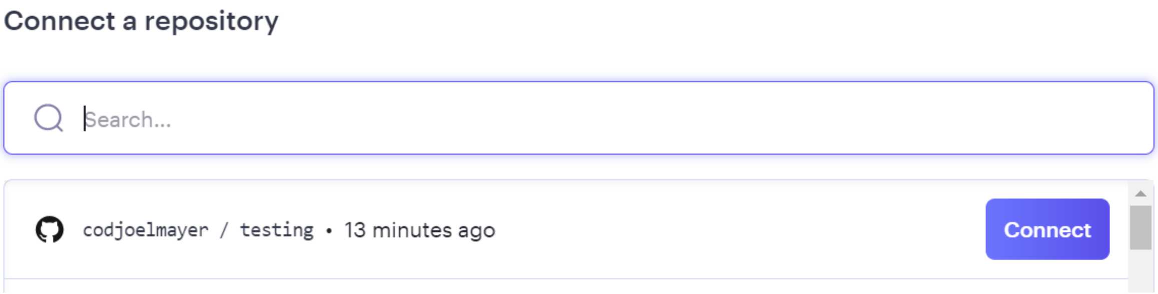


```
sting\api> nodemon index
[nodemon] 2.0.21
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index index.js`
The server is running @ port 4000
```


localhost:4000

Routes description

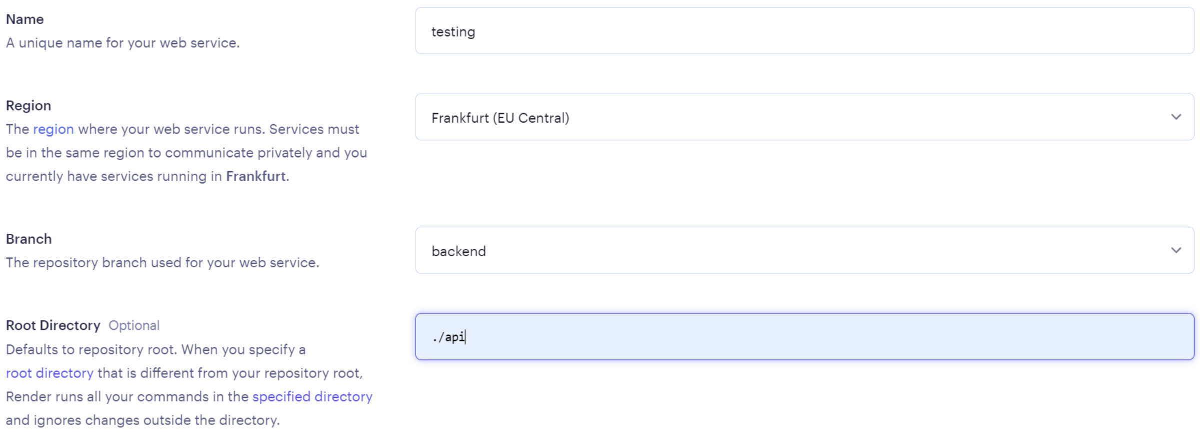
Let’s deploy our backend to render.com
Create a new web service
Connect to your cloned repository


Connect a repository

Search...

codjoelmayer / testing • 13 minutes ago

Since our api is under the api directory then we must specify the source.



Name
A unique name for your web service.

testing

Region
The region where your web service runs. Services must be in the same region to communicate privately and you currently have services running in **Frankfurt**.

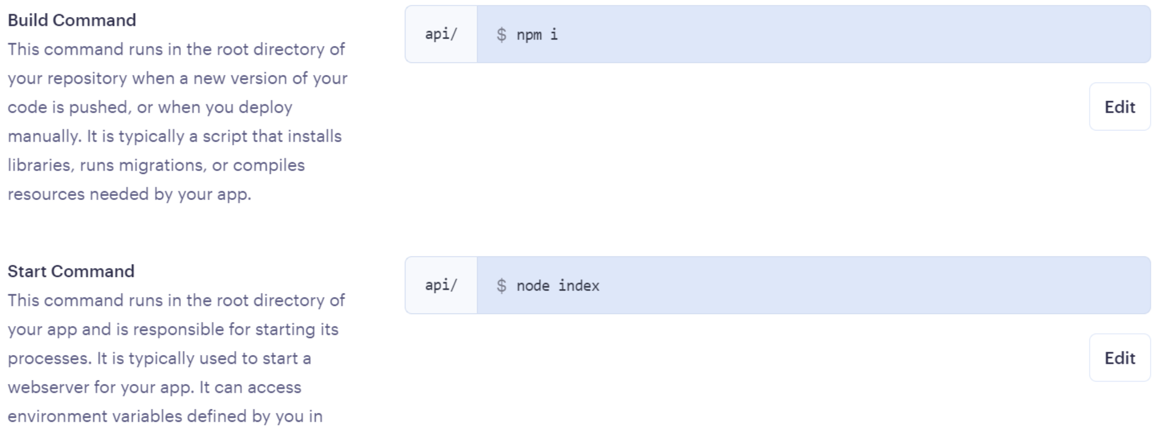
Frankfurt (EU Central)

Branch
The repository branch used for your web service.

backend

Root Directory Optional
Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.

