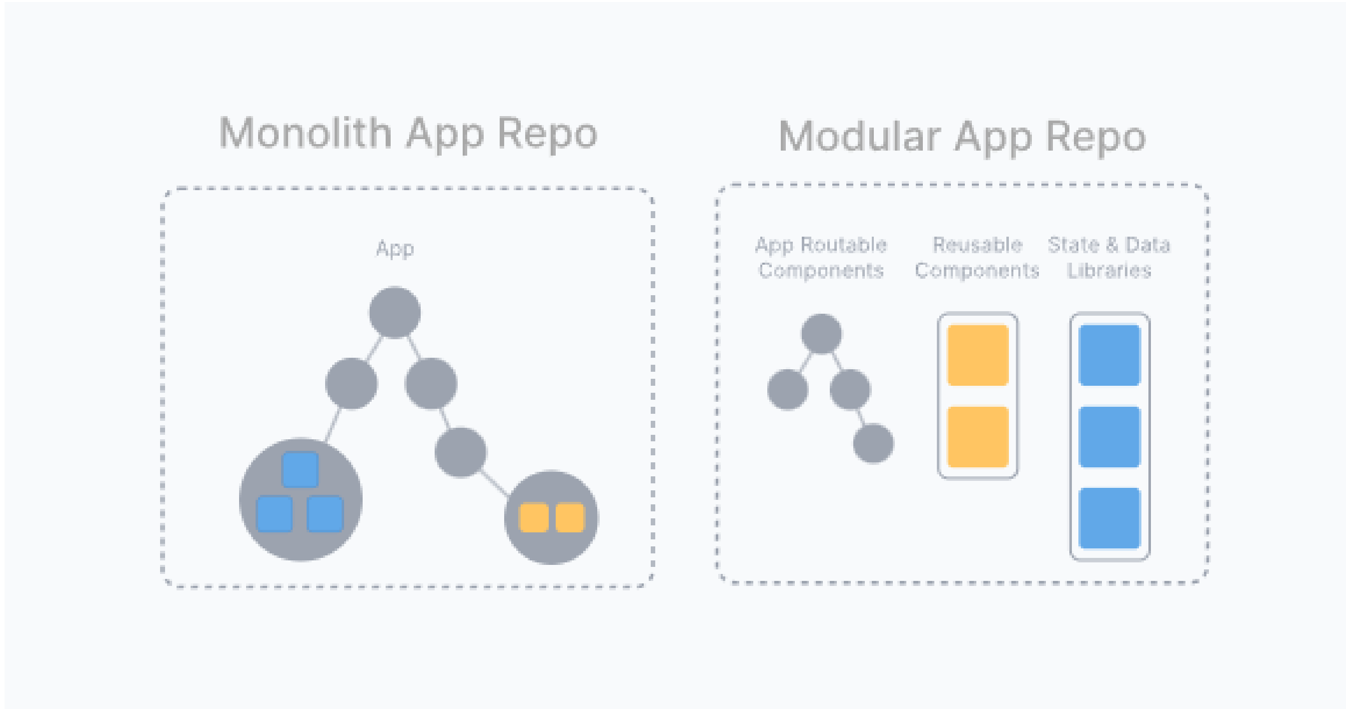
JavaScript Monorepo Using Lerna

Theethawat Savastham (Tin)
The Duck Creator & Intelligent Automation Research Center
theethawat.sa@one.th

Theethawat/>
Savastham

When a Project Need to scale

- When the backend server has too much logic and tasks that the REST API server cannot handle.
- Microservice is needed



Monorepo

- Mono repository but not Monolith
- Many Packages in one repository
- Fit for not very large and not small projects
- Central Library, Central Logic = less duplicate code

