

Fullstack Development

Preflight project - backend

[Github Repo](#)

Node JS backend framework

<https://github.com/vanodevium/node-framework-stars>

Packages

- Express JS

- `npm i express cors helmet body-parser`

- Typescript

- `npm i typescript ts-node tsconfig-paths`

- `npm i -D @types/cors @types/express tsc-alias`

- ORM

- `npm i drizzle-orm postgres dotenv`

- `npm i -D drizzle-kit nodemon`

ORM code

- All [files](#) in `./db` folder.
 - No need to copy `migration` folder.
- Change schema if needed.
- `.\drizzle.config.ts` ([Link](#))

Files

- `.env` (Copy from [here](#))
- `nodemon.json` ([Link](#))
- `.gitignore` ([Link](#))

Typescript

- `npx tsc --init`
- Note the `outDir` and `include` fields in `./tsconfig.json`. (See code on the next page)
 - Once we use `outDir` option any non-`./src` folder needs to be included, including the root.

./tsconfig.json

```
{
  "ts-node": {
    "require": ["tsconfig-paths/register"]
  },
  "compilerOptions": {
    // ...
    "outDir": "./dist",
    "baseUrl": "./",
    "paths": {
      "@src/*": [".src/*"],
      "@db/*": [".db/*"]
    }
    // ...
  },
  "include": ["./*", ".src/**/*", ".db/**/*"]
}
```







Script

```
{  
  // ...  
  "scripts": {  
    "db:generate": "drizzle-kit generate",  
    "db:push": "drizzle-kit push",  
    "db:migrate": "ts-node ./db/migrate.ts",  
    "db:prototype": "ts-node ./db/prototype.ts",  
    "dev": "nodemon",  
    "build": "tsc && tsc-alias",  
    "start": "node ./dist/src/index.js"  
  }  
}
```



Backend logic

- `./src/index.ts` [\(Link\)](#)
- Start dev
 - `npm run dev`
- Build
 - `npm run build`
- Start production
 - `npm run start`

Containerization

-  `./Dockerfile` ([Link](#))
-  `./dockerignore` ([Link](#))
-  `./docker-compose.yml` ([Link](#))
-  `./.env.test` from `./env`
 - Change `POSTGRES_HOST=postgres_container`
-  `docker compose --env-file ./env.test up -d --force-recreate --build`

Push to Dockerhub

- Create an account at <https://hub.docker.com>.
- Create repository called `pf-backend`.
- Tag image
 -  `docker tag pf-backend [DOCKERHUB_ACCOUNT]/pf-backend:latest`
- Push image
 -  `docker push [DOCKERHUB_ACCOUNT]/pf-backend:latest`