./api



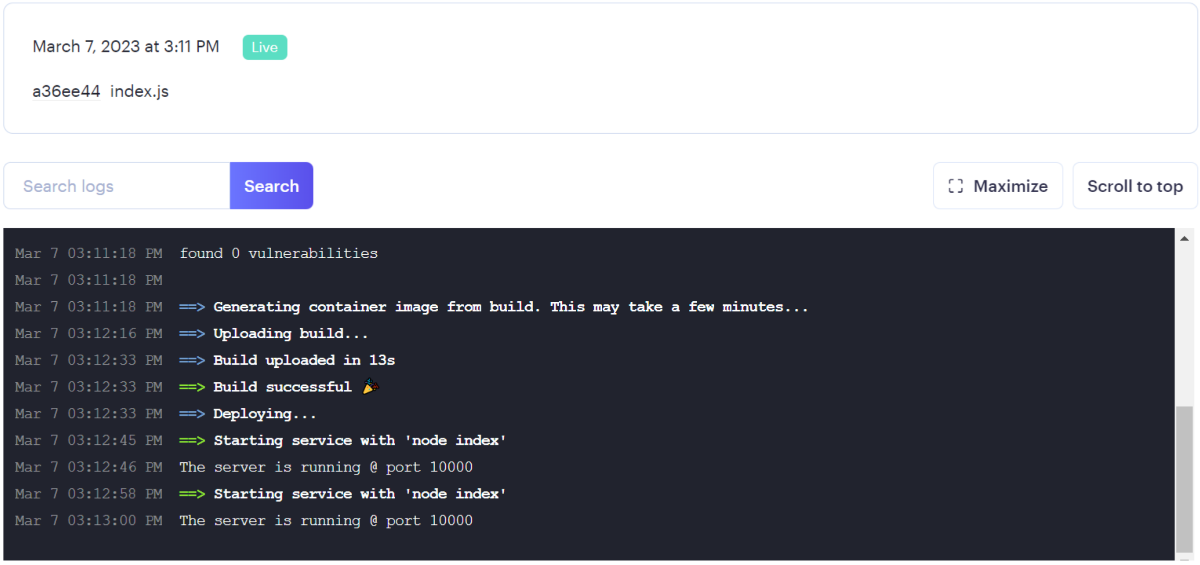
Build Command
This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

```
api/   $ npm i
```

Start Command
This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in

```
api/   $ node index
```

Click on the “Advanced” button to specify all your .env variables or .env files.
Since mine is not connected to the database, I don’t have to set it up.
Finally, I have to create a service.