SONGKHLA

Monorepo Tools

- Visit monorepo.tools

Bazel

Gradle

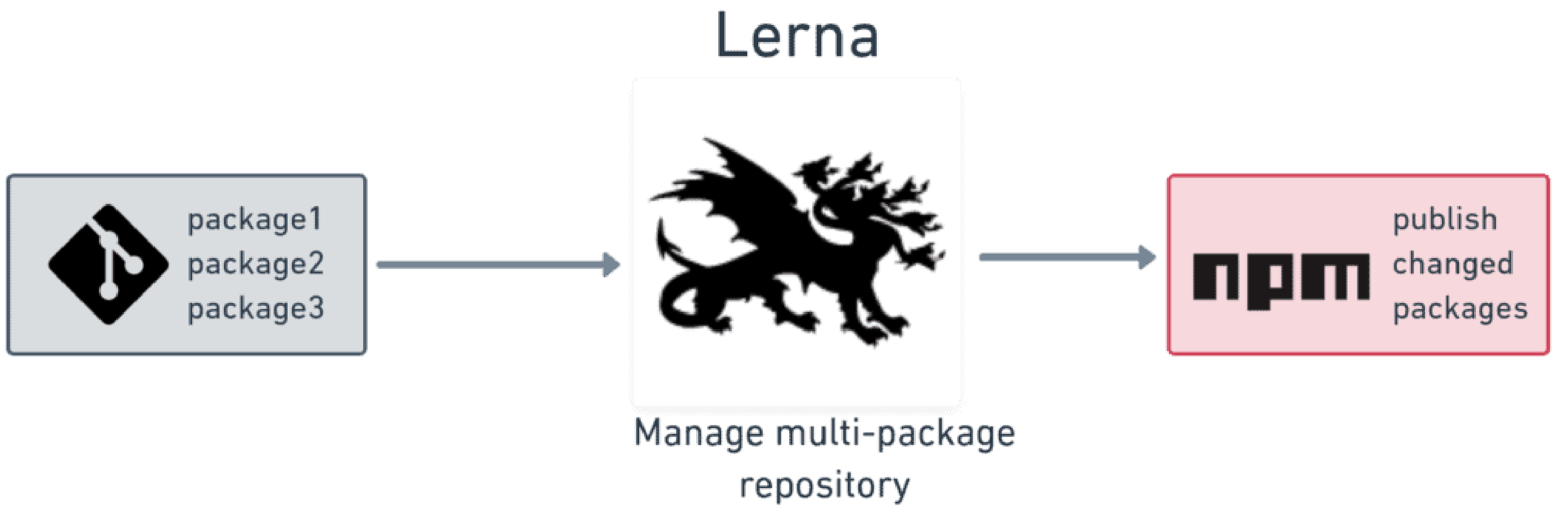
Lerna

Nx

rush

Lerna

- Fast, modern build system for managing and publishing multiple JavaScript/TypeScript packages from the same repository.
- More info at lerna.js.org

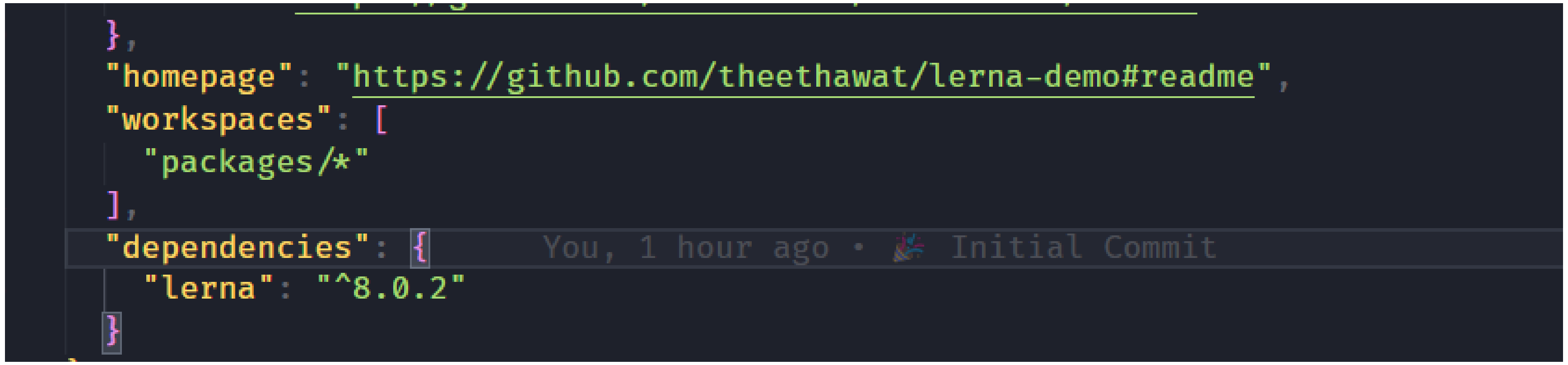


Demo on Lerna

- The Very basic Application from scratch using React, Express with together use constant library
- Final demo code can be found at GitHub theethawat/lerna-demo

Initial Project

- Initial your project with `npm init -y` and then `npx lerna init`
- Create folder `packages/` to store the application
- Checking in `package.json` at must include workspaces key

