Node.js V4

<https://hendrixer.github.io/API-design-v4/>

## Tools

### Runtime

nodeJs, is environment, it’s a runtime, javascript is the language

### Framework

Express

### Database

Psql, with Prisma as ORM.

### Hosting

Render

### The App

Imaginary chaneLog app, product manager or engineer post product updates.

Typescript is with import, cmJs is with require.

Diff from import to require?

When you create a server is like adding an event listener to a port, when a requests comes you just run the call back.

### Anatomy of the API

Server: app without visual representation that is always running, connected to network and shared with other clients. Gatekeeper. Servers operate on a port (where network connections start and end). IP address, unique server location on the network.

API: the code that runs on the server

Route: unique combination of URL path and a http method. It’s the name of the event (method+url).

-get, post, put, patch, delete, options…

Most popular design pattern for routes is REST.

Route handler: is the callback, the function that response to every user.

### ORM

Db is an abstraction of how you interact with an ssd on a disk.

ORM is the library you install to interact with your db.

npm i typescript ts-node @types/node prisma --save-dev

Then create a tsconfig.json file which is the config file for TypeScript. Add this to that file:

{

"compilerOptions": {

"sourceMap": true,

"outDir": "dist",

"strict": true,

"lib": ["esnext"],

"esModuleInterop": true

}

}

npx prisma init

this installed the prisma CLI not the client, not the ORM.

Now we install the prisma ORM.

Prisma generates different SDKs depending on your schema.

npm i @prisma/client –-save

Now you can run migrations, first make sure you added your d burl to prisma config.

npx prisma migrate dev --name init