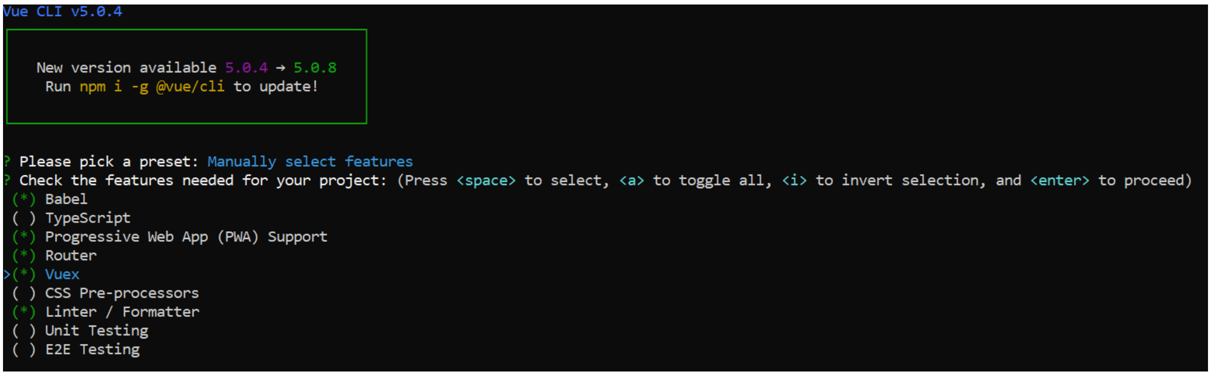
March 7, 2023 at 3:11 PM Live

a36ee44 index.js

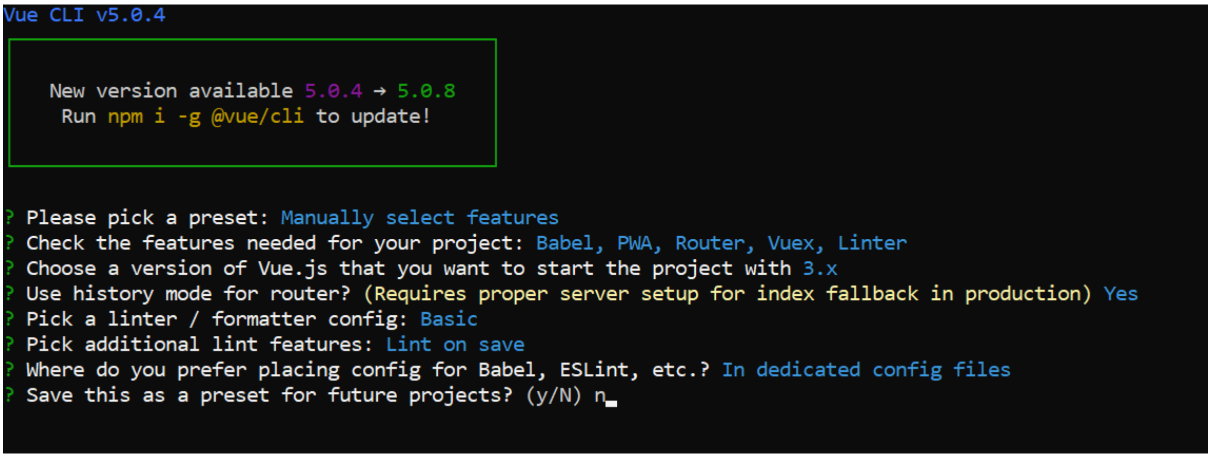
```
Mar 7 03:11:18 PM  found 0 vulnerabilities
Mar 7 03:11:18 PM
Mar 7 03:11:18 PM  ==> Generating container image from build. This may take a few minutes...
Mar 7 03:12:16 PM  ==> Uploading build...
Mar 7 03:12:33 PM  ==> Build uploaded in 13s
Mar 7 03:12:33 PM  ==> Build successful 🎉
Mar 7 03:12:33 PM  ==> Deploying...
Mar 7 03:12:45 PM  ==> Starting service with 'node index'
Mar 7 03:12:46 PM  The server is running @ port 10000
Mar 7 03:12:58 PM  ==> Starting service with 'node index'
Mar 7 03:13:00 PM  The server is running @ port 10000
```

Let’s work on our frontend
We still on the testing directory

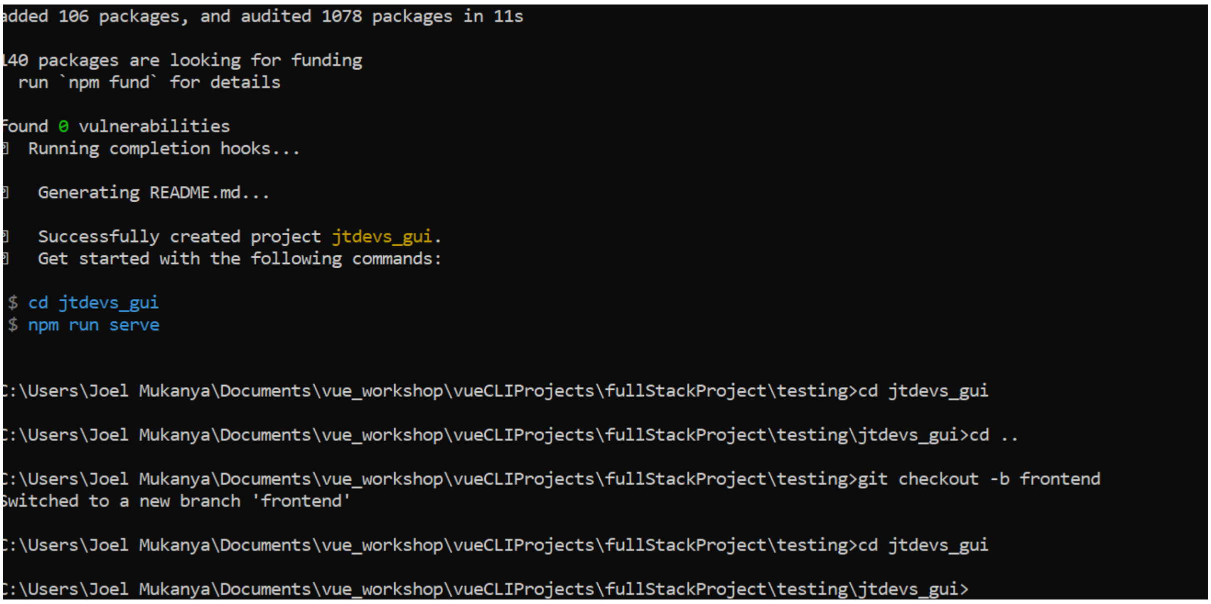
1. vue create jtdevs_gui



```
Vue CLI v5.0.4
┌─────────────────────────────────────────┐
│                                         │
│   New version available 5.0.4 → 5.0.8   │
│    Run npm i -g @vue/cli to update!     │
│                                         │
└─────────────────────────────────────────┘

? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
 (*) Babel
 ( ) TypeScript
 (*) Progressive Web App (PWA) Support
 (*) Router
>(*) Vuex
 ( ) CSS Pre-processors
 (*) Linter / Formatter
 ( ) Unit Testing
 ( ) E2E Testing
```