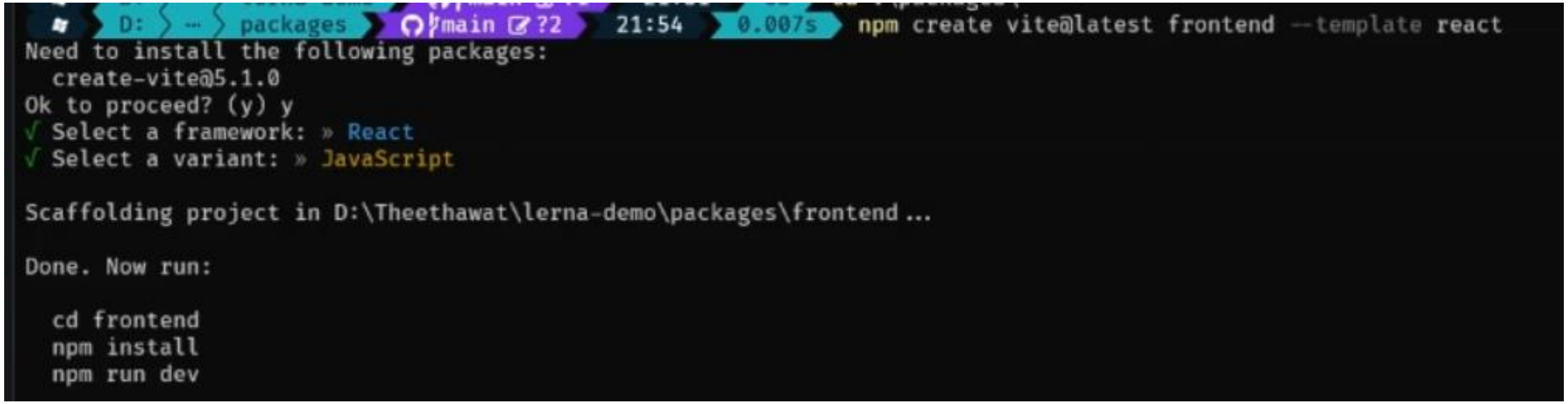


```
  },
  "homepage": "https://github.com/theethawat/lerna-demo#readme",
  "workspaces": [
    "packages/*"
  ],
  "dependencies": {        You, 1 hour ago • 🎉 Initial Commit
    "lerna": "^8.0.2"
  }
```

Theethawat/>
Savastham

Create Basic Application Inside

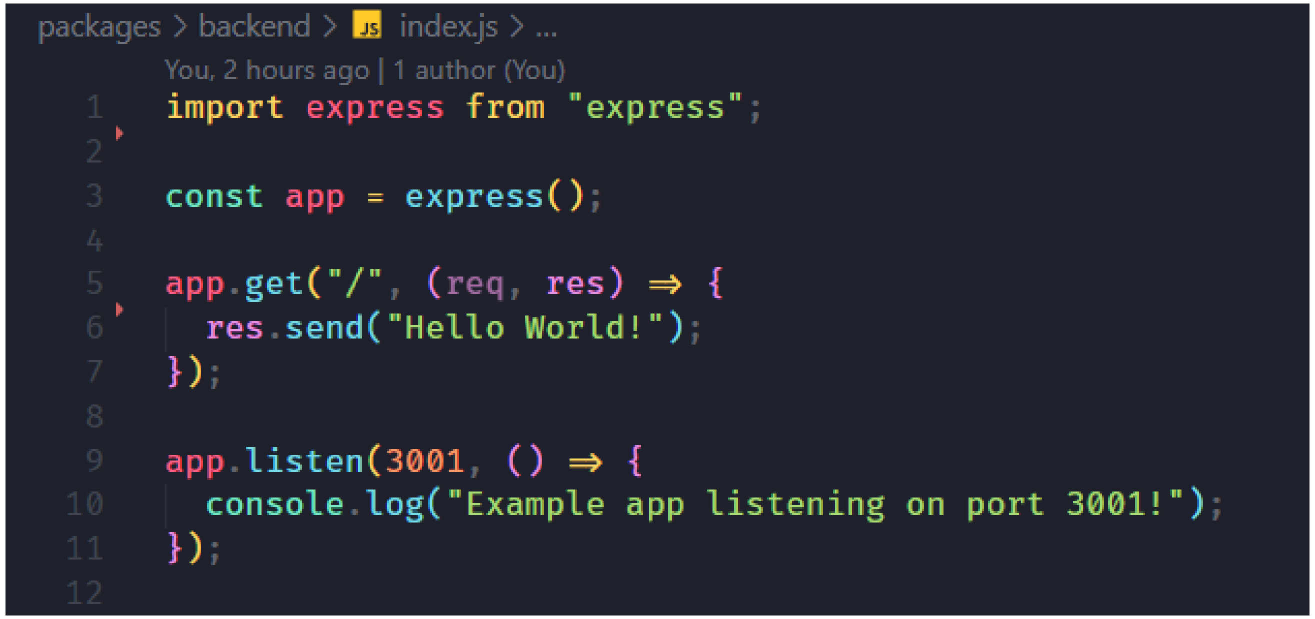
- Change directory into packages and create basic react app using vite
  - `npm create vite@latest frontend –template react`
  - `npm install`



```
 D: > … > packages > main ?2  21:54  0.007s  npm create vite@latest frontend --template react
Need to install the following packages:
  create-vite@5.1.0
Ok to proceed? (y) y
√ Select a framework: » React
√ Select a variant: » JavaScript

Scaffolding project in D:\Theethawat\lerna-demo\packages\frontend ...

Done. Now run:

  cd frontend
  npm install
  npm run dev
```

Create Basic Application Inside (Cont.)

- Create Node.js Express app on `packages/` create backend directory and initial with `npm init -y` and Create `index.js` file
- Install express and nodemon as dependencies using `npm install`

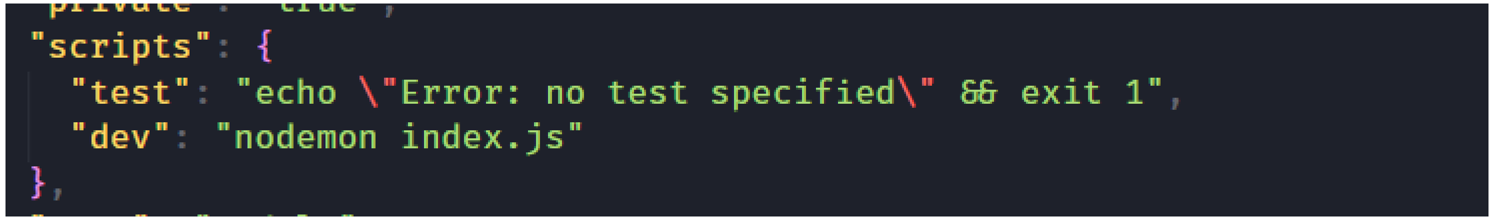

packages > backend > index.js > ...
You, 2 hours ago | 1 author (You)

```js
import express from "express";

const app = express();

app.get("/", (req, res) => {
  res.send("Hello World!");
});

app.listen(3001, () => {
  console.log("Example app listening on port 3001!");
});
```

Theethawat/>
Savastham

Create Running Script

- For frontend, Vite will create `dev` running script in `package.json`
- For backend, create dev script to running from nodemon

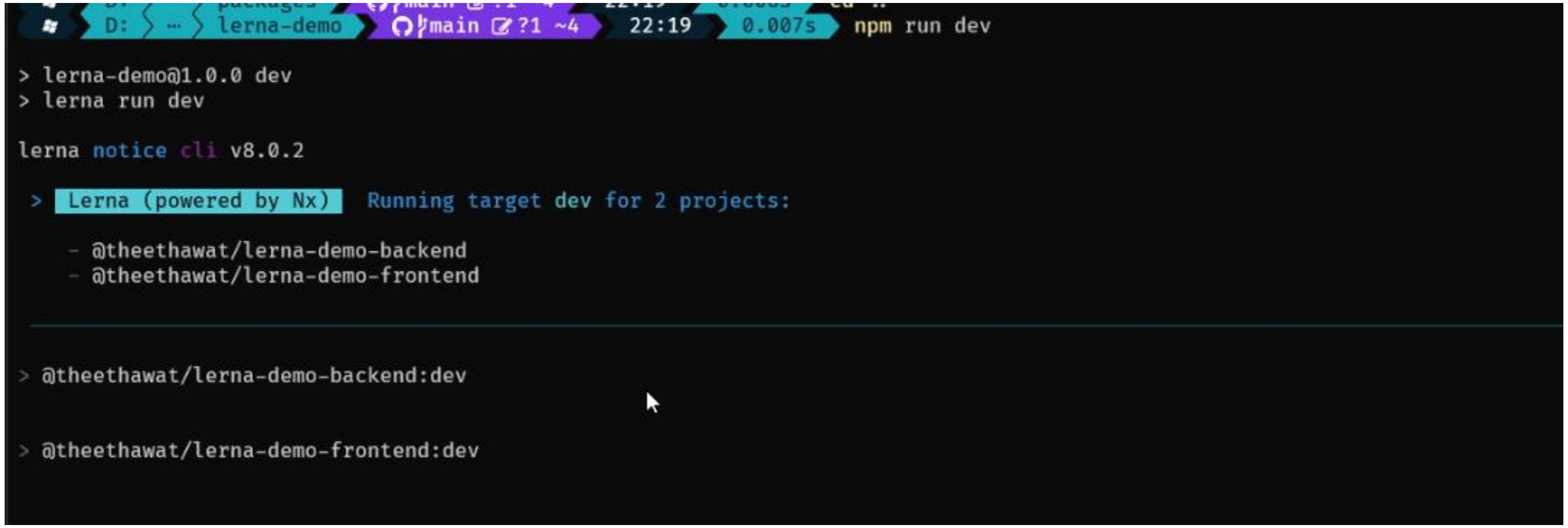


```
"private": true,
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "dev": "nodemon index.js"
},
```