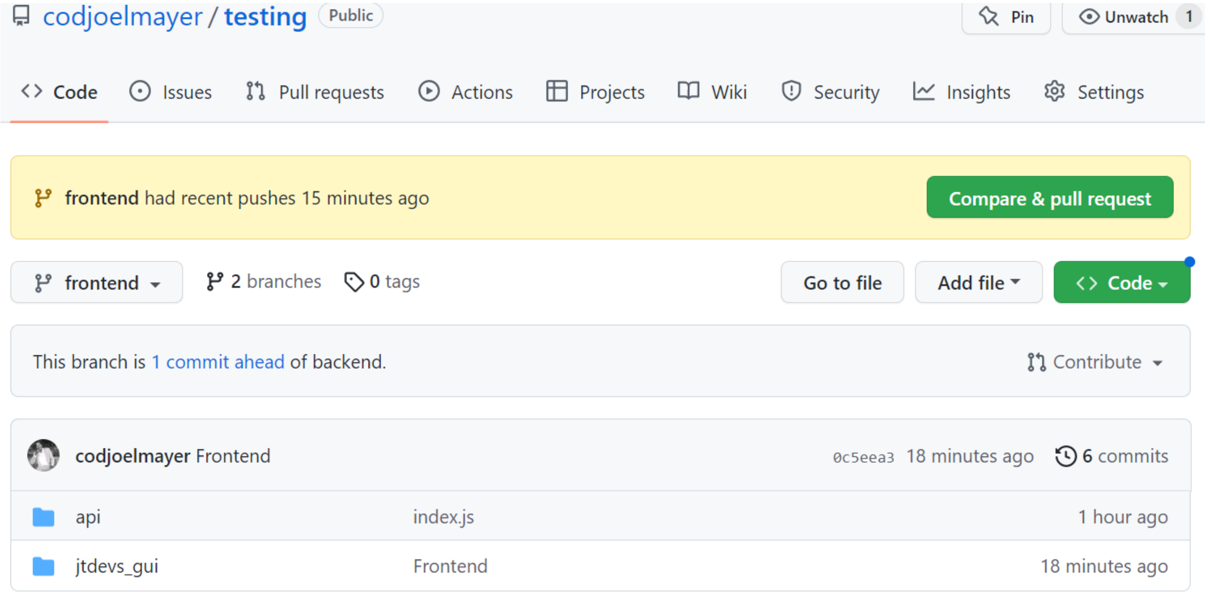


```
Vue CLI v5.0.4
┌─────────────────────────────────────────┐
│                                         │
│   New version available 5.0.4 → 5.0.8   │
│    Run npm i -g @vue/cli to update!     │
│                                         │
└─────────────────────────────────────────┘

? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, PWA, Router, Vuex, Linter
? Choose a version of Vue.js that you want to start the project with 3.x
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? (y/N) n
```



```
added 106 packages, and audited 1078 packages in 11s

140 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
▯  Running completion hooks...

▯   Generating README.md...

▯   Successfully created project jtdevs_gui.
▯   Get started with the following commands:

 $ cd jtdevs_gui
 $ npm run serve


C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>cd jtdevs_gui

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\jtdevs_gui>cd ..

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>git checkout -b frontend
Switched to a new branch 'frontend'

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing>cd jtdevs_gui

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\jtdevs_gui>
```

2. Under our testing directory or cloned directory
   Create a frontend branch
   Let's push our VueJS to frontend branch
   We have to run the following commands:
   git add jtdevs_gui
   git commit -m "Frontend"
   git push origin frontend



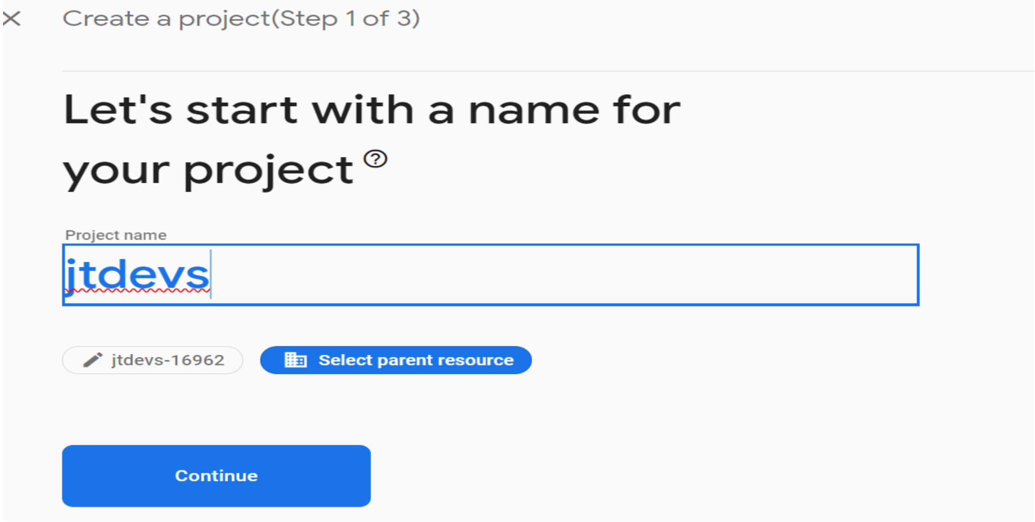
frontend had recent pushes 15 minutes ago

This branch is 1 commit ahead of backend.