- So both frontend and backend both have an dev script

Create Running Script (Cont.)

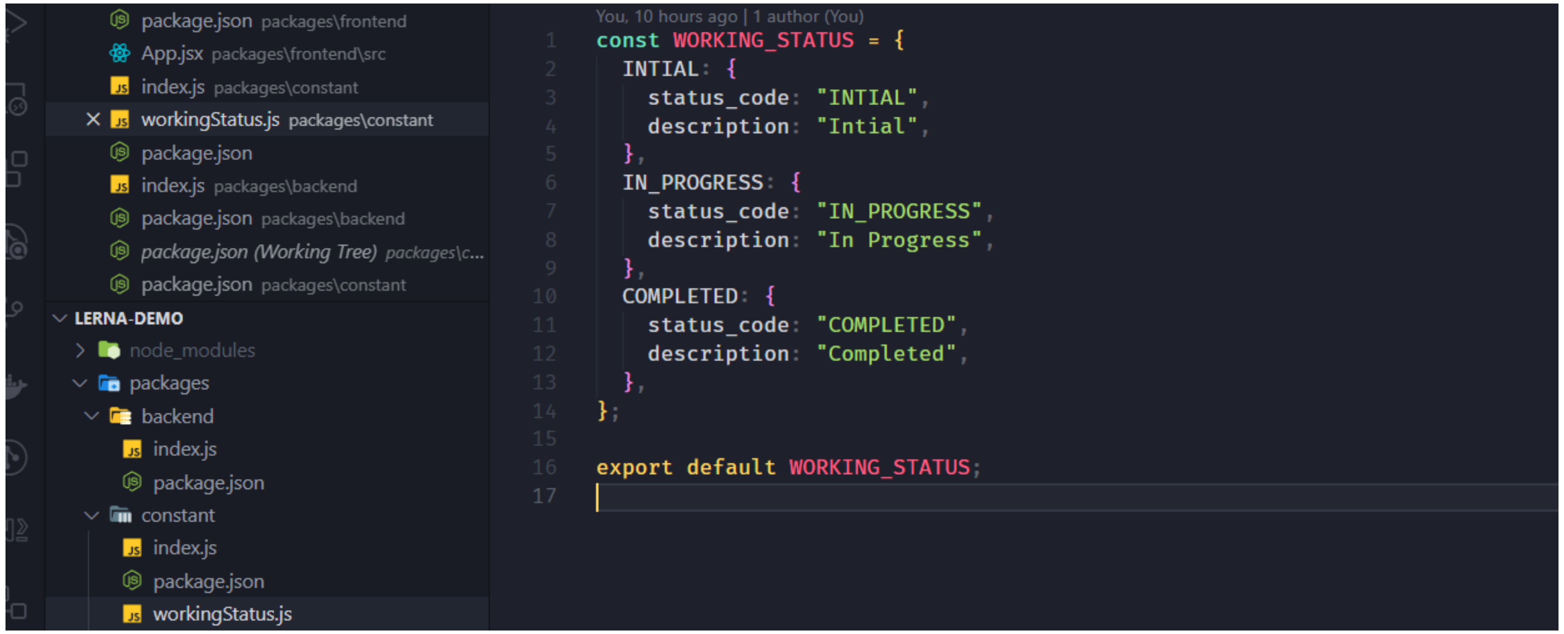
- Adding `dev` script on root project package.json add `lerna run dev`
- This command will run all inside projects with `npm run dev` command



```
 D: 〉 … 〉 lerna-demo   main ?1 ~4   22:19   0.007s   npm run dev

> lerna-demo@1.0.0 dev
> lerna run dev

lerna notice cli v8.0.2

 >  Lerna (powered by Nx)   Running target dev for 2 projects:

    - @theethawat/lerna-demo-backend
    - @theethawat/lerna-demo-frontend

 ————————————————————————————————————————————————————————————

> @theethawat/lerna-demo-backend:dev


> @theethawat/lerna-demo-frontend:dev
```