codjoelmayer Frontend 0c5eea3 18 minutes ago 6 commits

api	index.js	1 hour ago
jtdevs_gui	Frontend	18 minutes ago

3. Let's deploy our VueJs on Firebase
   Go to https://console.firebase.google.com/
   Create a new project

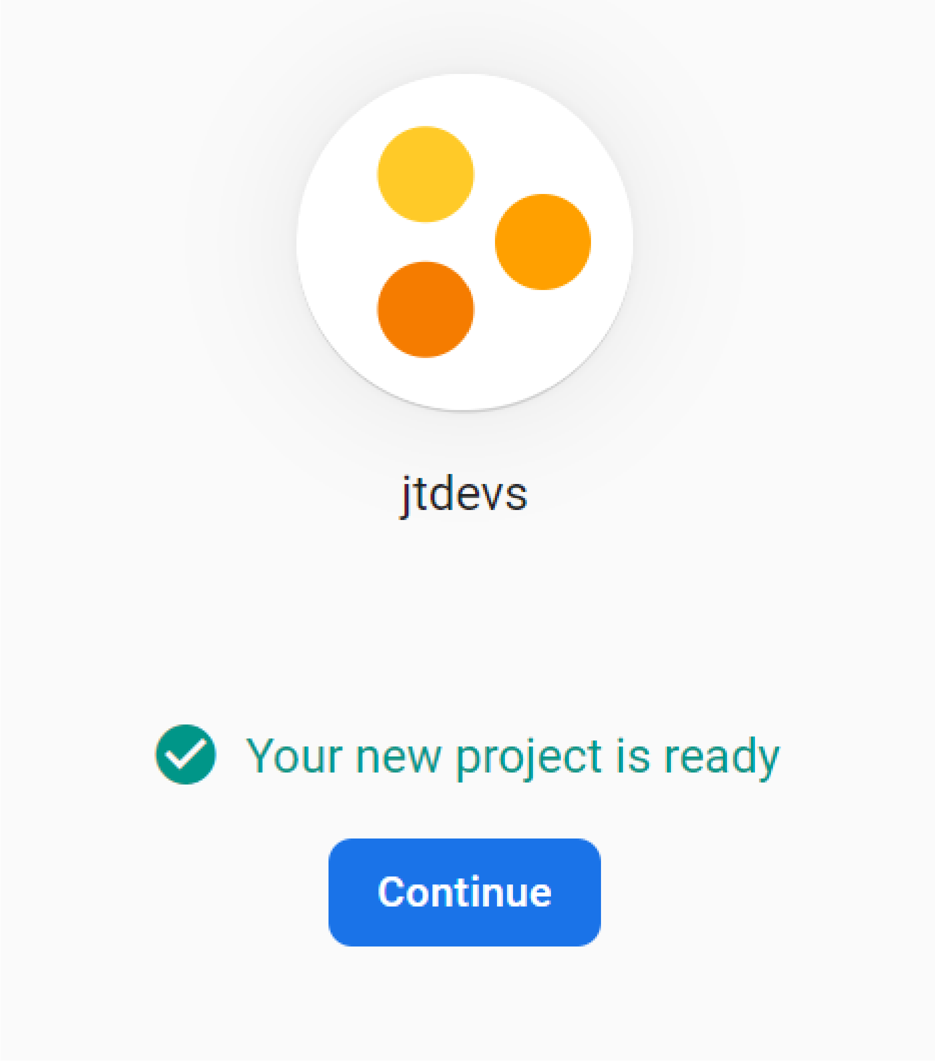


Let's start with a name for your project

Project name: jtdevs

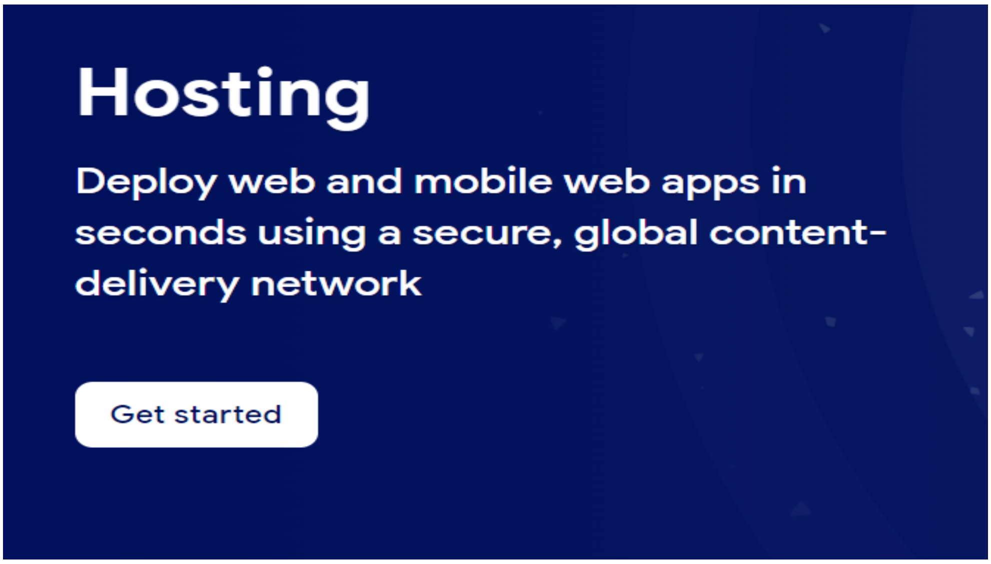
jtdevs-16962

   Click on "Create a project" button


jtdevs

Your new project is ready

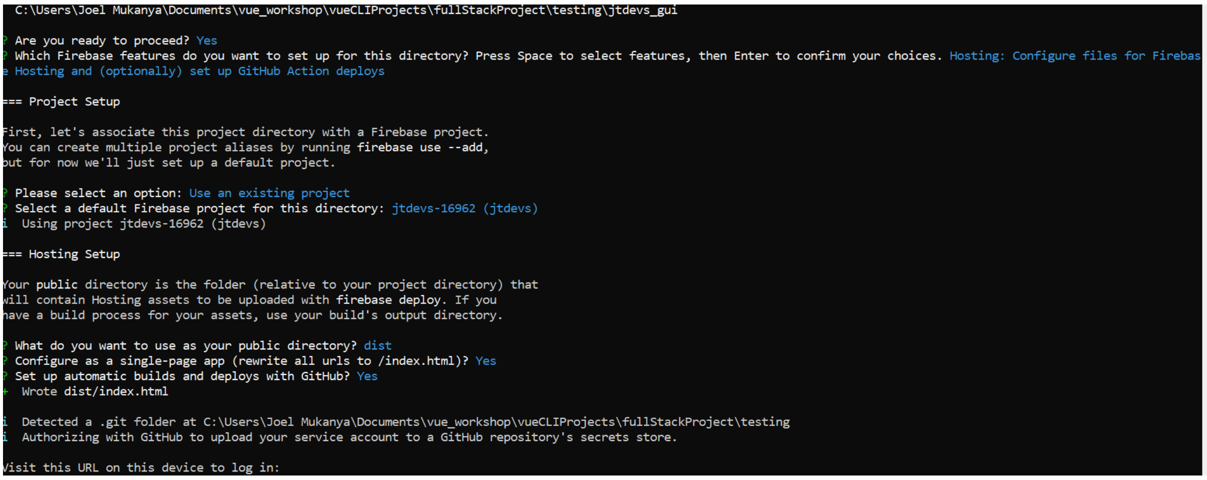
Under your product categories click on **build then hosting “Get started”**