Initial the constant package

- Create constant/ under folder packages/
- Initial with `npm init –y`
- Create first constant file



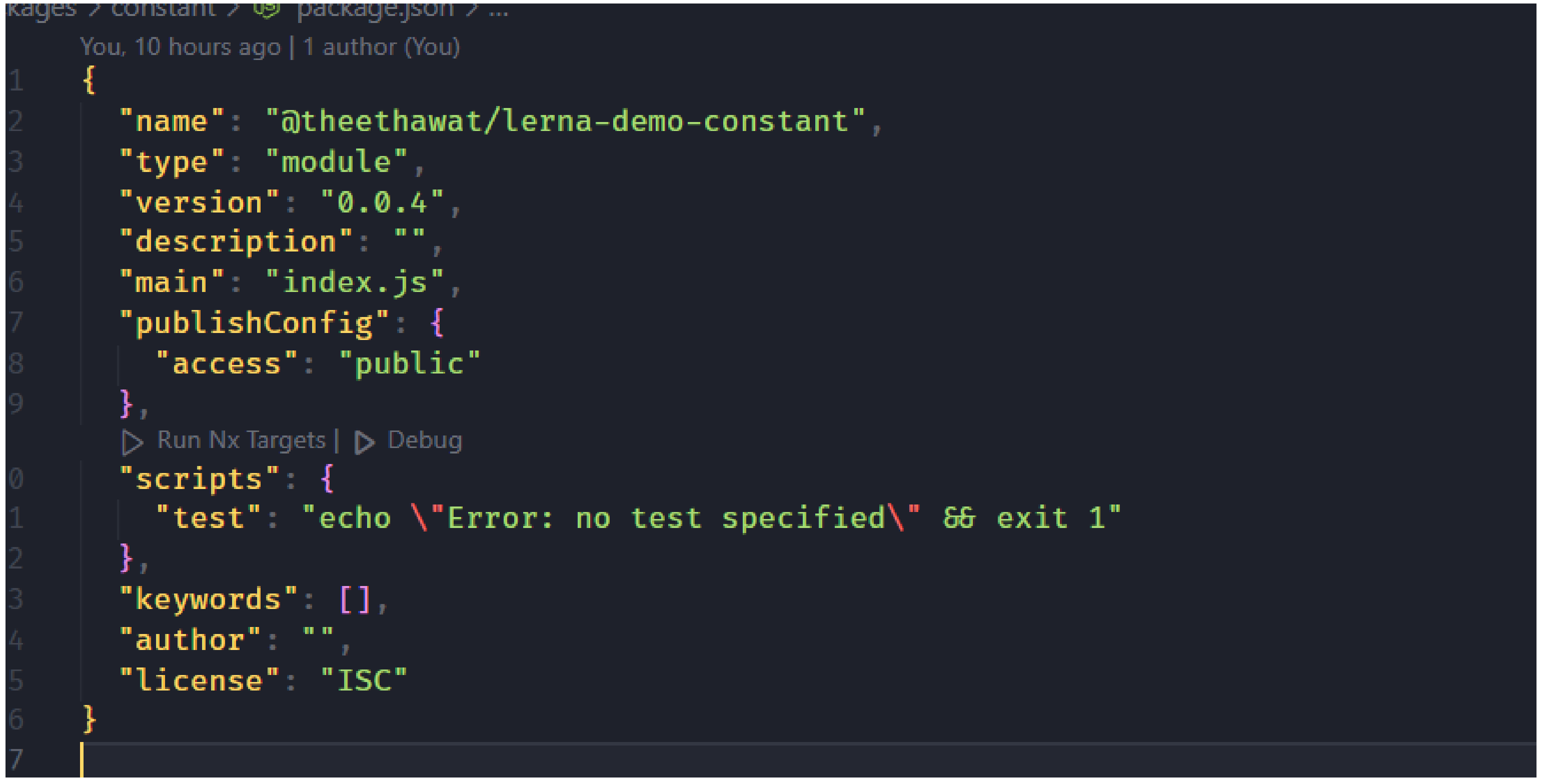
```
package.json packages\frontend
App.jsx packages\frontend\src
index.js packages\constant
× workingStatus.js packages\constant
package.json
index.js packages\backend
package.json packages\backend
package.json (Working Tree) packages\c...
package.json packages\constant
LERNA-DEMO
  > node_modules
  v packages
    v backend
        index.js
        package.json
    v constant
        index.js
        package.json
        workingStatus.js
```

```
You, 10 hours ago | 1 author (You)
const WORKING_STATUS = {
  INTIAL: {
    status_code: "INTIAL",
    description: "Intial",
  },
  IN_PROGRESS: {
    status_code: "IN_PROGRESS",
    description: "In Progress",
  },
  COMPLETED: {
    status_code: "COMPLETED",
    description: "Completed",
  },
};

export default WORKING_STATUS;
```

Theethawat/>
Savastham

Customize Package Info

- This Info will be our package info in npm registry when we publish



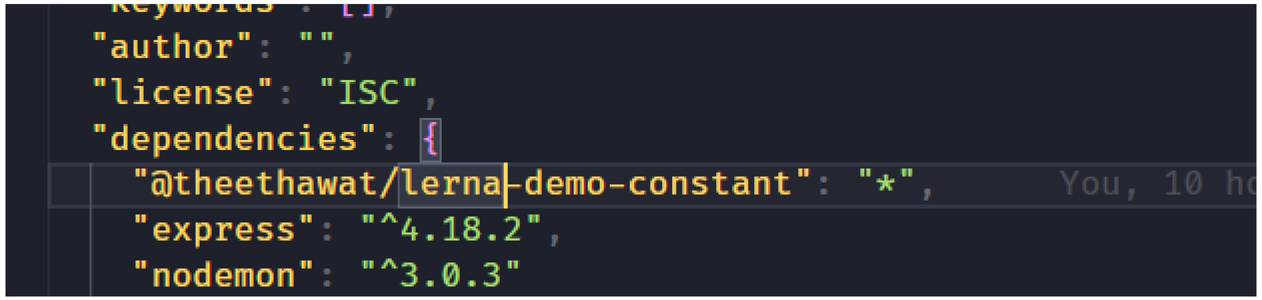
kages > constant > package.json > ...
You, 10 hours ago | 1 author (You)
Run Nx Targets | Debug

```
{
  "name": "@theethawat/lerna-demo-constant",
  "type": "module",
  "version": "0.0.4",
  "description": "",
  "main": "index.js",
  "publishConfig": {
    "access": "public"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Theethawat/>
Savastham

Import our package

- Import constant package to our backend

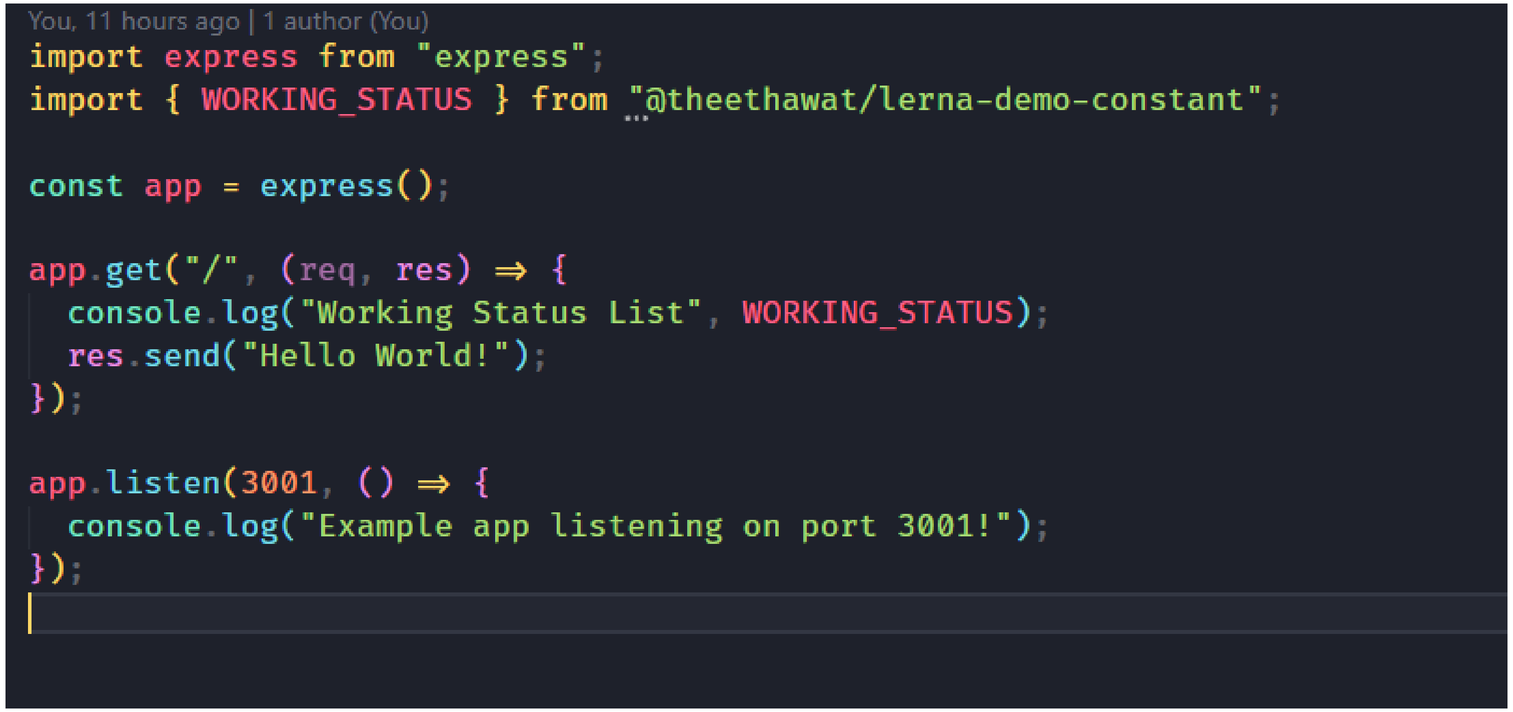