Hosting

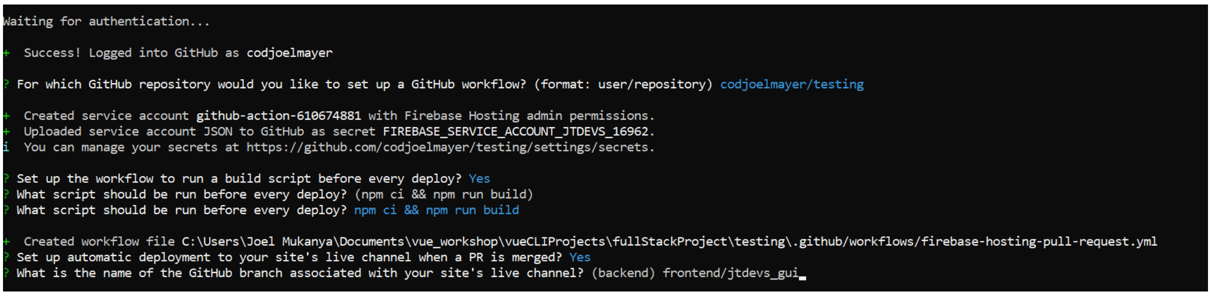
Deploy web and mobile web apps in seconds using a secure, global content-delivery network

NB: Before you can run the below commands, please make sure you are in the VueJS directory

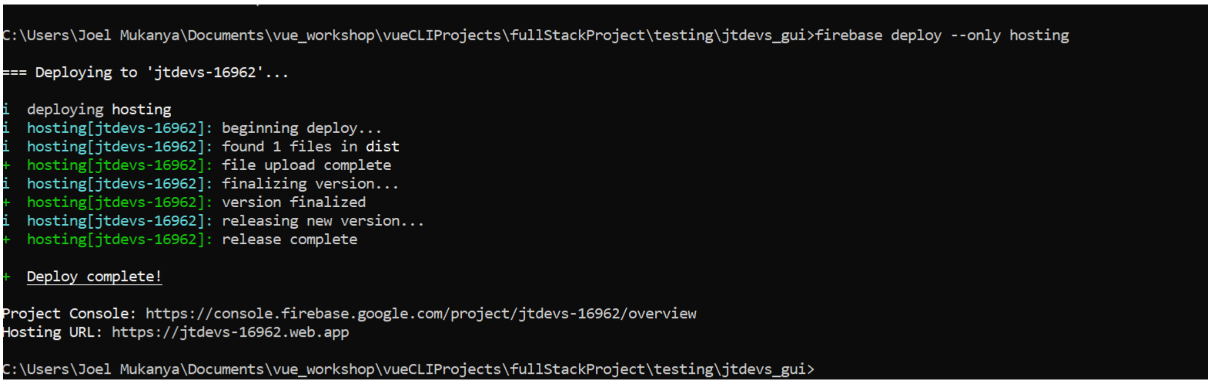
4. Run the following commands
   npm install -g firebase-tools
   npm run build
   firebase login
   firebase init
   Answer the following questions
   - Are you ready to proceed: y
   - Select: Hostings: configure files for Firebase hosting and (optionally) set up GitHub
   - Use an existing project
   - Select your created app
   - What do you want to use as your public directory? (public) dist
   - Configure as a single-page app (rewrite all urls to /index.html)? y