```
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@theethawat/lerna-demo-constant": "*",        You, 10 ho
    "express": "^4.18.2",
    "nodemon": "^3.0.3"
```

- Running `npm install` on root project and rerun again

Import our package (cont.)

- Now on we can use our package in our development, lerna will manage for us

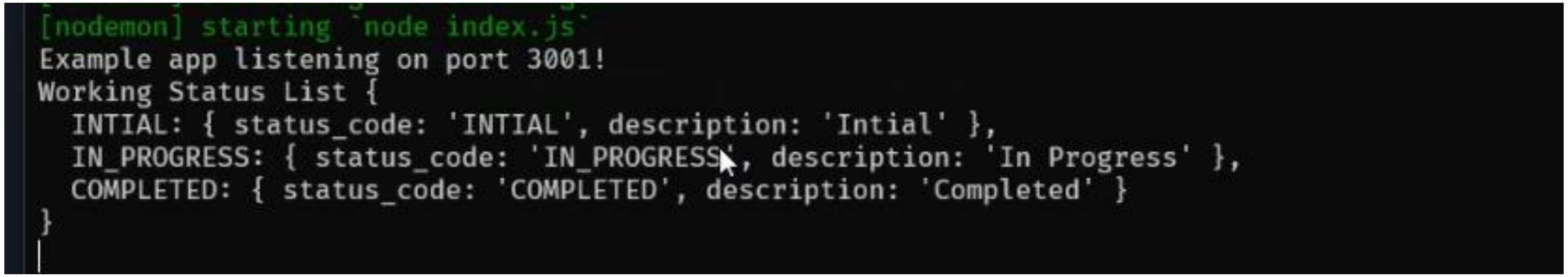


```
You, 11 hours ago | 1 author (You)
import express from "express";
import { WORKING_STATUS } from "@theethawat/lerna-demo-constant";

const app = express();

app.get("/", (req, res) => {
  console.log("Working Status List", WORKING_STATUS);
  res.send("Hello World!");
});

app.listen(3001, () => {
  console.log("Example app listening on port 3001!");
});
```

Theethawat/>
Savastham

Import our package (cont.)

- The Result when code is run and open for localhost:3001



```
[nodemon] starting `node index.js`
Example app listening on port 3001!
Working Status List {
  INTIAL: { status_code: 'INTIAL', description: 'Intial' },
  IN_PROGRESS: { status_code: 'IN_PROGRESS', description: 'In Progress' },
  COMPLETED: { status_code: 'COMPLETED', description: 'Completed' }
}
```

- We can use this concept on frontend packages too

Package Publishing

- Adding Script `lerna publish --no-private` in root project package.json
- Make repository clean
- Make sure npm account is logged in in your terminal
- Run `npm run publish`

Any Questions ?

Theethawat/> Savastham

theethawat.sa@one.th

theethawat

https://theethawat.dev