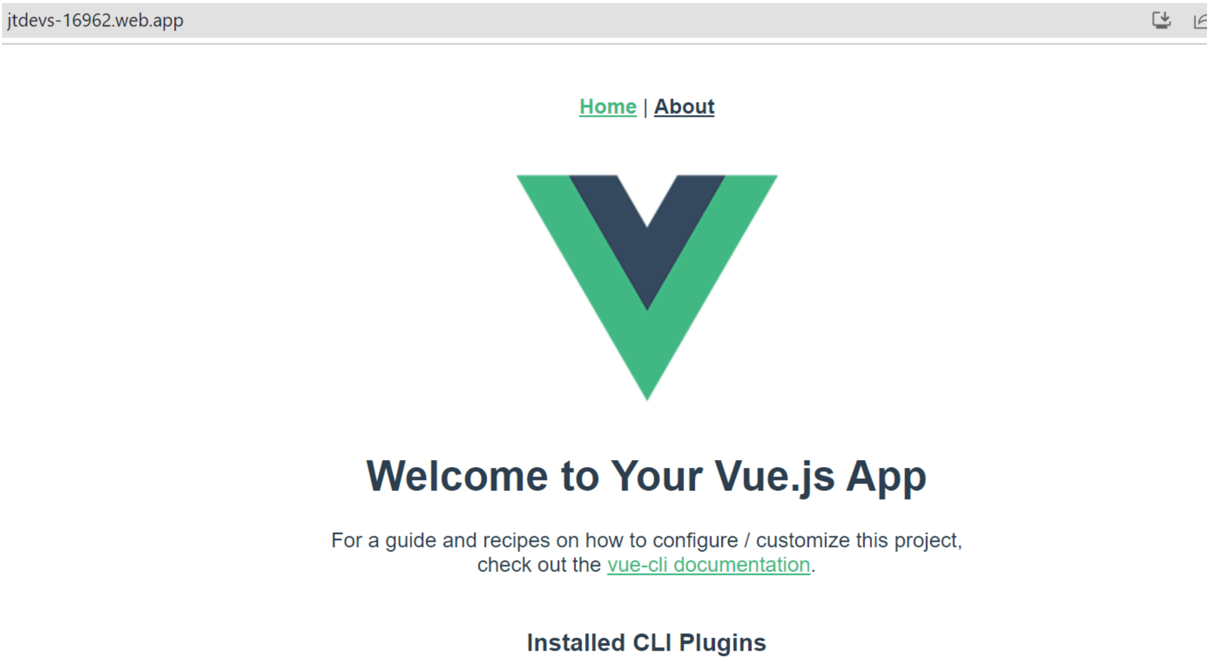
```
 C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\jtdevs_gui

? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. Hosting: Configure files for Firebas
e Hosting and (optionally) set up GitHub Action deploys

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory: jtdevs-16962 (jtdevs)
i  Using project jtdevs-16962 (jtdevs)

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? dist
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? Yes
+  Wrote dist/index.html

i  Detected a .git folder at C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing
i  Authorizing with GitHub to upload your service account to a GitHub repository's secrets store.

Visit this URL on this device to log in:
```



```
Waiting for authentication...

+  Success! Logged into GitHub as codjoelmayer

? For which GitHub repository would you like to set up a GitHub workflow? (format: user/repository) codjoelmayer/testing

+  Created service account github-action-610674881 with Firebase Hosting admin permissions.
+  Uploaded service account JSON to GitHub as secret FIREBASE_SERVICE_ACCOUNT_JTDEVS_16962.
i  You can manage your secrets at https://github.com/codjoelmayer/testing/settings/secrets.

? Set up the workflow to run a build script before every deploy? Yes
? What script should be run before every deploy? (npm ci && npm run build)
? What script should be run before every deploy? npm ci && npm run build

+  Created workflow file C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\.github/workflows/firebase-hosting-pull-request.yml
? Set up automatic deployment to your site's live channel when a PR is merged? Yes
? What is the name of the GitHub branch associated with your site's live channel? (backend) frontend/jtdevs_gui
```



```
C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\jtdevs_gui>firebase deploy --only hosting

=== Deploying to 'jtdevs-16962'...

i  deploying hosting
i  hosting[jtdevs-16962]: beginning deploy...
i  hosting[jtdevs-16962]: found 1 files in dist
+  hosting[jtdevs-16962]: file upload complete
i  hosting[jtdevs-16962]: finalizing version...
+  hosting[jtdevs-16962]: version finalized
i  hosting[jtdevs-16962]: releasing new version...
+  hosting[jtdevs-16962]: release complete

+  Deploy complete!

Project Console: https://console.firebase.google.com/project/jtdevs-16962/overview
Hosting URL: https://jtdevs-16962.web.app

C:\Users\Joel Mukanya\Documents\vue_workshop\vueCLIProjects\fullStackProject\testing\jtdevs_gui>
```



Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project, check out the vue-cli documentation.

Installed CLI Plugins

NB:In case it point to the wrong project, please run **firebase use projectID**